Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
December 2008, 15(4): 112–120
www.sciencedirect.com/science/journal/10058885
The Journal of China Universities of Posts and Telecommunications
www.buptjournal.cn/xben

Identifying file-sharing P2P traffic based on traffic characteristics

CHENG Wei-qing (✉)[1,2], GONG Jian[2], DING Wei[2]

1. College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China
2. School of Computer Science and Engineering, Southeast University, Nanjing 210096, China

Abstract

This article focuses on identifying file-sharing peer-to-peer (P2P) (such as BitTorrent (BT)) traffic at the borders of a stub network. By analyzing protocols and traffic of applications, it is found that file-sharing P2P traffic of a single user differs greatly from traditional and other P2P (such as QQ) applications' traffic in the distribution of involved remote hosts and remote ports. Therefore, a method based on discreteness of remote hosts (RHD) and discreteness of remote ports (RPD) is proposed to identify BT-like traffic. This method only relies on flow information of each user host in a stub network, and no packet payload needs to be monitored. At intervals, instant RHD for concurrent transmission control protocol and user datagram protocol flows for each host are calculated respectively through grouping flows by the stub network that the remote host of each flow belongs to. On given conditions, instant RPD are calculated through grouping flows by the remote port to amend instant RHD. Whether a host has been using a BT-like application or not can be deduced from instant RHD or average RHD for a period of time. The proposed method based on traffic characteristics is more suitable for identifying protean file-sharing P2P traffic than content-based methods. Experimental results show that this method is effective with high accuracy.

Keywords traffic identification, concurrent flows, P2P, discreteness of remote hosts, discreteness of remote ports

1 Introduction

There are many popular P2P systems tailored for sharing large files, network TV or music on the Internet, such as BT, PPLive, eMule, FastTrack, eDonkey, PPStream and KuGoo [1–4]. These P2P systems can overcome the limits of traditional download systems (Client/Server mode), and they have the features that the more users download the same file or enjoy the same network TV or music programs, the higher the download speed or the more uninterrupted play will be. The P2P traffic has accounted for about 40%–70% of Internet traffics [5]. The P2P systems can provide fast information sharing service by taking full advantage of peer communication capability to occupy more bandwidth than traditional applications. Because of finite bandwidth resource, however, file sharing P2P traffic is apt to result in poor performance of critical applications such as Web and E-mail. Thus, some internet service providers (ISP), enterprise networks, and campus networks may hope to take measures to limit the use of such P2P applications during working hours or rush hours on the Internet. Broadband ISPs may also wish to limit P2P traffic to cut down the expenditure for upstream ISPs [6]. To those ends, file-sharing P2P traffic should be identified first. To describe conveniently, we classify BT-like P2P applications that usually establish as many P2P connections as possible when sharing large files or rich media [7] as class I, whereas those that often establish a few P2P connections as class II, such as QQ, Skype and MSN. Considering the high cost, it is not much possible and economical to restrict P2P efficiently at the core network, while it may be a sensible choice to control BT-like traffic at the borders of a stub network, where it is convenient to enforce policy-based traffic control on host granularity with acceptable costs. In this article, we focus on identifying class I P2P traffic at the borders of a stub network, and try to observe which hosts are generating class I P2P traffic, to render restriction of class I P2P traffic on host granularity possible.

The commonly used methods for application recognition or traffic identification are content-based methods, such as those based on port numbers or application signatures [4,8–12]. However, because of the arbitrariness of the design and implementation of P2P protocol and software, and the lack of

Received date: 01-04-2008
Corresponding author: CHENG Wei-qing, E-mail: chengweiq@njupt.edu.cn

adaptability and scalability of these methods, identification rules or even identification software must be updated along with the new versions of known P2P applications. Moreover, these methods are usually incapable of identifying encrypted or unfamiliar P2P traffic. In this article, we propose a new method that makes use of comparatively steady non-content characteristics and traffic characteristics of applications, as the basis to identify class I P2P traffic. By observing and analyzing the traffic for individual user hosts and the application protocols, we find that the maximum difference between BT-like traffic of a single user and traffic of traditional or class II P2P applications might lie in the distribution characteristic of involved remote hosts. Moreover, the distribution of remote ports is also worthy of consideration. Therefore, we define two metrics 'the discreteness of remote hosts' (RHD) and 'the discreteness of remote ports' (RPD) that are used to observe whether user traffic contains class I P2P traffic therein. Tests on real Internet traffic show that BT-like traffic can be detected quite soon based on these two metrics with high accuracy. This article takes BT for an example to discuss the traffic characteristics of class I P2P applications, and that the traffic identification method proposed is universal to all class I P2P applications.

The remaining sections of this article are organized as follows. Section 2 discusses related research. Section 3 explores traffic characteristics of popular applications, and deduces that BT-like traffic can be identified according to the discreteness of remote hosts and remote ports in single host's traffic. Section 4 presents the RHD and RPD-based method to identify class I P2P traffic. Section 5 describes the experiments and results to show the efficiency of the proposed method. Section 6 concludes the article.

2 Related work

Traffic identification has already been widely applied for several years in many fields, such as Internet protocol (IP) quality of service, intrusion detection, usage-based accounting, and application-specific traffic engineering. For example, intrusion detection systems usually contain signature matching-based application recognition modules. Cisco Internet operating system (IOS) has introduced network-based application recognition (NBAR) [4] feature, which identifies applications and protocols from Layer 4 through Layer 7 based on contents of packets. Identifying P2P application traffic primarily uses fixed port-based, dynamic port-based, and application signature-based recognition methods [8,16]. But these methods are unlikely to identify encrypted P2P traffic or unfamiliar P2P traffic; moreover, identification rules must be updated frequently to adapt to the knowledge of P2P applications.

