國立交通大學

網路工程研究所

碩士論文

基於連線模式之即時P2P檔案分享的流量辨識方法

Real Time P2P File Sharing Traffic Identification Based on Connection Patterns

研究生：陳薏卉

指導教授：王國禎　博士

中華民國九十八年六月

基於連線模式之即時P2P檔案分享的流量辨識方法

Real Time P2P File Sharing Traffic Identification Based on Connection Patterns

研 究 生：陳薏卉　　Student：Yi-Hui Chen

指導教授：王國禎　　Advisor：Kuochen Wang

國 立 交 通 大 學

資 訊 學 院

網 路 工 程 研 究 所

碩 士 論 文

A Thesis

Submitted to Institute of Network Engineering

College of Computer Science

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

June 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年六月

基於連線模式之即時P2P檔案分享的流量辨識方法

學生：陳薏卉　　指導教授：王國禎 博士

國立交通大學 資訊學院 網路工程研究所

摘 要

由於P2P應用的日漸風行，尤其是檔案分享的應用造成了一些問題，如大量的網際網路頻寬被P2P流量所佔用及非合法授權軟體或檔案之侵權等。為了解決這些問題，在本論文中，我們提出一種基於連線模式之即時P2P檔案分享的流量辨識方法，簡稱 RTI，來協助網路管理。RTI只需要五秒的流量便可即時地辨識出P2P檔案分享的流量，以供網路管理用。RTI分成三個階段，第一個階段是利用埠號來過濾掉非P2P的封包。第二個階段利用三個探索法則來辨識有使用P2P的主機位址。最後階段我們從有使用P2P的主機中，利用四個探索法則來辨識出P2P檔案分享的流量。為了評估此方法的有效性，我們收集

了校園的網路流量，並根據封包特徵碼的分類器來驗證我們的結果。實驗結果顯示，我們提供的RTI辨識正確率高達96.2%，且只有3.5%的誤判率。相對於的一樣利用五秒的流量，John [9] 只有64.8%的正確率和高達74.19%的誤判率。

關鍵詞：點對點、檔案分享、流量、辨識、探索法、連線模式、傳輸層行為。

Real Time P2P File Sharing Traffic Identification Based on Connection Patterns

Student：Yi-Hui Chen　　Advisor：Dr. Kuochen Wang

Department of Computer Science

National Chiao Tung University

Abstract

The use of peer-to-peer (P2P) applications is growing dramatically, particularly for sharing large video/audio files and software, which results in several serious problems, such as internet piracy and unreasonable utilization of network resources. To conquer these problems, in this thesis, we propose a heuristic-based real time file sharing traffic identification (RTI) scheme at the transport layer for facilitating network management. The proposed RTI only needs a 5 seconds trace to effectively identify P2P file sharing traffic in real time for network management tools to timely filter, block, or record the traffic. The proposed RTI can be divided into three phases. In the first phase, we use port numbers to filter out non-P2P packets. In the second phase, we use three heuristics to identify P2P-using hosts. These heuristics are based on connection patterns of P2P networks, i.e., the numbers of distinct destination IPs and ports, and the usage of UDP packets. In the last phase, we use four heuristics to identify P2P file sharing traffic from the P2P-using hosts identified in the second phase. To evaluate the effectiveness of our scheme, we used traces collected in our campus network for P2P file sharing traffic identification and a payload-based classifier for verifying our traffic identification results. Experimental results indicate

that the proposed RTI had the accuracy of 96.2% and the FPRate (false positive rate) of 3.5%. In contrast, John [9] had the accuracy of only 64.8% and FPRate of 74.19% using the same trace.

Keywords: Peer-to-peer, file sharing, traffic, identification, heuristic, connection pattern, transport layer behavior.

Acknowledgements

Many people have helped me with this thesis. I deeply appreciate my thesis advisor, Dr. Kuochen Wang, for his intensive advice and guidance. I would like to thank all the members of the *Mobile Computing and Broadband Networking Laboratory* (MBL) for their invaluable assistance and suggestions.

Finally, I thank my family for their endless love and support.

Contents

List of Figures

List of Tables

Abbreviations List

The list contains the main abbreviations used throughout this thesis.

RTI: Real time file sharing traffic identification

sIP: Source IP address

sPort: Source port number

dIP: Destination IP address

dPort: Destination port number

#: The number of

PAT: Port association table

#PSW: The number of packet size switching

TP: True positive

TN: True negative

FP: False positive

FN: False negative

FPRate: False positive rate

ML: Machine learning

Chapter 1

Introduction

1.1 Motivation

Over the past few years, *peer-to-peer* (P2P) applications have dramatically grown in popularity and constituted a significant part of the total traffic in Internet [1], which has been shown to occupy as high as 70% of total Internet network bandwidth [19]. The large usage of the P2P traffic has led to serious problems, such as unreasonable utilization of network resources and internet piracy. These problems bring challenges to *Internet Service Providers* (ISPs) in providing *quality of service* (QoS) and dealing with legal issues. Currently, most of P2P traffic belongs to file sharing applications, such as BitTorrent [2], eDonkey [3], and Gnutella [16]. Due to the above reasons, the real time P2P file sharing traffic identification is a crucial research issue. Therefore, our research goal is to identify P2P file sharing applications in real time to facilitate network management.

1.2 P2P traffic identification methods

Different schemes to identify P2P traffic have been proposed. The key points of each scheme are discussed in the following and summarized in Table 1.

Table 1. P2P traffic identification methods.

