A Hybrid Approach for Accurate Application Traffic Identification

Conference Paper · January 2006

DOI: 10.1109/E2EMON.2006.1651273 · Source: DBLP

5 authors, including:

Young Jun Won
Catholic Kwandong University
61 PUBLICATIONS **1,141** CITATIONS

SEE PROFILE

Byungchul Park
University of Toronto
38 PUBLICATIONS **551** CITATIONS

SEE PROFILE

Hong-Taek Ju
Keimyung University
49 PUBLICATIONS **613** CITATIONS

SEE PROFILE

Myung-Sup Kim
Korea University
140 PUBLICATIONS **1,955** CITATIONS

SEE PROFILE

A Hybrid Approach for Accurate Application Traffic Identification

Young J. Won[1], Byung-Chul Park[1], Hong-Taek Ju[2], Myung-Sup Kim[3] and James W. Hong[1]
[1]Dept. of Computer Science and Engineering, POSTECH
{yjwon, fates, jwkhong}@postech.ac.kr
[2]Dept. of Computer Engineering, Keimyung University
juht@kmu.ac.kr
[3]Dept. of Electrical and Computer Engineering, University of Toronto
myungsup.kim@utoronto.ca

***Abstract*—The traffic dynamics of the Internet's dominant applications, such as peer-to-peer and multimedia, worsen the accuracy of the existing application traffic identification. There is a strong need for both practical and reliable identification methods with proof of accuracy. This paper proposes a hybrid approach of signature matching and session behavior mapping methods for accurate application traffic identification. In particular, the paper explores a priority-based signature matching scheme on early packet samples to replace conventional signature matching. It then uses session relationships to identify application traffic from the remaining, unidentified traffic. In validation, we present the accuracy analysis of applications using the Port Dependency Ratio (PDR) method for simulated traffic as well as real traffic.**

Keywords – Internet Traffic Monitoring, Application Traffic Identification, Signature Mapping, Session Behavior Mapping.

I. INTRODUCTION

Application traffic identification describes a procedure that determines the origin application of traffic in the unit of packet or flow. It is an important step toward providing informative snapshot of networks, fulfilling the prerequisite for QoS, and designing profitable billing policy for ISPs. The difficulty in application traffic identification has become more evident as we are witnessing a major shift change from traditional traffic (e.g., web, e-mail, ftp, telnet) to peer-to-peer (P2P) and multimedia traffic in the current Internet. The traffic dynamics of such applications deteriorate the accuracy credibility of previous identification methods, such as well-known port matching. Typical P2P applications employ ephemeral port allocation and multiple sessions during simultaneous communication with multiple peers. The increase of HTTP encapsulated traffic volume is also problematic to correct identification of application. The focus of this paper is to provide an accurate identification method for real-world networks within some boundary of practicality, even if it infers that the exhaustive search of signature or port information is unavoidable. Our work strives to provide a consistent and reliable identification method at all times and at least suggest a good starting point for network administrators to investigate in case of sudden changes in traffic dynamics.

This paper proposes a hybrid approach of signature matching and flow-based session behavior mapping methods. The priority-based signature matching scheme on limited packet samples was developed to replace the conventional signature matching. In order to verify the accuracy of the proposed approach, the manipulated traffic data is provided for testing along with the real network traffic. The concept of Port Dependency Ratio (PDR) is also introduced as an accuracy measure.

The structure of this paper is as follows. Section II described in detail a survey and categorization of application traffic identification algorithms. The terminologies used in this paper are defined in Section III. Section IV describes the proposed hybrid approach. The validation of our new hybrid approach is presented in Section V. Finally, concluding remarks are given and possible future work is discussed in Section VI.

II. RELATED WORK

This section provides an overview of application traffic identification methods and classifies them into the following categories: Session-based, Content-based, and Constraint-based Traffic Identification as illustrated in TABLE I. Discussion on shortcomings and assets for each category are supplied for further investigation. The session-based and Content-based methods rely on the exhaustive search of applications in advance and require frequent updates of port or signature information to maintain the high accuracy. The Constraint-based method is a more flexible approach that requires less of prior knowledge of target applications. However, many of these attempts have yet to claim that their results are 100% accurate; this is still an open problem for the research community.

A. Session-based Traffic Identification Methods

The well-known port matching method is based on simple matching of ports with the well-known port list, which can be obtained from exhaustive search or IANA [1]. For example, most web traffic is bound with port 80 or 8080. Many commercial traffic identification products still rely on this method because it is the simplest and most effective in detecting traditional applications (e.g., web, mail, ftp, telnet) Moore *et al.* [2] argue that port matching is no more accurate than 50 ~ 70% in the current Internet traffic. This increase of uncertainty corresponds to more deployments of firewalls, VPNs, and ephemeral port allocations by P2P applications.

This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment)" (IITA-2005- C1090-0501-0018).

BLINC [3], developed by Intel Research at Cambridge and UC Riverside, focuses on the session pattern of host and represents the patterns using graphs. The key advantage of BLINC is flexibility; no additional information of application, such as port, is required. Using a given set of pattern models, the authors claim that BLINC can identify application traffic with more than 90% accuracy. However, the level of identification results by BLINC sits apart from the expected output in this paper; the exact application name should be given with profound evidence, such as a limited portion of packet's payload. Identification by BLINC is done in a much broader sense.

Flow Relationship Map (FRM) [4] handles each distinct flow - a collection record of packets' header information - rather than packets themselves. This is based on enhanced port and session behavior mapping strategy which was developed by POSTECH in 2004. The idea of grouping flows using session patterns is quite similar to BLINC; however, port information is used to decide the actual name of application. This method is used as the first half of our hybrid approach.