Thus, to identify P2P traffic, new methods not merely relying on contents of packets have been adopted since 2004 [6,13–15]. In Ref. [13], a method of identifying P2P flows based on connection patterns of P2P networks in the Internet core is first proposed. However, there are two defects in this method: first, its algorithm relies on two primary heuristics to identify P2P flows and several other heuristics to decrease the risk of false positives, which results in poor performance. Second, all flows generated by both source and destination hosts behind network address translators (NAT) might be considered as P2P by error, according to one of the primary heuristics: source-destination IP pairs that concurrently use both transmission control protocol (TCP) and user datagram protocol (UDP) during t but do not use special ports are considered as P2P. Our method will overcome these defects. This method can identify BT-like traffic even if it traverses NATs by setting the measurement point ahead of the NAT of a stub network. In addition, the method requires no exceptional disposals and leads to better performance. BLINC (BLINd Classification) [14] analyzes patterns of host behavior at the transport layer and operates at three levels of host behavior: social, functional and application levels. It is the first study that associates hosts with applications. However, no full algorithm is given in Ref. [14], and BLINC should be carefully refined to improve its practicability. A technique that relies on observation of the size of the first five data packets of a TCP flow is proposed in Ref. [15] to identify applications, which is quite simple and allows early classification of applications. Another method based on the analysis of the protocol used by a P2P application and an extraction of specific patterns unique to the protocol that can be shown by an IP packet level, is designed in Ref. [6] to detect P2P traffic. It even supports identification of encrypted P2P traffic, although the pattern extraction is complex. However, both methods proposed in Refs. [15] and [6] can be easily circumvented, and they are not tolerant to loss in packet collection. In contrast, our method based on two elaborately defined metrics that are insensitive to loss in packet collection is efficient and robust.

3 Traffic characteristic analysis of BT-like and other applications

To find out the essential traffic difference between BT-like applications and other applications, we first study protocols and traffic of applications.

3.1 Definitions of TCP/UDP flows

Both traffic characteristic analysis and identification of class I P2P traffic on the borders of stub networks need

classify packets into flows. In this article, a flow is defined by 5-tuple (local IP, local port, remote IP, remote port, protocol), and a flow is considered to have expired if no more packets belonging to the flow have been observed for a certain period of time [16].

Internet protocol packets that shuttle between specific local endpoint (local IP, local port, protocol) and specific remote endpoint (remote IP, remote port, protocol), carry transport-level protocol data units, and arrive under specified timeout constraints, should belong to a TCP or UDP flow. An IP packet belonging to no active flows (see below) will result in a new flow. A flow has two states as follows:

1) S_ACTIVE, i.e. active state. The state for a new flow is S_ACTIVE, and it remains unchanged until no new packets of the flow arrive for longer than specified timeout interval.

2) S_TIMEOUT, i.e. timeout state. If no new packets of a flow arrive for longer than specified timeout interval, the state of the flow turns to S_TIMEOUT. The timeout intervals for TCP and UDP flows are both set as 4 seconds throughout this article.

Flows with the state of S_ACTIVE are termed active flows, flows with the state of S_TIMEOUT are termed inactive flows, and active flows that coexist at a time are called concurrent flows at that time.

3.2 Analysis method

To analyze traffic characteristics of applications, we designed a flow statistics tool. This tool can monitor IP traffic and classify IP packets into flows, calculate the duration, the total of inbound/outbound octets, and the total inbound/outbound packets of each flow, update the state of each flow, and reckon the number of concurrent flows for each local host at any time. To analyze traffic characteristics of different applications, we run each application separately and deduce traffic characteristics based on the flow statistics output of the tool and analysis of application protocols.

3.3 Traffic characteristics of BT

Through observations, we find that BT traffic of a single BT client usually contains both TCP and UDP flows. Now, NAT [17–18] devices are often placed at the borders of stub networks to allow internal network hosts with private addresses to transparently communicate with outer hosts. Among BT traffic of an internal network host, TCP flows consist of P2P connections with outer hosts with globally unique addresses (global hosts), TCP connections for attempting to connect with peers in other internal networks or newly off-line peers and TCP connections to tracker servers (if any). The UDP flows consist of P2P connections to hosts in other internal networks or to global hosts with ‘refusing all inbound SYN packets’ configured in their firewalls, and probably UDP flows between the host and tracker servers (if exist).

We sum up the characteristics of a BT client’s traffic as follows:

1) TCP and UDP flows often coexist. But a BT client seldom has both TCP and UDP flows within one and the same peer host at the same time during the process of downloading, although it may occur occasionally.

2) A BT client seldom has more than one association with one and the same remote endpoint at the same time during the process of downloading.

3) Because BT clients can operate on arbitrary ports, remote endpoints of a BT client usually differ in both remote port number and remote IP address.

4) The numbers of concurrent TCP or UDP flows are unstable, and so is the proportion between them. If no constraints are configured in a BT client, then the number and the proportion are mostly related to both the number of peers that tracker servers have returned and the ways those peers access the Internet. The number of concurrent flows usually ranges from a dozen to even more than a hundred.