Approach	Port based [1]	Payload based [5]	Machine learning [6]	Heuristic based (proposed)
Characteristics	Only check the source and destination port numbers	Check packet payload to identify P2P flows	Select traffic features and use ML algorithms to analyze flows	Identify P2P traffic based on connection patterns
Real time	Yes	Yes	▪ Training phase: no ▪ Classification phase: yes/no, depending on features obtained in real time or not	Yes
Pros	Easy and fast	High accuracy	Self learning that can classify unknown applications	Easy and fast
Cons	Not suited for P2P applications	Not working when packets are encrypted	More complicated and training data pre-classification required	Empirical heuristic thresholds

Traditionally, network traffic can be classified based on mapping the port number to the registered application on the Internet Assigned Number Authority (IANA) list [4]. This port-based scheme is very simple and does not need to inspect any packet's payload. But it is highly unreliable for P2P applications since they use dynamic port numbers, even though many of them have their own default port numbers, such as BitTorrent (6881-6883, 6889) [2], eMule (4662) [3], etc.

A more reliable scheme involves inspection of packet payloads, called DPI (Deep

Packet Inspection) in deployed commercial tools [5]. This scheme provides high accuracy given a complete set of payload signatures. But there are some drawbacks:

- The payload signatures must keep updating.
- Privacy and legal concerns.
- It does not work for encrypted payload information or new types of P2P traffic with unknown signatures.

A more recent research is *machine learning* (ML) techniques for IP traffic classification [6]. This scheme involves three steps. First, features are predefined that can differentiate IP traffic. Then the ML classifier is trained to associate the predefined features with the training data. Finally, a ML algorithm is applied to classify incoming traffic. Therefore, the success of ML highly depends on the accurate predefined features, such as flow duration, total flow bytes count (not real time classification) and packet lengths of the first few packets, the number of destination IP (real time classification), and training data.

In this thesis, we propose a heuristic-based real time file sharing traffic identification (RTI) scheme. These heuristics are based on not only *connection patterns*, i.e., the transport layer behavior, but also port numbers. RTI consists of three phases: pre-processing phase, P2P-using host identification phase and P2P file sharing traffic identification phase. The purpose of the pre-processing phase is to speed up the identification process by using port number information. In the P2P-using host identification phase, we can identify which host uses P2P applications. Finally, we can identify P2P file sharing traffic from the P2P-using hosts identified in the P2P file sharing traffic identification phase. In three phases, we need only an arbitrary *five-second trace* to identify P2P file sharing traffic. And we do not need to record the first few data packets of each TCP flow [13][14][18]. We can perform traffic identification for any 5 second traces. Therefore, the contribution of the proposed P2P

traffic identification scheme is that it is quick enough for network management tools to timely block, filter, or record of P2P file sharing traffic. That is, the proposed scheme can identify P2P traffic in real time.

The rest of this thesis is organized as follows. Chapter 2 introduces related work. Chapter 3 discusses our design approach. Experimental results are presented in Chapter 4. Finally, we give concluding remarks and future work in Chapter 5.

Chapter 2

Related Work

In the literatures, heuristic-based P2P traffic identification methods have been proposed to overcome the limitations of port based and payload based methods. The comparison of related heuristic-based methods is shown in Table 2. Some notations in

Table 2. Comparison of heuristic-based P2P traffic identification methods.

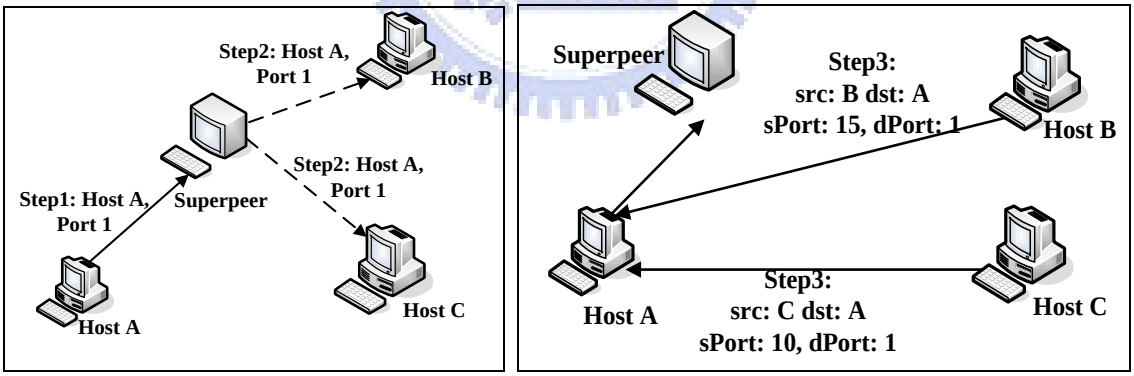
Heuristics	**Karagiannis [7]**	**Perenyi [8]**	**John [9]**	**RTI (proposed)**
P2P ports	Yes	Yes	Yes [7] [8]	No, because default port numbers may be different among P2P software versions and users
Non-P2P ports	Only for applications using use both TCP and UDP ports	Some common non-P2P ports	Heuristics from [7] and [8] , + well-known ports	0-1023 are non-P2P excluding web ports (80, 443, etc.)
Port usage	No	An IP address uses a port more than 5 times	If a port on a host is repeatedly used within 60 seconds, the host is marked as a P2P host	Port association [10] and flow aggregation
IP/port pairs	If |#dIP-#dPort| < 2, the traffic of {IP, port}pair is considered as P2P traffic	No	If #dPort-#dIP < 2 and #dIP > 5, the host is considered as a P2P host	The ratio between #dPort and #dIP of each host is large than a threshold

Table 2 have been defined in the Abbreviations List. In general, these heuristics can be divided into two kinds: *port numbers usage* and *connection patterns*.