B. Content-based Traffic Identification Methods

The first method, protocol inspection, extracts specific strings of information, such as port number and IP address, by parsing packet payload. SM-MON [5] and mmdump [6] apply such techniques to the control packet payloads to detect any ephemeral port number in negotiation for streaming data. The drawback is that even a tiny bit of change in the control packet (e.g., version update) or encryption of packet payload can deviate the result of such systems. In fact, this technique is rarely effective in today's networks.

In the second method, signature matching, a portion of payload data that is static, unique, and distinguishable is examined for all applications, regardless of their protocol. It is indicated as the signature of application. By comparing every packet payload with pre-determined signatures, this method identifies application traffic correctly. In this paper, we avoid searching every packet payload for signature. More details will be given on this in the upcoming section. The protocol matching method also shares a very similar concept of signature matching; however, we need to be aware of the complete protocol format. A monitoring tool, like Ethereal [7], offers the protocol matching functionality.

C. Constraint-based Traffic Identification Methods

This is a sub-category of session-based identification. Methods which fall under this category do not require any application-level protocol information. These methods borrow the concepts generally used in the area of statistics. With growing concerns of security and privacy issues, the methods here are worthwhile to study; yet, it is too early to apply these techniques into real-world networks.

Classification techniques, such as supervised machine learning [8] or statistical signature [9], group flows to a predetermined number of clusters according to the following constraints: Flow duration, average packet size of a flow, packet inter-arrival time and more. In addition, Wright *et al.* proposed the techniques using Hidden Markov Chain Profiling for building HMM profiles for applications using constraints namely, packet size and arrival time only [10]. These two are the only available constraints in the encrypted traffic data. Authors claim that this technique performs a surprising well in terms of accuracy. However, the effectiveness on more complex traffic (e.g., P2P file sharing) is yet to be verified.

TABLE I. APPLICATION TRAFFIC IDENTIFICATION METHODS

Category	Identification Method	Accuracy	Vulnerable to payload encryption	Cost of Operation	Exhaustive Searching	Applicability
Session-based Approach	Well-known Port Matching	Medium	No	Low	Yes	Practical
	Session Behavior Modeling	Low	No	Medium	No	Experimental
	Port + Session Behavior (e.g. FRM)	Medium	No	Low	Yes, Port Information	Practical
Content-based Approach	Protocol Inspection	Medium	Yes	High	Yes	Practical but very limited
	Protocol Matching	Medium /High	Yes	High	Yes	Practical
	Signature Matching	Medium /High	Yes	High	Yes	Practical
Constraint-based Approach	Supervised Machine Learning	Unknown	No	Unknown	No	Experimental
	Statistical Signature-based	Unknown	No	Unknown	No	Experimental
	HMM Profiling	Unknown	No	Unknown	No	Experimental
Hybrid	Signature Matching + Session (flow) Pattern	High	Yes	Medium	Yes	Practical

III. TERMINOLOGIES

We define an *application* as network-based software that uses a set of identical transport layer or above protocols (proprietary or open) to assemble the overall service requirement. This definition removes any ambiguity regarding many programs using multiple application-layer protocols. For example, a few P2P file-sharing applications, like KaZaA [11], uses its proprietary protocol for signaling and HTTP protocol for its service completion (i.e., file transfer). Both of these protocol traffic should be treated as from KaZaA. Moreover, MSN Messenger [12] provides a communication service through its proprietary protocols (i.e., instant messaging, voice chatting). It also simultaneously generates the HTTP traffic via the ad banner embedded in its chatting windows which is not directly related to any communication use. Therefore, whenever MSN Messenger is in use, the question might arise as to whether or not the traffic belongs to MSN Messenger. To our perspective, this traffic should be treated equally as from MSN Messenger if possible, not HTTP. Both eDonkey2000 [13] and the Korean version of eDonkey2000 called Pruna [14] also share the similar characteristics with MSN Messenger. The SIP protocol [15], which can be used in various communication applications, is another excellent example here.

In this paper, a *signature* refers to a pattern of hexadecimal digits or specific strings that are present in the payload of packet. However, the definition of signature could vary depending on what information is taken into consideration in traffic identification process. We divide the signature into two categories: packet signature and behavior signature. A packet

signature includes port numbers, protocol, the pattern present in hexadecimal digits or specific strings in the payload, and/or protocol flag fields used by the application. A behavior signature consists of the connection patterns, the measure of packet burst period (or inter-packet generation time), the number of protocols in use, and etc. Constraint-based identification methods in the previous section make use of signatures in this category. The application of certain traffic is decided by matching previously found signature or combinations of signatures.

IV. PROPOSED HYBRID APPROACH

The flow-based session behavior property, as applied in various previous research (e.g., FRM, IPTP [16], BLINC) was a rational choice to consider when identifying application traffic. Yet the goal of the paper mainly focuses on identification accuracy; we decided to remove any ambiguity in adapting the idea of session behavior mapping. For example, we do not group flows sharing the single IP address; this is equivalent to low scoring policy in FRM.

In practice, signature matching rather than port matching is still believed to be the most accurate method in determining the origin application. We focused on the fact that signatures do not appear in every packet of normal use applications, unlike the Internet worm or virus. Particularly, they appear in the initial few packets of the flow or sometimes in just first few bytes in packet's payload. Further, there was a strong need for reducing the sample packet size while extracting signature; it was virtually impossible to inspect millions of packets.

Our hybrid method suggests the midpoint where we can nicely combine the following two advantages: Accuracy of signature matching and the capability of session behavior mapping to uncover the concealed application traffic.