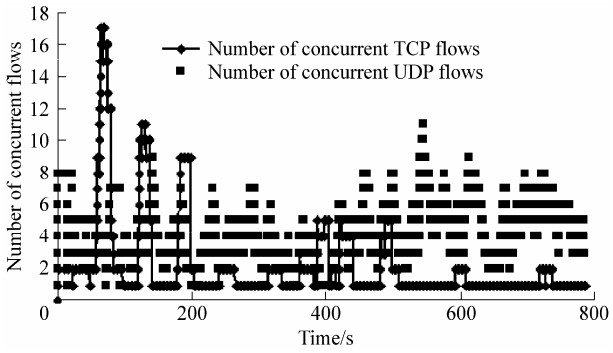
5) The ratio of short flows (containing less than 5 packets) to all flows is fairly high. The reasons for this are that peers as nonprofessional servers may take part in or leave BT downloading freely, and that there are sorts of limitations brought by NATs, firewalls and configuration of BT clients.

6) BT traffic usually contains more long flows than traffic of other type applications. Long flow is the flow that has plenty of packets (e.g., more than 100 packets) and lasts long time (e.g. more than 15 s).

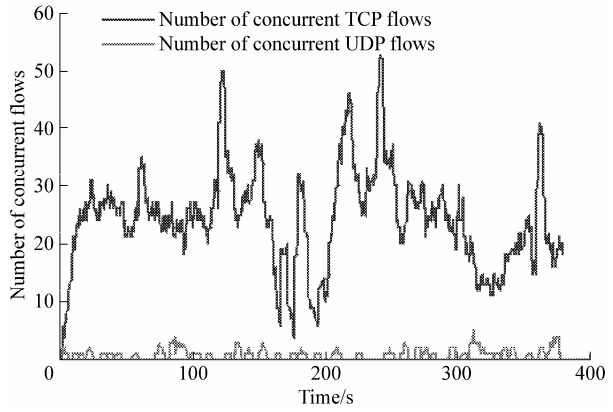
Source-destination IP pairs that concurrently use both TCP and UDP during t and do not use some specified ports (such as 53), are considered as P2P in Ref. [13]. From the above 1), it is seen that only a very small part of BT traffic can be identified by this method. From the analysis earlier, it seems that such features as the number of long flows, proportion of short flows, in-out packets ratio of a flow, and particularly the number of concurrent flows might be used to identify BT traffic. Here we only discuss the last feature.

We use BT to download a non-hot file and a fairly hot file respectively, and monitor the BT traffic using the flow statistics tool. The numbers of concurrent flows over a period of time in two cases are shown in Fig. 1. It shows that the number of concurrent TCP or UDP flows fluctuates with time; the total number of both types of concurrent flows when downloading the non-hot file may be only several at times, whereas it is usually a dozen or even more when downloading the hot file. Furthermore, the bit rate obtained when downloading a hot file is usually much higher. It is obvious that the more peers a BT client connects to, the more

concurrent flows will be and the more bandwidth the BT client can obtain.



(a) The number of concurrent flows when downloading a non-hot file (Nov 8 14:08–14:21 2005)



(b) The number of concurrent flows when downloading a fairly hot file (Nov 13 15:02–15:09 2005)

Fig. 1 The number of concurrent TCP/UDP flows in two cases of BT downloading

3.4 Traffic characteristics of other types of applications

Here, we first discuss three traditional client/server-based applications: Web, file transfer protocol (FTP), and E-mail. During visiting Web sites, a user often tells the Web client Web addresses by clicking hyperlinks. One click will result in one or even several Web pages returned. A Web page is usually composed of a hyper-text markup language (HTML) document and other elements (such as images, flashes, and even other HTML documents) that may even be separately stored in several Web servers. A Web client usually in turn opens several local TCP ports that are used to establish connections with service ports of relevant Web servers for downloading all elements of a Web page in parallel. Figure 2 shows the variation of the number of concurrent flows when a single user accessing SINA Web site. The Web traffic of a single user is generally intermittent, and has quite a few concurrent TCP flows now and then, where multiple flows commonly share one remote endpoint. In addition, UDP flows in Web traffic are just composed of domain name system (DNS) requests and responses. However, according to the flow definition and the specified timeout, there might be a number of 'concurrent' UDP flows.

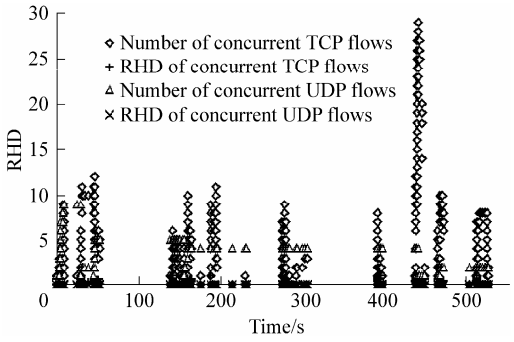


Fig. 2 The number and RHD of concurrent flows when visiting SINA Web site on a host (Nov 18 11:42–11:51 2005)

With regard to FTP, a traditional FTP client always establishes a control connection with a FTP server to transfer commands (such as cd, ls, get) and the execution status. Any command having contents to be returned will render a data connection to be established. Therefore, there are 1 or 2 concurrent TCP flows and 1 concurrent UDP flow (DNS related) at most during accessing FTP servers by using a FTP client.