2.1 Port numbers usage

The port-based scheme has been adopted in [7][8][9][10]. In [7][8][9], they mentioned that port-based analysis is still appropriate to distinguish traffic of common applications. But web ports (80, 8080, 443, etc.) are not among these, because the usage of web ports is not only for web surfing but also for some P2P applications. Lin et al. [10] proposed a method called *ports association*. When a session of an application needs to build multi-connections, the kernel assigns a contiguous ports range for every application, even for applications using randomized port numbers. Our scheme uses this property to speed up the traffic identification process and to aggregate associated flows to identify P2P file sharing traffic.

2.2 Connection patterns



(a) Initial connection from source host. (b) Destination hosts connect to source host.

Figure 1. A unique behavior of P2P applications [7].

Karagiannis [7] proposed a heuristic that uses a unique behavior of P2P applications when they are sharing files or making connections, which is shown in Figure 1 [7]. In Figure 1(a), host A informs the superpeer of its IP address and port willing to accept connections from other hosts. And the superpeer forwards the {IP,

port} pair to other hosts. In Figure 1(b), hosts willing to connect to host A, use the advertised {IP, port} pair. For host A, the corresponding port numbers of destination IP address (B, C) are (10, 15). This heuristic is also used in the proposed approach; however, we have scaled the threshold. Perenyi [8] presented a set of heuristics to identify P2P traffic, which can be divided into two categories. One is port numbers usage, and the other is P2P applications using both TCP and UDP protocols, which is similar to that in [7]. This heuristics is not used in our approach, because we only need short traces to identify traffic. John [9] used a combination of the heuristics by Karagiannis [7] and Perenyi [8] with additional heuristics.

2.3 Real time P2P traffic identification

Timely P2P traffic identification is essential if network management is intended to quickly react to the presence of P2P traffic. To meet this requirement, [13] [14] [18] present a ML based schemes, they use the first five packets of the flow to perform

Table 3. Comparison of real time P2P traffic identification methods.

Approach	Li [18]	Erman [20]	RTI (proposed)
Why real time	Only considered the first five packets of a flow	Only considered the total number of packets in a flow	The trace interval is only 5 seconds
Accuracy	TP / (TP+FP) = 97.5%	Correctly classified bytes / total bytes = 77.5%	● (TP + TN) / #flows = 96.2% ● TP / (TP+FP) = 98.1%
Cons	The first five packets of a flow cannot be missed	Do classification only after a flow finished	When the length of a trace is changed, the thresholds must be adapted

identification, but they cannot deal with missing of some initial packets of a flow. In [20], it uses the count of the total number of a flow to do classification. The key points of each approach are summarized in Table 3, including the proposed RTI.

Chapter 3

Design Approach

In this chapter, we introduce our heuristic-based real time file sharing traffic identification (RTI) scheme that is divided into three phases: *pre-processing phase*, *P2P-using host identification phase* and *P2P file sharing traffic identification phase*, as summarized in Figure 2. The purpose of the pre-processing phase is to speed up the identification process by using port number information. In the second phase, we identify which hosts use P2P applications. Then we can filter out the traffic from non-P2P using hosts and further identify P2P file sharing traffic from P2P-using hosts. There are total nine heuristics in our proposed RTI scheme, as shown in Table 4.

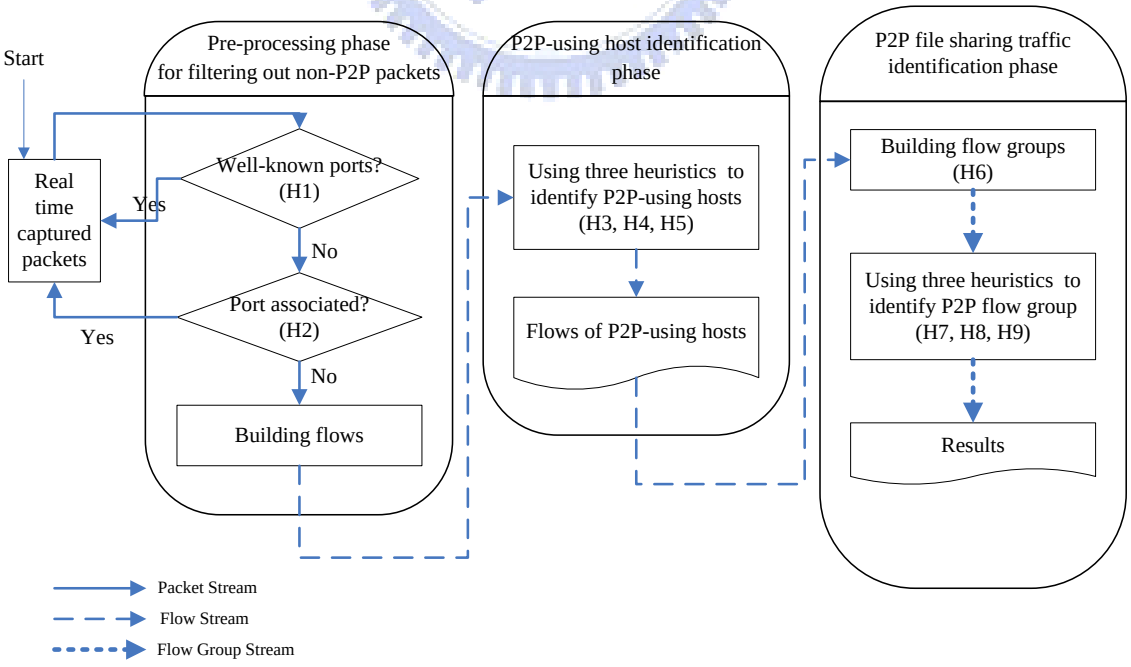


Figure 2. The flow chart of the proposed RTI scheme.