A. Assumptions

The following assumptions were made throughout our identification process.

1. Packets occurring in the close time interval (< 1 minute) and sharing the same 5-tuple (source IP address, source port, destination IP address, destination port, and protocol) had originated from the same application.
2. Reverse packets (displacement of 5-tuple information, protocol must be the same) in the close time interval (< 1 minute) belong to the same application
3. Packets occurring in the close time interval (< 1 minute) and sharing the same source (or destination) IP address and port are originated from the same application.
4. For limited applications (e.g., passive ftp), packets belonging to the multiple sessions between the two distinct hosts (IP addresses) are originated from the same application.

Assumptions 1 and 2 follow the fundamentals of the concept of flow. The packets belonging to these categories can be treated as being from the identical application without loss of generality. Assumptions 1-3 are identical to the conditions of Property Dependency Grouping in FRM as well as IPTP, the effectiveness of these conditions was proved.

B. Identification Procedure

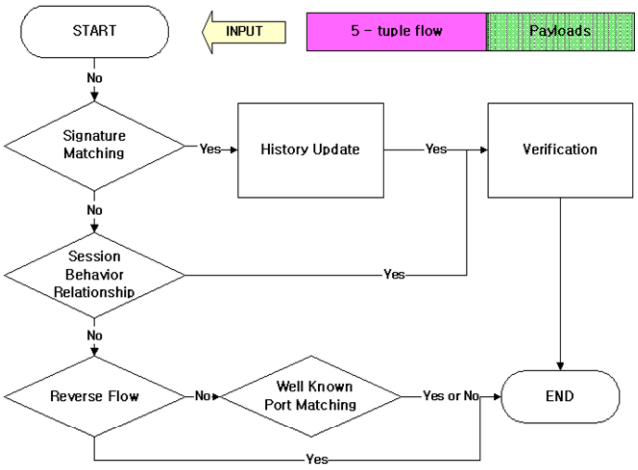

Figure 1. Application traffic identification procedure

Figure 1. illustrates the overall procedure of our application traffic identification. The input to this procedure is a slight modification of Cisco NetFlow Version 5. In addition, the payloads of initial few packets are attached to corresponding flow. Payload refers to the application layer or above portion of packet in binary. The number of attached payloads to flow is configurable.

The origin application of flow can be determined through signature matching on the attached payloads only. To minimize any false positive or negative, the priority-based signature matching method is being applied on limited number of packet payloads. Priority-based signature matching has the following policy:

- Applications with a longer matching signature have higher priority in case of conflict.
- Applications with multi-positioned signature have higher priority in case of conflict. e.g., the eDonkey traffic shares 'e3' at offset 0 and '01' at offset 5.
- 'HTTP' traffic signatures have the lowest priority.

We avoid using any short strings as signatures unless it is a part of multi-positioned signature. Many applications, especially P2P file-sharing (e.g., KaZaA) and web-storage applications, disguise themselves as web traffic using HTTP encapsulation. For such applications, the signatures are extracted independent from the HTTP protocol signature. Applying these signatures prior to the HTTP signature is crucial when distinguishing between true web traffic. At last, the Karp-Rabin algorithm [17] was implemented for string matching; whose expected runtime of KP algorithm is $O(n+m)$ where n and m refer to each length of two strings being compared.

Among the undetermined flows, we searched for any flows that shared the same source (or destination) IP address and port with those of the identified flows after the signature matching

process. The {IP, port} pair information of the identified flows were stored until no such packet had arrived in the last 2 minute interval. The {IP, IP} pair information was kept in the same passion in the history update step.

Some might argue whether the {IP, port} pair relationship is safe to assume. For example, the host running the web server can initiate various different types of flows, like simple text or streaming data, via multiple sessions sharing the same source IP address and port 80. However, if distinct signature is present in each flow, signature matching which is the first step in the whole procedure should be able to identify correctly. Also, the corresponding destination IP addresses and port numbers must be different in each flow.

The next step is to search for any reverse flows of the identified flows up to this point. They will be tagged with the matching application name. This is straight forward as in Assumption 2.

Finally, well-known port matching was applied selectively to the left-over portion of undetermined flows. Its influence to the overall identification result is designed to be as minimal as possible due to raising question of accuracy. Berkeley Port Allocation Scheme [18] is common to most UNIX systems as well as most other non-UNIX TCP/IP implementation. According to the scheme, port numbers between 0 and 1023 are system privileged; thus, many common services, such as ssh (22), telnet (23), mail (25) and more, still rely heavily on fixed port usage. Nonetheless, we cannot assume that any of the identification results from well-known port matching guarantees 100% accuracy. They are referred to as 'Suspected' portion of traffic in the paper. This is another reason why signature matching should be the first step in the procedure because it searches for trace in every flow regardless of port number.

In the verification step, we gathered statistics of the identified traffic: Port Dependency Ratio (PDR) of the major port and PDR distributions of the rest of the ports in use. The PDR provides information on how much traffic was bounded to that particular port number in the pool of flows which are already identified via signature matching. The following two interpretations have been made using this measure. The PDR specifies the classification probability of the suspected amount; in other words, it estimates how reliable the suspected amount traffic is. Secondly, the high PDR of port can be interpreted as an indicator of how accurate the found signatures are for some applications. We focused on the fact that many types of applications, including P2P, allocates the same port or ports in the close range repeatedly.