As for E-mail, there are two styles of E-mail prevailing today. One is the web style. Traffic of such E-mail differs from traffic of general Web in that its uploading traffic might be large. The number of concurrent flows is also associated with web pages. The other style is traditional E-mail based on simple mail transfer protocol and post office protocol with only few concurrent flows in the traffic.

Class II P2P applications such as popular QQ and Skype, primarily provide text, audio or video exchange between peers. A QQ client usually uses UDP to contact QQ servers or to establish P2P connections with its peers for voiced chatting, while uses TCP for written chatting. The number of concurrent flows in a single host's QQ or Skype traffic primarily depends on the number of peers the client simultaneously communicates with, and it may grow big during accessing servers or updating peers information.

3.5 The difference in essence between BT-like and other applications

It can be seen from the analysis earlier that there usually exist few concurrent flows in a single user's traffic of FTP, traditional E-mail, or class II P2P applications. Whereas, both Web and BT traffic may contain many concurrent flows, thus it is inappropriate to identify BT traffic simply based on the

number of concurrent flows.

We show flow keys of concurrent TCP or UDP flows at certain times of BT traffic in Fig. 1(a) in Table 1. Table 2 provides flow keys of concurrent TCP flows at certain times of Web traffic in Fig. 2. By comparing the flow keys in the two tables, we find that the reason why BT-like applications can get more network resources is just the essential difference between them and other applications. A client of class I P2P applications is usually voluntary to connect to more peers to the best of its abilities to download large files as soon as possible. Whereas, the main goal of class II P2P applications is to facilitate users communication with each other; thus, each client usually communicates with merely few peers simultaneously. Traditional applications use the C/S mode and in general each client communicates with only one or a small quantity of servers simultaneously. Therefore, even if there exist a number of concurrent flows, they can be aggregated into few groups by visited servers with multiple flows versus one server. For example, in general, many concurrent flows may be involved in a Web page's traffic, but the involved servers are always few in one or few stub networks. Compared with the Web, a BT client occupying volumes of bandwidth almost always communicates with many peers; moreover, the possibility that the peers scatter in various stub networks is much bigger than the possibility that peers congregate in few stub networks. Therefore, we consider using the discreteness of remote hosts (RHD) of concurrent flows to identify BT-like traffic.

Table 1 Examples of flow keys, RHDs and RPDs of concurrent flows when downloading a non-hot file on a single host

Time	t_1	t_2
Concurrent TCP or UDP flows at a time: (local port, remote IP address, remote port)	TCP flows: $n=8$ 1061 221.x.7.152 12170 1060 60.x.83.68 40617 1059 218.x.216.40 15922 1057 218.x.17.228 27113 1056 60.x.123.191 16277 1055 218.x.216.22 15922 1051 218.x.17.228 27113 1048 218.x.111.100 8080	UDP flows: $n=7$ 15922 62.x.154.47 14342 15922 61.x.160.216 2994 15922 61.x.241.159 16881 15922 221.x.79.175 31832 15922 221.x.55.209 10215 15922 24.x.49.82 21018 15922 218.x.17.228 27113
(a stub network that remote hosts reside in, the number of flows whose remote hosts reside in this network)	221.x.7.152/23 1 60.x.83.68/23 1 218.x.216.40/23 2 218.x.17.228/23 2 60.x.123.191/23 1 218.x.111.100/23 1	62.x.154.47/23 1 61.x.160.216/23 1 61.x.241.159/23 1 221.x.79.175/23 1 221.x.55.209/23 1 24.x.49.82/23 1 218.x.17.228/23 1
RHD	$m=6$, RHD=2	$m=7$, RHD=2.81
RPD	$m=6$, RPD=2	$m=7$, RPD=2.81

Note: x substitutes for any value ranging from 0 to 255.

Moreover, today's BT-like applications seldom use well-defined port numbers; they prefer randomly designated port numbers with the intention to avoid being identified easily. In contrast, Web application usually uses fixed port numbers, and each class II P2P application usually uses both predefined and random port numbers. Therefore, it is suggested to combine the RPD with RHD to identify BT-like traffic.

Table 2 Examples of flow keys, RHDs and RPDs of concurrent TCP flows when visiting SINA web site on a single host

Time	t_1	t_2
Concurrent TCP flows at a time: (local port, remote IP address, remote port)	2907 a.b.201.130 80 2902 a.b.201.89 80 2901 a.b.201.89 80 2899 a.b.201.89 80 2896 a.b.201.89 80 2893 c.d.78.202 80 2890 a.b.201.12 80 $n=7$	3613 a.b.201.90 80 3612 a.b.201.235 80 3611 a.b.201.90 80 3610 a.b.201.130 80 3609 a.b.201.130 80 3607 a.e.167.57 80 3605 a.b.201.98 80 3601 a.b.201.98 80 3587 a.b.201.98 80 $n=9$
(a stub network that remote hosts reside in, the number of flows whose remote hosts reside in this network)	a.b.201.130/23 6 c.d.78.202/23 1	a.b.201.90/23 8 a.e.167.57/23 1
RHD	$m=2$, RHD = 0.43	$m=2$, RHD = 0.37
RPD	$m=1$, RPD = 0	$m=1$, RPD = 0

Note: a, b, c, d and e substitute for specific values for bytes in an IP address.

4 RHD and RPD-based identification of class I P2P traffic

4.1 Definition of RHD

In information theory and communication theory, the quantity of information for a message is tightly associated with the probability that the message occurs, where the smaller probability is, the larger amount of information the message will contain. Similarly, with regard to concurrent flows of a single host, the more proportion of flows with their remote hosts belonging to the same stub network, the less discreteness of remote hosts of these flows.