Table 4. Heuristics summary.

Heuristic	Purpose	In which phase
H1: well-known ports	Using well-known ports to identify non-P2P packets	Pre-processing phase
H2: port association	If a subsequent port number is same as or contiguous with the previous ones, then these ports are associated	
H3: the ratio between #dPort and #dIP	P2P applications usually have the same number of distinct IP and ports; this heuristic can be used to identify P2P-using hosts	P2P-using host identification phase
H4: the usage of UDP packets	P2P applications must have UDP packets	
H5: the number of distinct dIP	For file sharing purpose, P2P peers usually connect to multiple peers	
H6: building flow groups	Similar to H2. All flows in a flow group have associated port numbers	P2P file sharing traffic identification phase
H7: the ratio between #dPort and #dIP	Similar to H3, but used to identify P2P flow groups	
H8: the number of distinct dIP	Similar to H5, but used to identify P2P flow groups	
H9: the number of packet size switching (PSW) in UDP packets	P2P file sharing applications use UDP for signaling traffic, and its packet size is usually small; this heuristic is good for filtering Skype traffic	

3.1 Pre-processing phase

In this phase, we do some pre-processing to filter out some non-P2P packets, and use port numbers to speed up the identification process.

H1: well-known ports: Even though classification based on port numbers is unreliable, well-known port numbers are still suitable to identify traffic of some common applications. P2P applications usually choose port numbers from 1024-65535, so applications with all well-known ports (0-1023) excluding web ports (80, 443, etc.) are non-P2P. Note that HTTP ports are not only used for web surfing but also for some P2P applications.

H2: port association: In general, when a session of an application needs to build multi-connections, the kernel assigns a continuous ports range for each application, even for applications using randomized port numbers. We maintain the hosts and their listening port numbers in a table called *Ports Association Table* (PAT), in which each item is a two-tuple <IP, port>. When a packet arrivals, if the source or destination IP address can be found in the PAT and source or destination port numbers are contiguous or same as that in the PAT, then the packet belongs to a non-P2P application. To update the PAT, when a packet arrivals, if the source port is a well-known port, then add the destination IP and port into the PAT, and vice versa. In [10], we know that port association can speed up the identification process and make it more accurate. The port association algorithm in pseudo code is given in Figure 3. The time complexity of this algorithm is *O(#packets)*.

Building flows: After applying the above two heuristics, we filter out the packet whose *nonP2PFlag* is marked false. And we build flow by five-tuple: source IP address (sIP), source port (sPort), destination IP address (dIP), destination port (dPort), and transport layer protocol.

Port Association Algorithm

Input: all packets in the trace

Output: packets with *nonP2PFlag* = false

Alogorithm:

```
1:  boolean nonP2PFlag;            // true: non-P2P
2:  for ( each packet )
3:      if ( both sPort and dPort are well-known ports )
4:          return nonP2PFlag = true
5:      else if ( sPort is well-known port )
6:          Add dIP and dPort into PAT    //PAT: Ports Association Table
7:          return nonP2PFlag = true
8:      else if ( dPort is well-known port )
9:          Add sIP and sPort into PAT
10:         return nonP2PFlag = true
11:     else
12:         if ( sIP in PAT ) and ( sPort is contiguous or same as that in
PAT)
13:             return nonP2PFlag = true
14:         else if ( dIP in PAT ) and ( dPort is contiguous or same as that in
PAT)
15:             return nonP2PFlag = true
16:         else
17:             return nonP2PFlag = false
18:  end for
```

Figure 3. Port association algorithm in pseudo code.

3.2 P2P-using host identification phase

We use three heuristics to find out P2P-using hosts in this phase. The proposed heuristics include some thresholds which were derived empirically through experiments based on a number of traces. The P2P-using host identification algorithm in pseudo code is given in Figure 4. The time complexity of this algorithm is *O(#hosts · #flows).*

P2P Host Identification Algorithm

Input: the packets not filtered in the pre-processing phase

Output: all flows of the P2P-usimg hosts

Alogorithm:

```
1:  for ( each host h )
2:      Allflows = get all flows of host h
3:      while ( Allflows.readLine != null )
4:          #dIP = the number of distinct destination IP addresses
5:          #dPort = the number of distinct destination port numbers
6:          ratio = #dPort/#dIP
7:          if ( there is a UDP packet in Allflows )
8:              UDPFlag = 1;
9:      end while
10:     if ( ( ratio >= 0.85 ) && ( UDPFlag = =1 ) && ( #dIP > 9 ) )
11:         h = P2P-Using host
12: end for
```

Figure 4. P2P-using host identification algorithm in pseudo code.

H3: the ratio between #dPort and #dIP: As indicated by [7][15] and Figure 1, P2P peers usually maintain only one connection to other peers, which means that each host has the same number of distinct destination IP addresses (#dIP) and number of distinct ports (#dPort) connected to it. In our RTI scheme, if the ratio between #dPort and #dIP from a certain host is less than 0.85, the host is considered as a non-P2P host.

H4: the usage of UDP packets: Most P2P file sharing applications use UDP to find a peer neighbor or share files with a peer. If there is no UDP packet, the host is considered as a non-P2P host.

H5: the number of distinct dIP: Non-P2P traffic typically uses multiple connections to one server. If the number of distinct destination IP addresses is less than or equal to 9, the host is considered as non-P2P host.

3.3 P2P file sharing traffic identification phase

This phase uses four heuristics to identify P2P file sharing traffic of the P2P-using hosts which were identified in the previous phase. Instead of inspecting every flow, the identification is associated with flow groups. We can identify one packet flow as long as the flow is associated with others. The P2P file sharing traffic identification algorithm in pseudo code is given in Figure 5. The time complexity of this algorithm is *O(#flowgroup ・#UDPpackets)*.