C. Signature matching on early packets

Application signatures are more reliable when they are present in control packets, most likely in the initial few packets of flow or few bytes of payload. If we determine the origin application of flow in just first few packets, then we can avoid unnecessary expensive string matching cost on upcoming packets and minimize any false positives. The following four scenarios were tested repeatedly with the identical traffic trace.

- Signature matching on the first packet payload of flow (1_flow)
- Signature matching on the first five packet payloads of flow (5_flow)
- Signature matching on the first ten packet payloads of flow (10_flow)
- Signature matching on every packet of flow

We collected the traffic trace from one of the two Internet junctions at POSTECH, a university with a user population of about 3500 people composed of students, researchers, faculty, and staff. To avoid any possible packet loss, the monitoring sensors equipped with Endace's DAG 4.3GE [19] were used to monitor 1Gbps Ethernet links. Due to storage limitation and privacy concerns, we collect packet traces (maximum of 100 bytes per packet) during 23:00 ~ 23:59, Sept. 19, 2005 – (130 million packets, 83 GB, avg. 190 Mbps, referring *Period I*)

The first three scenarios refer to 'Partial Matching' and the last one refers to 'Full Matching'. Each partial matching case is written as 'x_flow' where x is the number of payloads attached to flow. Note that no other steps other than signature matching are applied here.

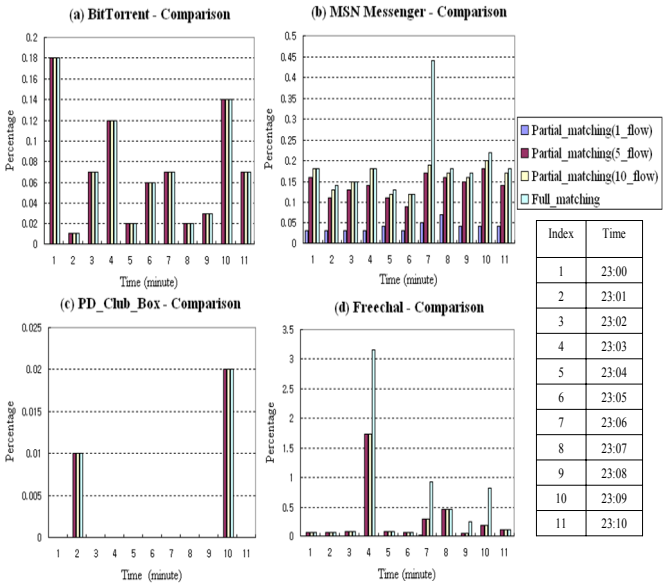

Figure 2. Byte counts for BitTorrent, Messenger, Pd Club Box, and Freechal

Figure 2. illustrates the difference between the determined byte counts in the four scenarios. The results are from the first 11 minutes of data of a one hour trace. The total determined byte counts of 5_flow and 10_flow were almost identical in all four selected applications. For most of cases, the results of full_matching were also equivalent to those of 5_flow and 10_flow. In other words, the signature was present within the first 5 packets of flow. There were sudden peaks of full_matching in Figure 2 b) and c), MSN Messenger and Freechal, respectively. For example, the total determined bytes for Freechal at 23:03 showed a difference of 2% between full_matching and the rest. This difference could be left as the undetected portion of Freechal traffic in the rest of scenarios.

The following case illustrates why we cannot trust the identification result of full_matching. Matching signature on every packet is unnecessary and could generate unreliable identification results.

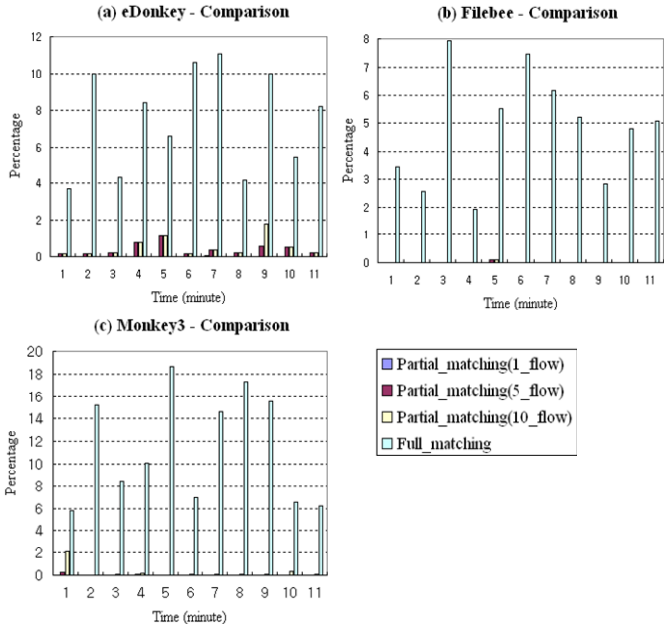

Figure 3. Byte counts for eDonkey, File bee, and Monkey3

Unlike the applications in Figure 2. , we observed the large difference of the determined byte counts between full_matching and the rest in Figure 3. Filebee [21] and Monkey3 [22] are currently other popular file sharing applications in Korea. Especially, Figure 3. b) and c) demonstrate that the identification ratio of the partial_matching scenarios is almost 0% while full_matching discovers the maximum 8% and 18% of the total traffic, respectively. There could be two possible answers to this phenomenon. The first possible answer could be that the signature was located after the first 10 packets of flow. The other could be possible false positives in the identified portion of application traffic.