Therefore, referring to principle of entropy, we define RHD for concurrent TCP or UDP flows at time t as follows:

$$D_{\text{inst}}(t)=\frac{1}{n}\sum_{i=1}^{m}\log_2\frac{n}{x_i} \tag{1}$$

where n denotes the number of concurrent flows at time t, m denotes the number of diverse stub networks that remote hosts of the flows belong to (obviously, $m\leqslant n$), and x_i denotes the number of flows with their remote hosts residing in the same network i. The network prefix length of stub networks is set as 23 in this article, and RHDs should be calculated for concurrent TCP or UDP flows.

4.2 Comparison of RHDs for different application traffic

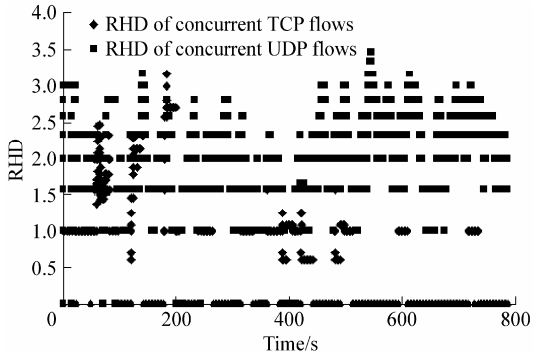
1) Discreteness of remote hosts for BT traffic of a single host

The number of concurrent flows for BT traffic of a single

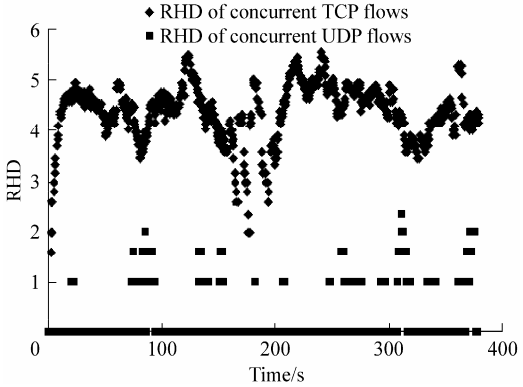
host fluctuates with the amount and states of peers (Fig. 1), but there are great probabilities that RHD has fairly high numerical value anyway (Fig. 3). To make it clear, we show both flow keys and RHDs of concurrent TCP or UDP flows at some traffic times in Fig. 1(a), as shown in Table 1.

2) RHD for traditional and class II P2P application traffic of a single host

During the time of a user visiting a Web site, the number of concurrent flows may be great at some times, but RHD is always low because different flows often have the same remote host and remote port. Figure 2 shows both RHD and the number of concurrent flows when a single user accesses SINA Web site. Table 2 provides specific flow keys and RHDs of concurrent TCP flows at some times. When a user visits multiple Web sites simultaneously (i.e., multiple Web pages are downloaded from different Web sites), the RHD of TCP flows may increase. However, except for the Web traffic aroused by special programs that can automatically access many Web sites at the same time, the RHD for ordinary Web traffic of a user usually remains moderately low, because few users visit many (e.g., more than 5) Web sites simultaneously.



(a) RHD of concurrent flows when downloading a non-hot file



(b) RHD of concurrent flows when downloading a fairly hot file

Fig. 3 RHD for BT traffic in two cases of Fig. 1

When a user accessing a FTP server, the number of concurrent flows remains few, and RHD remains 0 because of duplicate remote host for both control and data connections. The RHD will increase when a user accesses multiple FTP servers. However, this case seldom occurs. The reasons might be that seldom users have such habit because the obtainable data rates usually dissatisfy users. The RHD for E-mail traffic of a single host is also low.

For QQ or Skype traffic of a single host, RHDs usually remain moderately low and moderately stable because only a small amount of concurrent flows exist therein. However, if a host uses QQ and Web concurrently, RHD may approach to the value of downloading a non-hot file with the use of a BT client. Thus, we introduce RPD below and some preprocesses to make hosts generating BT-like traffic more prominent.

4.3 Definition of RPD

Similar to RHD, RPD for concurrent TCP or UDP flows at time t is defined as:

$$P_{\text{inst}}(t)=\frac{1}{n}\sum_{i=1}^{m}\log_2\frac{n}{x_i} \tag{2}$$

where n denotes the number of concurrent flows at time t, m denotes the number of diverse port numbers of the flows (obviously, $m\leqslant n$), and x_i denotes the number of flows with their remote ports equal to the ith port number. RPDs should also be calculated for concurrent TCP or UDP flows respectively.

In general, the RPD for BT-like traffic is the highest, followed by the RPD for class II P2P traffic, and the lowest is the RPD for traditional application traffic. Examples are given in Tables 1 and 2.

4.4 Algorithm

Our RHD and RPD-based method is built on the earlier analysis of both RHD and RPD for traffic of different applications on a single host and user behavior characteristics. This method needs to monitor traffic of every internal host at the borders of a stub network, and uses two criteria as follows:

Criteria 1 (C1) Instant RHD-based identification: if instant RHD for UDP flows of a host's traffic is greater than threshold D_{U} (e.g. 2.2), or the sum of instant RHD for TCP flows and that for UDP flows is greater than D_{sum} (e.g. 2.8), it can be concluded that BT-like traffic is contained in the host's traffic.