H6: building flow groups: This heuristic uses the property of port association. Figure 6 shows that an example of port locality for a specific host that its packets are separated into three groups.

The following heuristics are only concerned of UDP packets.

H7: the ratio between #dPort and #dIP: Similar to H3, if the ratio between #dPort and #dIP from a specific flow group is less than 0.85, the flow group is considered as a non-P2P flow group.

P2P File Sharing Traffic Identification Algorithm

Input: all flows of P2P-using hosts

Output: P2P file sharing traffic

Alogorithm:

```
1:   Use port association to build flow groups
2:   for ( each flowgroup fg )
3:       AllPackets = get all UDP packets of flow group fg
4:       while ( Allpackets.readLine != null )
5:           #dIP = the number of distinct destination IP addresses
6:           #dPort = the number of distinct destination port numbers
7:           ratio = #dPort/#dIP
8:           if ( || packetSize – lastPacketSize || >= 365 )
9:               #PSW ++
10:      end while
11:      if ( ( ratio >= 0.85 ) && ( #PSW < 11 ) && ( #dIP > 3 ) )
12:          all traffic in fg = P2P file sharing traffic
13:  end for
```

Figure 5. P2P file sharing traffic identification algorithm in pseudo code.

H8: the number of distinct dIP: Similar to H5, if the number of distinct destination IP addresses is less than or equal 2, the flow group is considered as a non-P2P flow group.

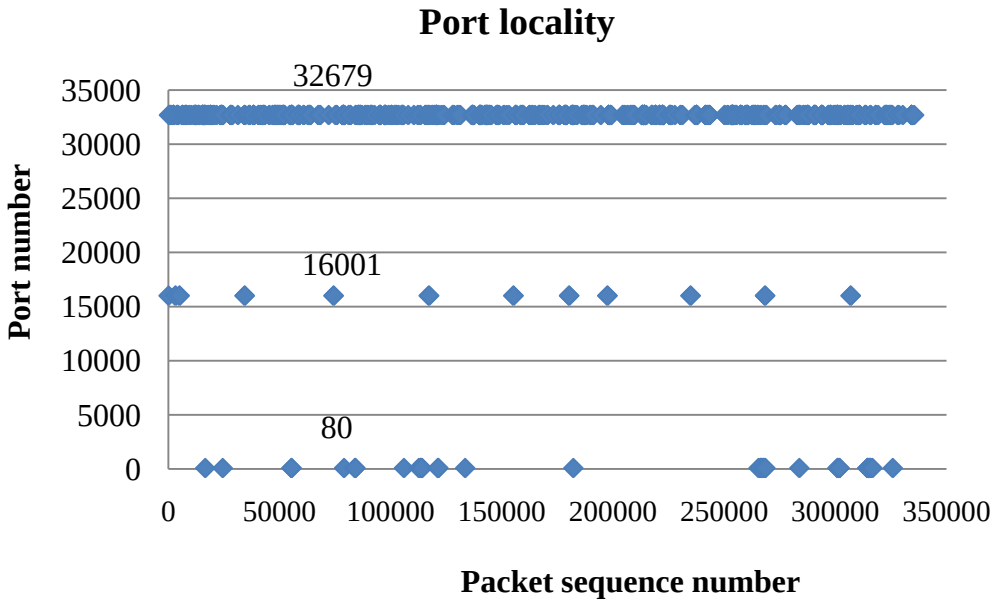


Figure 6. An example of port locality for a specific host.

H9: the number of packet size switching (PSW) in UDP packets: Packet size switching was originally proposed by [11] for identifying P2P flows, but we only use it for UDP packets. PSW is the number of packet size switching between a packet and its previous packet exceeding 365 bytes. If the number of PSW in UDP packets is greater than or equal to 11, the flow group is considered as a non-P2P flow group. P2P file sharing applications use UDP for signaling traffic, the packet size is usually small. This heuristic is good for filtering out Skype traffic, because media traffic flowed directly between Skype clients over UDP [17].

Chapter 4

Evaluation

4.1 Trace collection

Traffic traces used for experiments in this research were captured from the dormitories of the National Chiao Tung University on February 25, 2009 from 3:00:01 *a.m.* to 3:00:06 *a.m.* (t1) and 20:00:00 *p.m.* to 20:00:05 *p.m.* (t2). These traces are 5 seconds long, which was pre-classified by a payload-based classifier for verifying our P2P traffic identification results. There were 610 and 838 users, respectively. We identified P2P file sharing traffic which included BitTorrent, eDonkey, and Gnutella applications. And we also prepared a longer trace of 250 seconds on February 25, 2009 from 3:00:01 *a.m.* to 3:04:11 *a.m.* (t3) for John [9], another heuristic-based scheme, for comparison.

The information we were concerned on packets are source IPs and ports, destination IPs and ports, transport layer protocol, and packet length. These data can be found in the header easily, and we did not inspect any payload.

4.2 Performance metrics

Two performance metrics were used to evaluate the effectiveness of our scheme. They are *Accuracy* = *(TP + TN)* / *N* and *False Positive Rate* (FPRate) = *FP* / *(FP + TN)* [12], where *TP* represents the number of correctly identified samples of P2P, *TN* represents the number of correctly identified samples (packets, flows, or flow groups)

of non-P2P, *FP* represents the number of falsely identified samples that belong to P2P, and *N* represents the total number of samples, which equals to *TP+TN+FP+FN*.