TABLE II. FALSE POSITIVE RATIOS OF FULL_MATCHING

		eDonkey		File bee		Monkey3	
Index	Time	Determined Byte – MB(%)	False Positive Ratio	Determined Byte – MB(%)	False Positive Ratio	Determined Byte – MB(%)	False Positive Ratio
1	23:00	50 (3.7%)	81% +	47 (3.44%)	78% +	78 (5.78%)	62% +
2	23:01	135 (10%)	74% +	34 (2.54%)	55% +	206 (15.19%)	35% +
3	23:02	60 (4.33%)	46% +	111 (7.93%)	74% +	118 (8.44%)	59% +
4	23:03	110 (8.42%)	72 % +	24 (1.91%)	29% +	132 (10.12%)	17% +
5	23:04	82 (6.6%)	37 % +	68 (5.5%)	55% +	233 (18.66%)	unknown
6	23:05	136 (10.6%)	79 % +	95 (7.45%)	unknown	89 (6.98%)	45% +
7	23:06	136 (11.1%)	76 % +	75 (6.14%)	unknown	180 (14.62%)	56% +
8	23:07	49 (4.16%)	25 % +	62 (5.22%)	43% +	206 (17.31%)	74% +
9	23:08	118 (10%)	69 % +	33 (2.84%)	90% +	184 (15.51%)	48% +
10	23:09	62 (5.39%)	68 % +	56 (4.82%)	80% +	76 (6.53%)	46% +

We were able to determine the false positive amount of traffic in TABLE II. For example, the first entry for eDonkey indicates about 50 MB (3%) of the total traffic and at least 81% false positive ratio. It infers that more than 40 MB out of 50 MB does not actually belong to eDonkey but some other applications.

Therefore, we chose the signature matching scheme on the first five packets for further study. One interesting observation is that the byte counts between 5_flow and 10_flow were almost identical in all the graphs shown in Figure 2. and Figure 3. It is worthwhile to focus on the trade-off between early signature presence and the rest.

V. VALIDATION

The section evaluates the accuracy of the hybrid approach using both simulated traffic and real traffic.

A. Isolated Application Traffic vs. Synthetic Traffic Mix

The fundamental difficulty in proving the correctness of traffic identification is to obtain a suitable traffic data set for testing. In order to measure the correctness of algorithm, we need to be fully aware of all the applications which occupy the data set; ironically, it is the ultimate goal of this research. It is also difficult to attain the dynamics of real-traffic in the simulation environment.

This section provides an early analysis of accuracy of the proposed algorithm by creating a reliable data set. We have selected the seven representative applications based on their popularity and traffic complexity, and collected the traffic trace (every minute basis – one packet trace file per one minute) while they were running independently on the single host: BitTorrent, eDonkey2000, Freechal, KaZaA, Monkey3, MSN messenger, and PD Club Box. Each application traffic trace contains 16 minutes of data. In other words, each trace solely consists of the corresponding application only. The proposed identification algorithm ran on each traffic trace separately and identified them up to 99%. This is referred to as 'Single'.

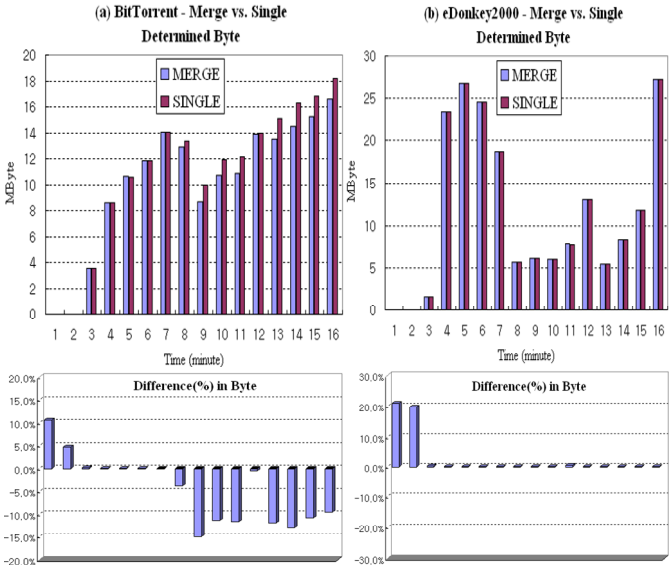

Figure 4. Byte counts for BitTorrent and eDonkey2000 – Merge vs. Single

A synthetic traffic mix, called 'Merge', can be generated by shuffling all the traffic traces of the seven applications. In the shuffling process, the packet arrival sequence of each application must be preserved in order to maintain the consistency in the data set. We ran our identification algorithm on the newly created 16 merge files. Theoretically, the amount of identified byte for each application should match between the two data sets – 'Single' and 'Merge'.

The first two minutes for BitTorrent and eDonkey2000, as seen in Figure 4. illustrate that we are counting slightly more traffic in the hybrid identification process testing with the 'Merge' data set. This phenomenon relates to unsorted packet sequence in the shuffling process rather than the false positives by our identification method. If the percentage difference of the determined bytes is positive (negative), then there are larger (less) byte counts when testing with the 'Merge' data set. eDonkey shows the perfect match later in the graph. Our algorithm identifies the less of BitTorrent traffic in overall. However, these missing or gaining amounts of traffic are so small, almost negligible. It is likewise for Freechal and KaZaA.

Figure 5. illustrates the relatively well matching byte counts for Monkey3, PD Club Box, and MSN Messenger. A few occurrences of sudden peaks in the difference of Figure 5 a) and b) are due to the unclear distinction of traffic between the two applications and web. Monkey3 and PD Club Box partially rely on web to initiate the search and download process. Under the definition of application in this paper, they should be counted as other than web; however, it could not be done in this case. Nonetheless, it is difficult to conclude that our identification is completely wrong because they are indeed web traffic too.