Criteria 2 (C2) Average RHD-based identification: if the sum of the average RHD for TCP flows and that for UDP flows of a host's traffic during an measurement interval (e.g. T=10 s) is greater than threshold D_{sumOfAvg} (less than D_{sum}, e.g., 2.5), it can be concluded that BT-like traffic is contained in the host's traffic.

The algorithm is described as follows:

Monitor every inbound or outbound IP packet and classify it into a specific flow based on the 5-tuples and the flow

timeout.

Flows are first grouped by their local IP addresses. A flow record includes flow key, flow state, start time, and last packet arrival time of the flow. No packet payload needs to be monitored and recorded.

For each active host ***H*** during every measurement interval (T s):

At every G s (G<T/20):

1) Check whether each active TCP or UDP flow of host H has timed out, and update the state of timed out flows.

2) If the number of concurrent TCP flows is not less than MintoPreDispose (tentatively set as 12), then group the TCP flows according to pseudo remote endpoint (remote IP network prefix, remote port), and weed those groups with at least MinFCtoWeed (tentatively set as 5) members (not delete, only exclude these groups of flows from calculating RHD or RPD). The same step is used for concurrent UDP flows.

3) Calculate two RHDs (i.e. instant RHD, defined by Eq. (1)) for concurrent TCP and UDP flows of the host, and RHD is taken as zero if the number of concurrent flows is zero.

4) If RHD is between [LowRHD, HighRHD], then group concurrent flows according to their remote ports, calculate the RPD (i.e. instant RPD defined by Eq. (2)), and update instant RHD as follows:

$$D_{\text{inst}}(t) = w_1 D_{\text{inst}}(t) + w_2 P_{\text{inst}}(t) \tag{3}$$

where LowRHD, HighRHD, w_1 and w_2 are tentatively set as 0.5, 3.5, 0.6 and 0.4 respectively.

5) Use criteria 1 to distinguish whether BT-like P2P traffic is contained in the current traffic of host H. If yes, then report, 'It was identified that BT-like traffic was contained in traffic of host H according to C1'.

6) Update two average RHDs of TCP and UDP flows of the host. Let $D_{\text{inst}}^{(k)}$ be the kth instant RHD during this measurement interval, and the average RHD is defined as:

$$D_{\text{avg}}^{(k)} = \frac{k-1}{k} D_{\text{avg}}^{(k-1)} + \frac{D_{\text{inst}}^{(k)}}{k}; \quad k \geqslant 1 \tag{4}$$

Adopt Criteria 2 to distinguish whether BT-like P2P traffic is contained in traffic of host H. If yes, then report, 'It was identified that BT-like traffic was contained in traffic of host H according to C2'.

Delete records of timed out flows.

Class I P2P traffic can be fairly efficiently identified quite soon using C1 that only checks instant-RHDs for concurrent flows of a host. However, if thresholds are set higher, BT-like traffic with few peers or with a few peers accompanied by much Web traffic may not be identified. On the contrary, if thresholds are set lower, user traffic for accessing many traditional servers providing different services concurrently may be identified as containing BT traffic. Since false positives should be minimized, thus thresholds should rather be set a little higher.

Because of the burst property of Web traffic, even if the instant RHD for pure Web traffic of a single host may be a little high sometimes, it will not last long, i.e., the average RHD is usually very low. The instant RHD for mixed traffic of Web and class II P2P of a single host may be rather high sometimes, but it will not last long, either, i.e., the average RHD is usually low. The instant RHD for traffic of a single host containing BT-like traffic is often not low; moreover, it will last long, which usually results in high average RHD. Thus, C2 is better in accuracy while not better in real time than C1.

In addition, it must be mentioned that the method proposed in this article is used to identify whether the traffic of user hosts (excluding traditional servers) in a stub network contains BT-like traffic; traffic of traditional servers (e.g. Web, FTP, E-mail, TELNET servers) in the network must be omitted.

5 Experiments

5.1 Experimental setup

We designed a filter according to the above algorithm. To verify the efficacy of our RHD- and RPD-based method, we also designed a content filter that identifies applications by matching port numbers and application signatures. These two filters are run on two hosts, which as well as some other hosts are connected to a share Hub residing on a subnet in our campus network. The two filters monitor the traffic between hosts connected to the Hub and hosts outside the subnet and perform class I P2P traffic identification.

In our tests, by experience, we set both flow timeouts for TCP and UDP flows as 4 s, the length of network prefix as 23, the measurement interval T as 10 s, and D_{sumOfAvg} as 2.5.

5.2 Experimental results

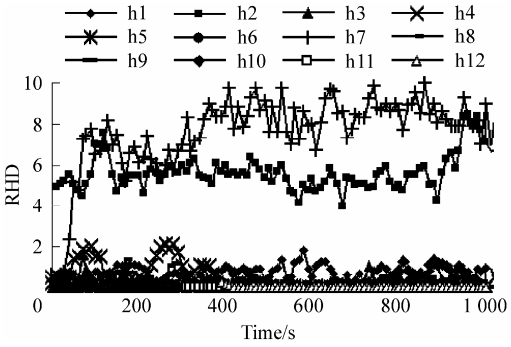
To save the length of the article, we mainly verify the efficacy of C2. Results of three experiments are shown in Fig. 4, where X-coordinate is time (unit: s), and Y-coordinate is the sum of average RHD for TCP flows and that for UDP flows of each host.