4.3 Results of P2P-using host identification

The accuracy and FPRate obtained by applying different combinations of H3 ~ H5 are presented in Figures 7 and 8. When the threshold of H3 was set to 0.85, we got the highest accuracy of 92.295% and the FPRate of 6.579% in the host level. Although using H4 did not improve the accuracy much, there is still about 2% difference. For trace t2, the accuracy is 91.050% and FPRate is 8.918%, which are close to the results of trace t1. Therefore, the proposed thresholds and traces are independent.

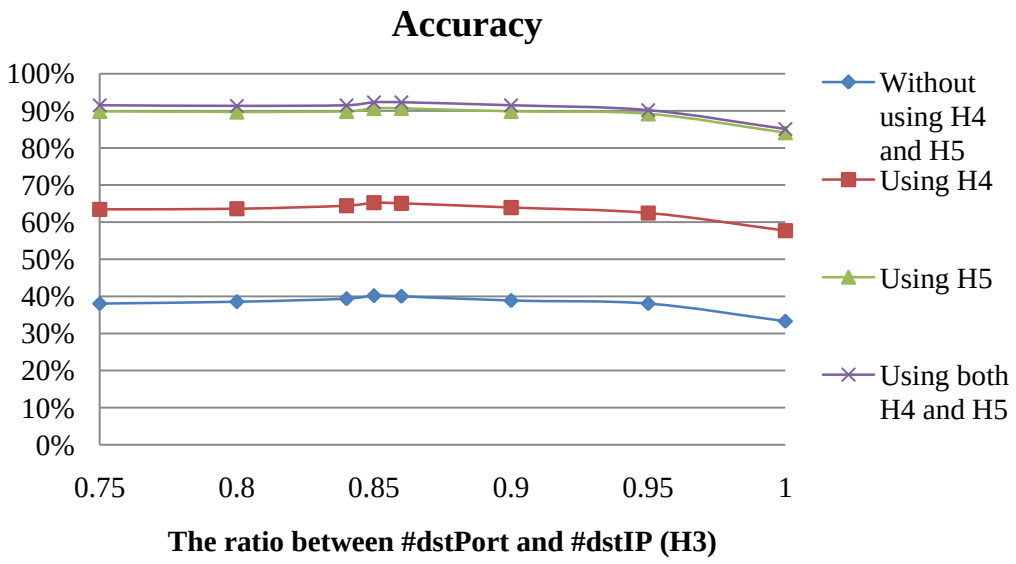


Figure 7. The accuracy of P2P-using host identification phase.

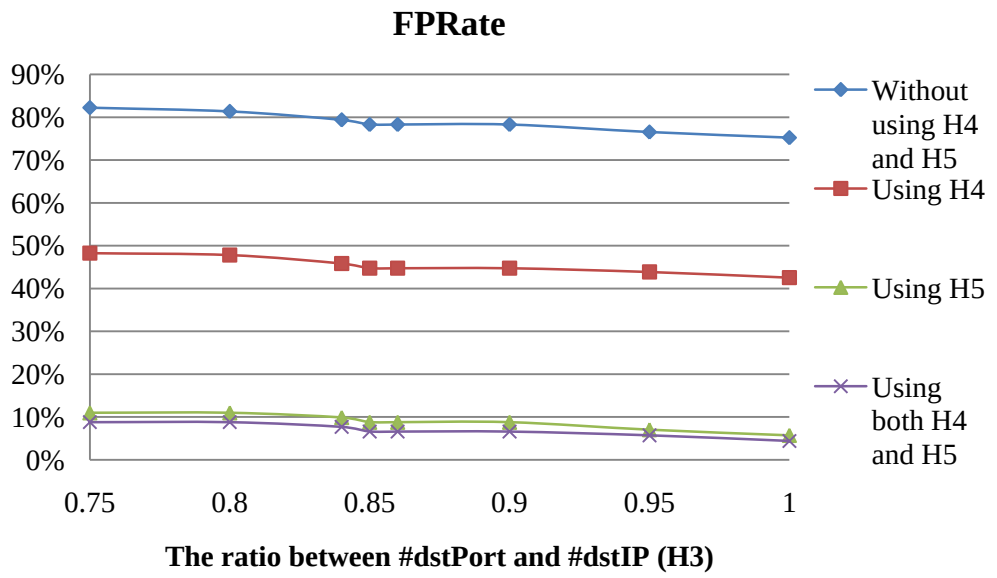


Figure 8. The FPRate of P2P-using host identification phase.

4.4 Results of P2P file sharing traffic identification

The results of P2P traffic identification are shown in Figures 9 and 10. When the threshold of H7 was set to 0.85, we got the highest accuracy of 98.288% and FPRate of 1.442% in the flow group level. For trace t2, the accuracy is 97.114% and FPRate is 2.286% in the flow group level.