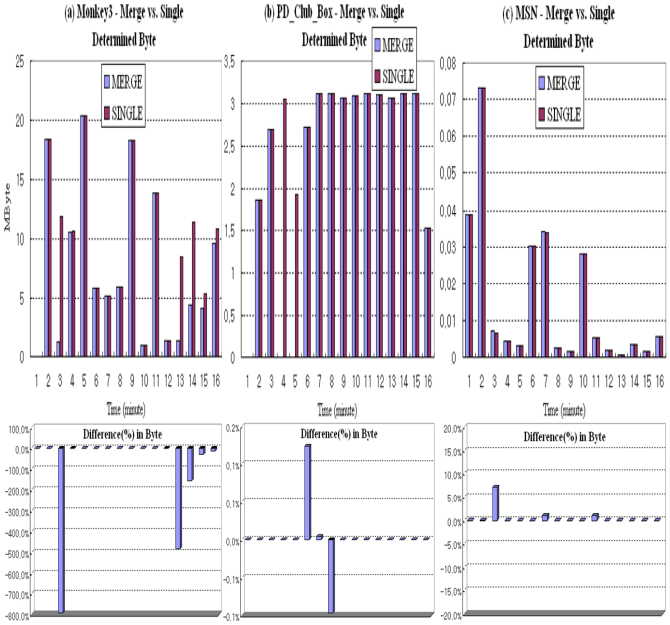

Figure 5. Byte counts for Monkey3, Pd Club box, and MSN Messenger – Merge vs. Single

Overall, the determined byte difference between the 'Merge' and 'Single' data sets is minimal and the accuracy of our algorithm is trustworthy for further analysis.

B. *Real Traffic Mix*

In this section, we provide identification results of the real-network traffic from *Period I*. TABLE III. illustrates that over 90% of traffic (in byte, the number of packets, and the number of flows) had been identified when using the hybrid method. Although the hybrid approach had more unknowns (10%), it could provide the measure of identification accuracy while FRM and any other similar methods could not.

TABLE III. FRM VS. HYBRID APPROACH

Type	Application		Byte (%)	
	FRM	Hybrid Approach	FRM	Hybrid Approach
Web	-	-	20 ~ 30 %	20 ~ 40%
P2P	eDonkey, freechal, bitTorrent, pd_club_box, and more	eDonkey, monkey3, freechal, bitTorrent, pd_club_box, file bee, and more	60 %	50+ %
Messenger	NateOn, MSN messenger and more	NateOn, MSN messenger and more	1 %	1 %
Others	ftp, mail, MS_dir_service, idisk, NetBios, and more	ftp, mail, MS_dir_service, idisk, NetBios, and more	15 %	10 %
Unknowns	N/A	N/A	5 %	10 %

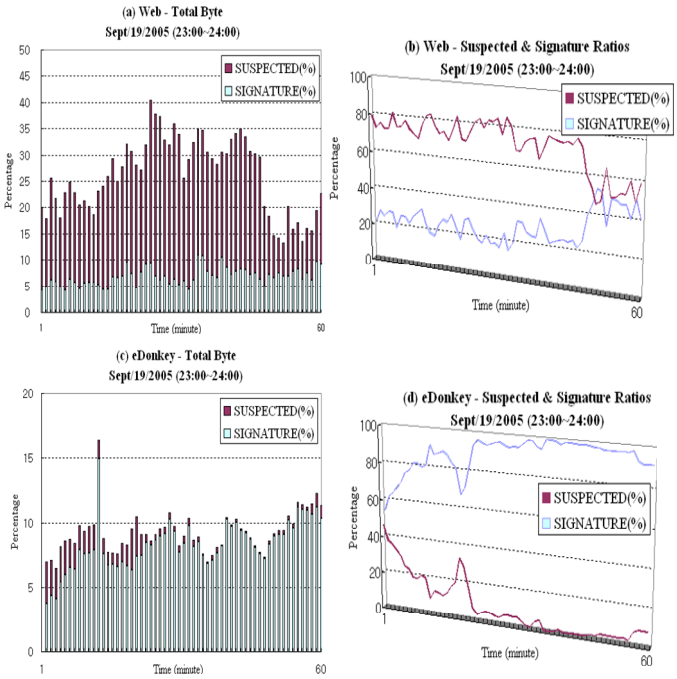

Figure 6. Identification results – Web and eDonkey

Figure 6. a) and c) illustrate the total occupied bytes of web and eDonkey respectively; however, these figures include the suspected portion of traffic. During the monitoring period, web and eDonkey occupied 20 ~ 40% and 10% of the total traffic, respectively. We made an interesting observation that the ratio of suspected traffic amount to the total identified traffic amount is declining as in Figure 6. b) and d). For web, the ratio of suspected traffic bytes reaches down to 50% from almost

80% in the beginning. eDonkey showed an even more dramatic decrease, almost reaching down to 0; in other words, the determined eDonkey traffic at this moment can guarantee 100% accuracy. This is a clear indication that the proposed algorithm can recover a possible false positive in the start phase and promise more accurate identification result with a longer monitoring period. Once we encounter the initial packets of flow, containing signature, then we can improve the accuracy and rely less on the port matching method.