According to C2, Fig. 4(a) shows that host h7 and h2 generate BT-like traffic. By matching IP packets with well-known port numbers and application signatures, it is seen that host h7 uses PPLive to watch network TV, host h2 uses BT to download files, host h1 uses Web, QQ, and MSN, host h4 uses network time service and the Web, and other hosts primarily use traditional applications. Thus, the judgments about all hosts prove to be true.

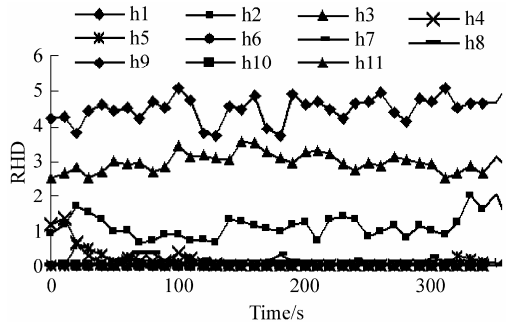
Figure 4(b) shows that hosts h1 and h3 generate BT-like traffic. By analyzing IP packets, it can be seen that host h1 uses PPLive, host h3 uses KuGoo to enjoy music, host h2 uses QQ and the Web, and other users use Web or FTP service. There are no false positives or false negatives in this experiment.

Figure 4(c) shows that hosts h2 and h5 generate BT-like traffic. By analyzing IP packets, it can be seen that hosts h2 and h5 use BT, host h1 uses MSN, QQ and the Web, and host h6 uses network time service and the Web, and other users primarily use traditional applications. There are hardly any false positives or false negatives in this experiment.

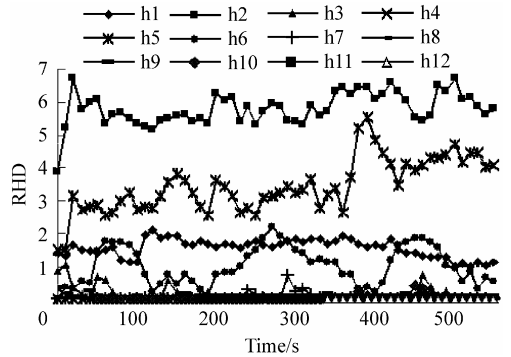
Many experiments are conducted, of which the results show that the accuracy of BT-like traffic identification based on C2 averagely exceeds at least 95%.



(a) Experiment 1 (22:03–22:19 July 5, 2006)



(b) Experiment 2 (15:50–15:56 June 26, 2006)



(c) Experiment 3 (21:45–21:54 July 5, 2006)

Fig. 4 The sum of average RHD for TCP flows and that for UDP flows of each host

To have a glance at C1, we display the sum of instant RHD for TCP flows and that for UDP flows of each host in experiment 3 every 400 ms or so in Fig. 5. Figure 5 shows that although there exist false positives and false negatives when using C1 to identify BT-like traffic, the confidence of the judgments by C1 is fairly high.

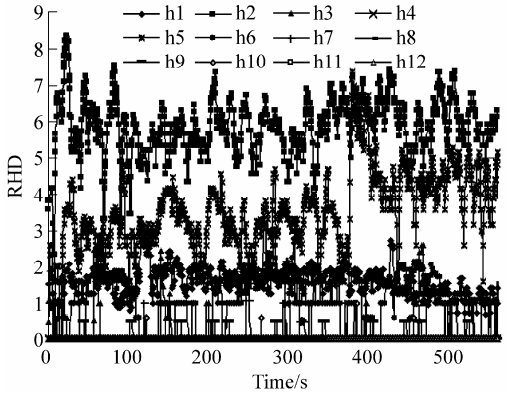


Fig. 5 The sum of instant RHD for TCP flows and that for UDP flows of each host in experiment 3 (21:45–21:54 July 5, 2006)

6 Conclusions

This article focuses on identification of file-sharing P2P traffic of single hosts based on a stable non-content characteristic, that is, traffic characteristics. By analyzing application protocols and traffic, it is found that the most striking difference between traffic of file-sharing P2P applications and traffic of traditional and QQ-like P2P applications lies in the distribution of remote hosts as well as remote ports involved. Remote hosts of flows always cluster in few networks in the latter traffic, whereas always fairly disperse in the former traffic. Remote ports of flows are usually well known and few in traffic of traditional applications, whereas greatly disperse in BT-like traffic. There are both fixed and random remote port numbers in the QQ-like P2P traffic. Therefore, a method based on RHD and RPD is proposed to identify whether a local user host has been generating BT-like traffic at the edge of stub networks (such as access networks or enterprise networks), where IP addresses of local hosts are visible. Experimental results show that the accuracy of our method is more than 95%.

Unlike content-based methods, the method proposed needs neither anatomy of P2P protocols nor packet-payload examination. The method is traffic characteristics-based, and it can identify whether there are class I P2P traffic no matter what ports (fixed, random, or disguised ports) and application signatures are used, no matter whether encryption is used or not, and no matter whether protocols are known or unknown. Thus, the method is better in scalability than content-based

methods.

Compared with other non-content methods, the proposed method has similar scalability, better performance, better accuracy particularly for traffic between hosts behind NATs, and is more suitable for detection of BT-like traffic to restrict it during working hours.

Acknowledgements

This work was supported by the National Basic Research Program of China (2003CB314804), the Research Program of NUPT (NY206010).