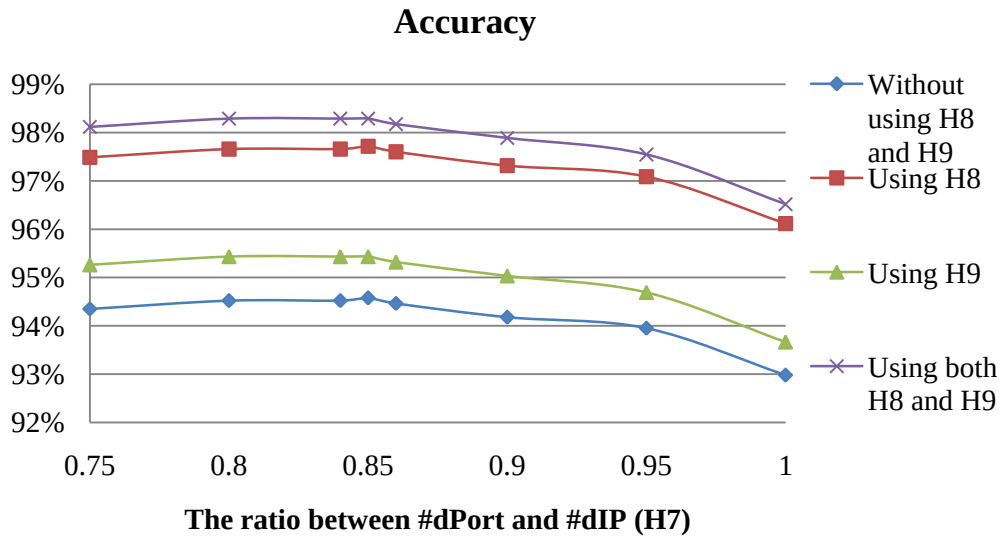


Figure 9. The accuracy of P2P traffic identification phase.

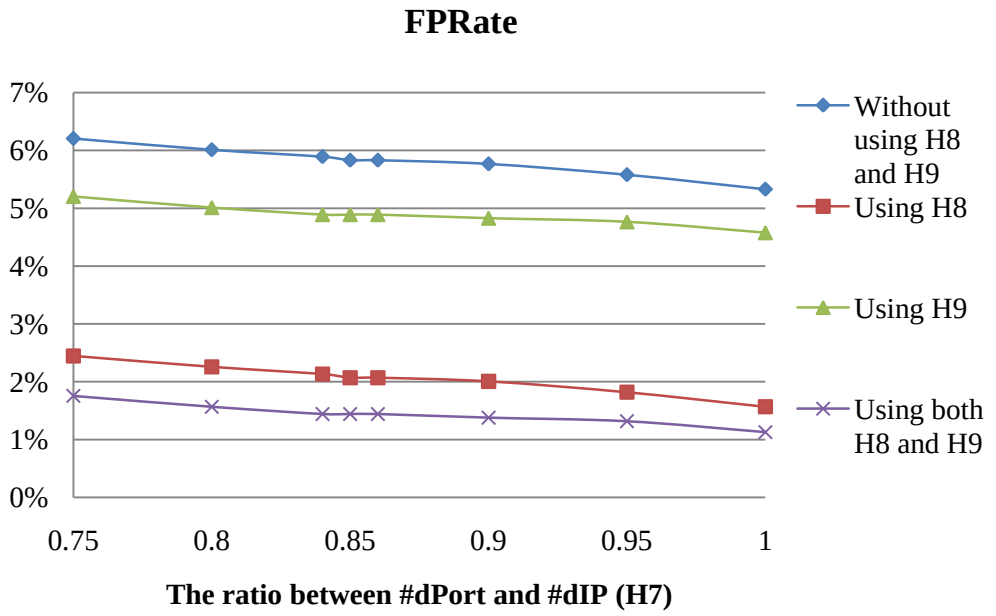


Figure 10. The FPRate of P2P traffic identification phase.

4.5 Compared to existing approaches

The overall results are presented in Figure 11, which involves three real traces. For our RTI using trace t1, the accuracy is 96.19% and the FPRate is 3.5% in the flow level. For our RTI using trace t2, the accuracy is 95.262% and the FPRate is 5.549% in the flow level. For Perenyi [8] using trace t1, the accuracy is 70.95% and the FPRate is 81.77% in the flow level. For John [9] using trace t1, the accuracy is 64.76% and the FPRate is 74.19% in the flow level. Note that the accuracy of John [9] is worse than that of Perenyi [8] is because that the duration of t1 is too short for John's heuristics, particularly the heuristic of *IP/port pairs*, as shown in Table 2.

We also implemented John [9] using t3, the accuracy is 82.824% and the FPRate is 48.647% in the flow level. The results are better than those using t1. However, its FRPate is still high. This is due to their thresholds and heuristics are not suited for our traces. In Table 5, we show the performance evaluation results of different approaches using trace t1. Our proposed RTI has the best performance.

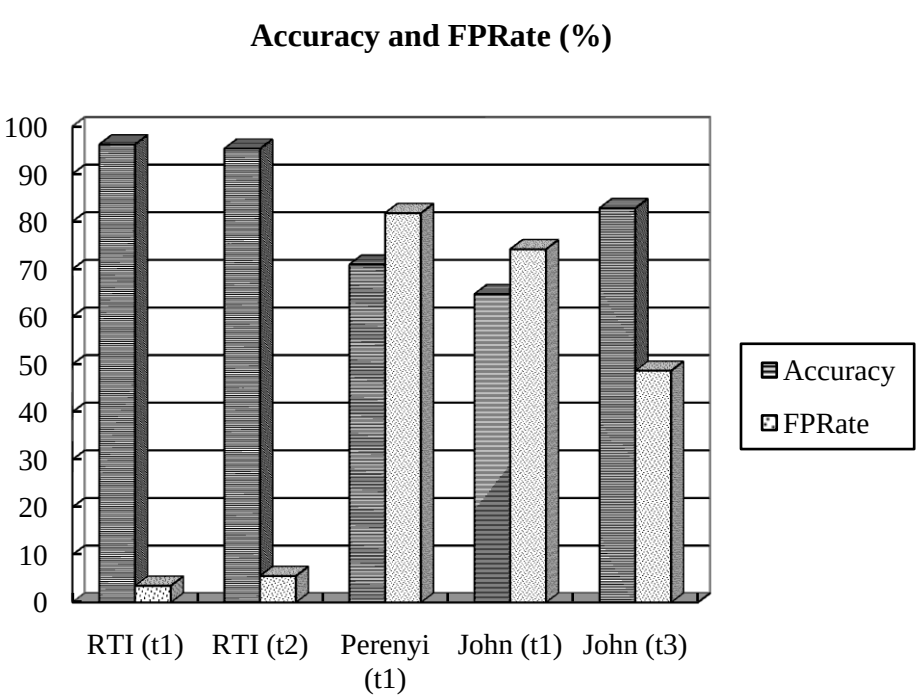


Figure 11. The accuracy and FPRate of our scheme in comparison with those of existing schemes.