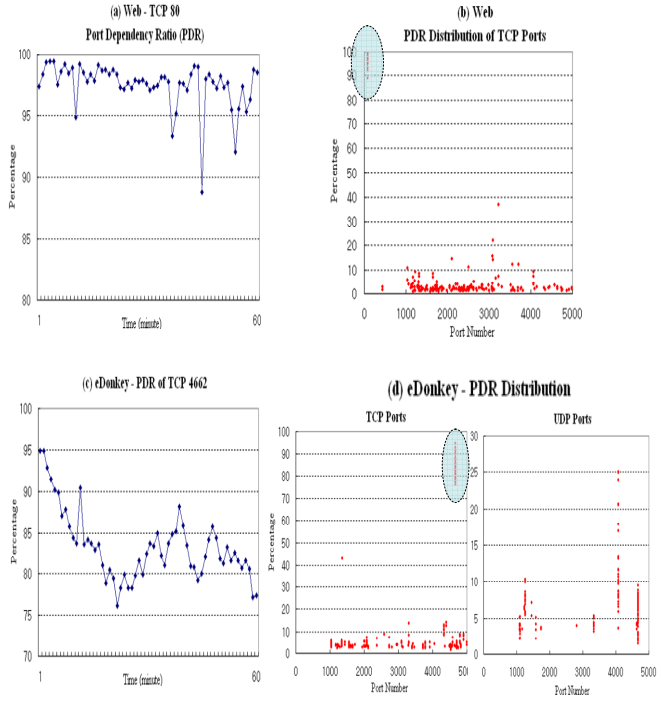

Figure 7. PDR distribution of port – Web and eDonkey

It is also possible to estimate how accurate the suspected amount traffic of each application is. Figure 7. illustrates the major port dependency ratios (PDR) and the PDR distribution graphs of web and eDonkey. The PDR, in percentage, indicates how much traffic is bounded to that particular port number in the pool of flows which are identified via signature matching. Any port number follows a Zipf like distribution; it is marked as the major port and its PDR is computed.

Figure 7. a) and b) indicate the PDR of TCP port 80 over time and the rest of port distribution for web, respectively. The PDR of TCP port 80 reaches up to 99% at peak, such high ratio reassures that the IANA's port listing for web is reliable. The shaded area in Figure 7 b) indicates the TCP port 80. The suspected traffic amount of web in Figure 6. a) was also revealed by matching TCP port 80, so that it mostly likely belonged to web.

eDonkey follows a very similar trail of web as illustrated in Figure 7. c). TCP port 4662 shows the highest PDR (70 ~ 90%) among the ports in use. The frequent allocation of TCP port 4662 was also verified through exhaustive search of eDonkey. UDP port 4672 and 4073 seem to have relatively higher PDR among the UDP ports.

The determined traffic for Freechal (about 15% of the total traffic) and Monkey3 (about 10% of the total traffic) was bounded to a fixed port. In Figure 8. , the PDR of TCP port 9493 is 100% which means every flow belonging to Freechal employs TCP port 9493. We can also observe that the Monkey3's traffic depends on TCP port 8008 with high PDR.

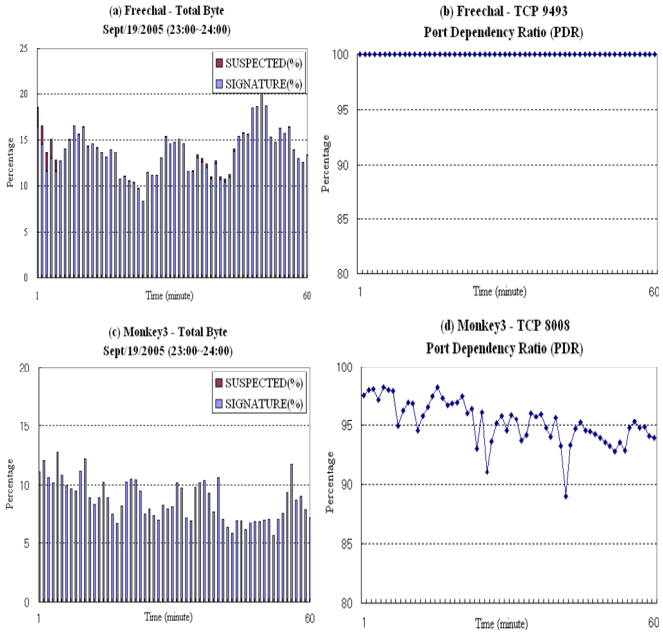

Figure 8. Identification results and PDR of major port - Freechal and Monkey3

These high PDRs can be interpreted from a different perspective. They could be an indicator of how accurate the discovered signatures are for some applications. We focused on the fact that many types of applications, including some P2P files-sharing applications, allocate the same port or ports in the close range repeatedly. For example, we knew the frequently allocated port for eDonkey in advance, TCP port 4662. If the majority of determined flows for eDonkey turn out to be using 4662, then it will be further proof for the accurate signature. However, such analysis is impossible for the applications, like BitTorrent, which have more sporadic port allocation behavior.

Figure 9. a) and c) illustrate that the determined byte for BitTorrent and Pd Club Box occupy 5% and 20% of the total traffic, respectively. As mentioned above, the three shaded regions in the PDR distribution graph for BitTorrent infer that its sporadic port allocations. There is no single major port as in the previous applications. Instead, the ports in frequent use are evenly distributed over the wider range of port number.

PD Club Box actually refers to two separate applications: PD Box and Club Box. We treated them as one because of the interoperability and identical user manual between the two. However, Club Box does not have a clear signature while PD Box does. In order to discover the Club Box traffic, we had to rely on matching TCP port 19101 for now. PD Box traffic was bounded to TCP ports within the range of 150xx as in Figure 9. d).