References

1. BitTorrent. 2005, http://wiki.bitcomet.com/help-zh/ tracker (in Chinese)
2. Sung L, Li H. Neighbor selection strategies for P2P systems using tit-for-tat exchange algorithm. 2005, http://www. cs.uwa terloo.ca/~lgasung/
3. PPLive. 2006, http://www.pplive.com/zh-cn/index.html (in Chinese)
4. Network-based application recognition and distributed network-based application recognition. 2005, http://www.cisco.com/en/US/products/ ps6616/ productsios_protocol_grou p_home.html
5. BitTorrent will not be blocked and telecom providers are planning to charge by traffic. 2005, http://tele com.chinabyte.com/191/2129691.shtml (in Chinese)
6. Spognardi A, Lucarelli A, DiPietro R. A methodology for P2P file-sharing traffic detection. Proceedings of Second International Workshop on Hot Topics in Peer-to-Peer Systems (HOT-P2P 2005), Jul 21, 2005, San Diego, CA, USA. 2005: 52–61
7. Milojicic D, Kalogeraki V, Lukose R, et al. Peer-to-peer computing. 2003, http://www.hpl.hp.com/techreports/ 2002/HPL-2002-57R1.pdf
8. Choi T, Kim C, Yoon S, et al. Content-aware Internet application traffic measurement and analysis. Proceedings of IEEE/IFIP Network Operations and Management Symposium (NOMS 2004), Apr 19–23, 2004, Seoul, Korea. 2004: 511–524
9. Moore A, Papagiannaki K. Toward the accurate identification of network applications. Proceedings of Passive and Active Measurement Workshop 2005 (PAM2005), Mar 31–Apr 1, 2005, Boston, MA, USA. 2005: 41–54
10. Sen S, Spatscheck O, Wang D. Accurate, scalable in-network identification of P2P traffic using application signatures. Proceedings of the 13th international conference on World Wide Web (WWW 2004), May 17–22, 2004, New York, NY, USA. 2004: 512–521
11. Haffner P, Sen S, Spatscheck O, et al. ACAS: automated construction of application signatures. Proceedings of ACM SIGCOMM workshop on Mining network data. Aug 22–26, 2005, Philadelphia, PA, USA. New York, NY, USA: ACM, 2005: 197–202
12. Chou S C. Network behavior analysis and performance evaluation of peer-to-peer application. Taipei, China: National Taiwan University, 2004 (in Chinese)
13. Karagiannis T, Broido A, Faloutsos M, et al. Transport layer identification of P2P traffic. 2004 ACM SIGCOMM Internet Measurement Conference (IMC 2004), Oct 25–27, 2004, Taormina, Sicily, Italy. 2004: 121–134
14. Karagiannis T, Papagiannaki K, Faloutsos M. BLINC: Multilevel traffic classification in the dark. Proceedings of ACM SIGCOMM. Aug 22–26, 2005, Philadelphia, PA, USA. 2005: 229–240
15. Bernaille L, Teixeira R, Akodkenou I, et al. Traffic classification on the fly. Computer Communication Review, 2006, 36(2): 23–26
16. Sadasivan G, Brownlee N, Claise B, et al. Architecture for IP flow information export. 2006, http://www.ietf.org/ internet-drafts/draft-ietf-ipfix-architecture-12.txt
17. Egevang K, Francis P. The IP Network Address Translator (NAT). RFC1631. 1994
18. Senie D. Network Address Translator (NAT)-friendly application design guidelines. RFC3235. 2002

From p. 80

6. Tian J, Li L, Yang X. Fingerprint-based identity authentication and digital media protection in network environment. Journal of Computer Science and Technology, 2006, 21(5): 861–870
7. Clancy T C, Kiyavash N, Lin D J. Secure smart card-based fingerprint authentication. Proceedings of the 2003 ACM SIGMM Workshop on Biometrics Methods and Application, Nov 2–8, 2003, Berkley CA, USA. New York, NY, USA: ACM, 2003: 45–52
8. Juels A, Sudan M. A fuzzy vault scheme. Proceedings IEEE International Symposium on Information Theory, Jun 30–Jul 5, 2002, Lausanne, Switzerland. 2002: 408–409
9. Uludag U, Pankanti S, Jain A K. Fuzzy vault for fingerprints. Proceeding of International Conference on Audio- and Video- Based Biometric Person Authentication, Jul 20–22, 2003, Rye Brook, NY, USA. Berlin, Germany: Spring, 2005: 310–319
10. Hao F, Anderson R, Daugman J. Combining crypto with biometrics effectively. IEEE Transactions on Computers, 2006, 55(9): 1081–1088
11. Dodis Y, Reyzin L, Smith A. Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. Proceeding of Advances in Cryptology-Eurocrypt 2004 (LNCS 3027), May 2–6, 2004, Interlaken, Switzerland. Berlin, Germany: Springer-Verlag, 2004: 523–540
12. Sahai A, Waters B. Fuzzy identity-based encryption. Proceeding of Advances in Cryptology-Eurocrypt, May 22–26, 2005, Aarhus, Denmak. Berlin, Germany: Springer- Verlag, 2005: 457–473
13. Jin A T B, Ling D N C, Goh A. Biohashing: two factor authentication featuring fingerprint data and tokenised random number. Pattern Recognition, 2004, 37(11): 2245–2255
14. Liu M H, Jiang X D, Kot A C. Fingerprint reference-point detection. EURASIP Journal on Applied Signal Processing, 2005, 2005(4): 498–509