Table 5. The performance evaluation results of different approaches using trace t1.

Approach	**TP / (TP + FN)**	**FN / (TP + FN)**
	FP / (FP + TN)	**TN / (FP + TN)**
RTI (proposed)	96.02%	3.98%
	3.50%	96.50%
Perenyi [8]	99.78%	0.22%
	81.77%	18.23%
John [9]	86.07%	13.93%
	74.19%	25.81%

Chapter 5

Conclusion and Future Work

5.1 Concluding remarks

In this thesis, we have presented three phases with nine heuristics to identify P2P file sharing traffic in real time. Our RTI method operates at three levels: (1) the packet level: using well-known port numbers to filter non-P2P packets, (2) the host level: finding out which host has used P2P file sharing applications, (3) the flow group level: identifying P2P file sharing traffic from the P2P-using hosts. These heuristics derived from the behaviors of P2P applications and port numbers information. We have applied our RTI method to real traces without accessing any packet payload information. Experimental results have shown that the proposed RTI had high accuracy of 96.2% and low false positive rate (FPRate) of 3.5% by using only 5 seconds of a real trace. This means the proposed RTI can identify P2P file sharing traffic in real time to facilitate network management for dealing with the problems of internet piracy and unreasonable utilization of network resources.

5.2 Future work

In our RTI scheme, we only considered the issue of identifying P2P file sharing traffic. Our future work will focus on P2P applications classification to identify a specific P2P application (e.q., BitTorrent or eMule, etc.) for achieving more effective network management.

References

[1] Subhabrata Sen , Jia Wang, Analyzing peer-to-peer traffic across large networks, *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, November 06-08, 2002, Marseille, France.

[2] “BitTorrent,” [Online]. Available: http://www.bittorrent.com/.

[3] “eMule,” [Online]. Available: http://www.cs.huji.ac.il/labs/danss/presentations/emule.pdf.

[4] “Internet Assigned Numbers Authority (IANA),” [Online]. Available: http://www.iana.org/assignments/port-numbers.

[5] S. Subhabrata, S. Oliver and D. Wang, “Accurate, scalable in-network identification of P2P traffic using application signatures,” in *Proceedings of Thirteenth International World Wide Web Conference*, pp. 512-521, 2004.

[6] Thuy T. T. Nguyen, Grenville Armitage, "A survey of techniques for internet traffic classification using machine learning," in *IEEE Communications Surveys & Tutorials*, vol. 10, no. 4, Mar 2008, pp. 56-76

[7] T. Karagiannis, A. Broido, M. Faloutsos, and K. Claffy, “Transport layer identification of p2p traffic,” in *Proceedings of the 4th ACM Conference on Internet Measurement*, 2004, pp. 121-134.

[8] M. Perenyi, D. Trang Dinh, A. Gefferth, and S. Molnar, “Identification and analysis of peer-to-peer traffic,” in *Journal of Communications*, vol. 1, no. 7, 2006, pp. 36–46.

[9] W. John and S. Tafvelin, “Heuristics to Classify Internet Backbone Traffic based on Connection Patterns,” In *ICOIN ’08: Proceedings of the 22nd International Conference on Information Networking*, January 2008, pp.1-5.

[10] Y.D. Lin, C.N. Lu, Y.C. Lai, W.H. Peng, P.C. Lin, “Application classification using packet size distribution and port association,” in *Journal of Network and Computer Applications*, 2009

[11] F. G. Chou, "P2P Flow Identification," Master Thesis, National Taiwan University of Science and Technology, Taiwan, 2006.

[12] http://en.wikipedia.org/wiki/Receiver_operating_characteristic

[13] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian, “Traffic classification on the fly,” in *ACM Special Interest Group on Data Communication (SIGCOMM) Computer Communication Review*, vol. 36, no. 2, 2006.

[14] L. Bernaille, R. Teixeira, and K. Salamatian, "Early Application Identification," in *International Conference On Emerging Networking Experiments And Technologies*, no. 6, December 2006.

[15] T Karagiannis, K Papagiannaki, and M Faloutsos, "BLINC: Multilevel traffic classification in the dark, " *Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, August 22-26, 2005, Philadelphia, Pennsylvania, USA

[16] “Gnutella hosts,” [Online]. Available: http://www.gnutellahosts.com.

[17] S. Baset and H. Schulzrinne, "An analysis of the skype peer-to-peer Internet telephony protocol," in *Columbia University Technical Report CUCS-039-04*, September 2004, pp.1-11.

[18] Jun Li, Shunyi Zhang, Yanqing Lu, and Junrong Yan, “Real-time P2P traffic identification,” in *Global Telecommunications Conference*, Nov 2008, pp.1-5.

[19] A. Madhukar and C. Williamson, “A Longitudinal Study of P2P Traffic Identifiaction,” in *MASCOT’06*, August 2006, pp. 179-188.

[20] Jeffrey Erman, Anirban Mahanti, Martin Arlitt, Ira Cohen, and Carey Williamson, “Offline/realtime traffic classification using semi-supervised

learning," in *26th International Symposium on Computer Performance, Modeling, Measurements, and Evaluation*, Volume 64, Issues 9-12, October 2007, pp. 1194-1213.