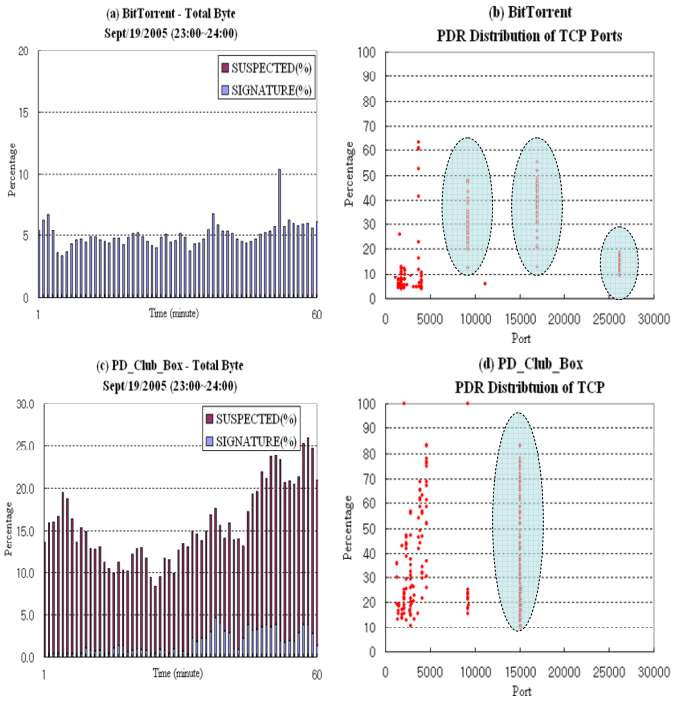

Figure 9. Identification results and PDR distributions of TCP port – BitTorrent and Pd Club Box

VI. CONCLUDING REMARKS

The traffic dynamics of the Internet's dominant applications (e.g., P2P, multimedia, web-storage) deteriorate the credibility of accuracy of previous identification approaches. In this paper, we presented a proposal for the hybrid approach of the existing identification techniques. Priority-based signature matching on early packets of flow was investigated for its applicability to deciding the origin application of flow as well as the session behavior property. Merging these two could lead to even more accurate identification ratio than single handed signature matching as well as improved efficiency. Plus, we provided an analysis example for identification accuracy. The analysis consisted of the identification results on the manipulated traffic as well as the real traffic and the interpretation of the accuracy measure variable, PDR.

In order to maintain the accuracy of the proposed algorithm, the periodic update of signature is essential. Currently, the signatures are discovered by the manual search of packets only. For efficient updates of both known and unknown signatures, the automation for signature extraction must be involved and it is currently under development as future work. This also relates to the part of our next research which is to build self-learning application traffic identification system. It will require the minimum level of user intervention. On-line deployment of the proposed identification algorithm is also under plan. This will be an important step toward providing an informative snapshot of networks.

REFERENCES

[1] IANA, IANA port number list, http://www.iana.org/assignments/port-numbers.

[2] A. Moore and K. Papagiannaki, "Toward the Accurate Identification of Network Applications", Passive and Active Measurements Workshop, Boston, MA, USA, March 31, April 1, 2005.

[3] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: Multilevel Traffic Classification in the Dark", ACM SIGCOMM, Philadelphia, PA, August 2005.

[4] Myung-Sup Kim, Young J. Won, and James Won-Ki Hong, "Application-Level Traffic Monitoring and Analysis on IP Networks", ETRI Journal, Vol.27, No.1, Feb. 2005, pp.22-42.

[5] Hun-Jeong Kang, Myung-Sup Kim, and James Won-Ki Hong, "A Method on Multimedia Service Traffic Monitoring and Analysis", LNCS 2867, 14th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM 2003), Heidelberg, Germany, October, 2003, pp. 93-105.

[6] J. E. van der Merwe, R. CÃ ¡ cere, Y-H. Ch, and Cormac J. Sreenan. "mmdump - A Tool for Monitoring Internet Multimedia Traffic", ACM Computer Communication Review, 30(4), ACM Press, 2000.

[7] Ethereal, http://www.ethereal.com.

[8] M. Roughan, S. Sen, and O. Spatscheck, "Class of Service Mapping for QoS: A Statistical Signature based Approach to IP Traffic Classification", Internet Measurement Conference, Taormina, Sicily, Italy, October 25-27, 2004.

[9] A. Moore and D. Zuev, "Internet Traffic Classification Using Bayesian Analysis Techniques", ACM SIGMETRICS, Banff, Canada, June 2005.

[10] W. Charles, M. Fabian, M. Gerald, "HMM Profiles for Network Traffic Classification (Extended Abstract)", In Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security, Washington, DC, USA, October 29, 2004.

[11] KaZaA, http://www.kazaa.com, Sharman Networks.

[12] Microsoft, MSN Messenger, http://messenger.msn.co.kr.

[13] Meta Search, eDonkey 2000, http://www.edonkey2000.com.

[14] Pruna, http://www1.pruna.com.

[15] J. Rosenberg et al., "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[16] T. Karagiannis, A. Broido, N. Brownlee, KC Claffy, and M. Faloutsos. "Is P2P dying or just hiding?", IEEE Globecom, Dallas, Texas, USA, Nov. 20 –Dec. 3, 2004.

[17] Karp-Rabin algorithm, http://www-igm.univ-mlv.fr/~lecroq/string/node5.html#SECTION0050.

[18] Berkeley Port Allocation Scheme, http://csrc.nist.gov/publications/nistpubs/800-7/node158.html.

[19] Endace, DAG 4.3GE, http://www.endace.com/dag4_3GE.htm.

[20] Pd box, http://www.pdbox.co.kr.

[21] File bee, http://www.filebee.co.kr.

[22] Wisepeer, Monkey3, http://www.monkey3.co.kr.