Application Traffic Identification Based on Remote Subnet Grouping

Jae Yoon Chung and Jian Li
Department of Computer Science and Engineering
Pohang University of Science and Technology
Pohang, Korea
{dejavu94, gunine}@postech.ac.kr

Yeongrak Choi and James Won-Ki Hong
Division of IT Convergence Engineering
Pohang University of Science and Technology
Pohang, Korea
{dkby, jwkhong}@postech.ac.kr

***Abstract*—Recently, the number of Internet applications available for use on both desktop computers and smartphones has rapidly increased. The Internet traffic generated by these applications has increased significantly as well. Network operators are required to be aware of the status of managed networks in terms of application usage. However, the application traffic observed in a network is dependent on users and the geographical location of the network. As a result, selecting applications that are expected to appear frequently in the network is required as the initial step in the generation of ground-truth traffic and extraction of classifiers. In this paper, we propose a traffic identification methodology for the initial step of the classifier generation procedure. The proposed approach is based on remote subnet grouping, which refers to the collection of the same (or similar) application traffic. We also validate the proposed methodology in terms of completeness, feasibility, and accuracy by using the traffic in a campus network.**

Keywords-component; Traffic Classification, Traffic Identification, Signature Extraction

I. INTRODUCTION

Internet traffic generated has rapidly increased owing to the proliferation of Internet applications. The diversity of Internet applications and network devices produces a complex mix of application traffic. For these reasons, analyzing Internet traffic in order to remain aware of network usage is becoming more difficult for network operators. Internet traffic classification is a significant research topic in the field of network management. It provides detailed information on network status in terms of application-level usage, protocol-level usage, abnormal traffic pattern, etc., which is not provided by current Internet protocols. Even though many traffic classification methodologies have been proposed, it is difficult to apply other methodologies to a managed network because application usage in managed networks is different from that in other networks. Consequently, exhaustive parameter settings and classifier generation are required in order to apply the other traffic classification methodologies to managed network. For example, many applications, such as eDonkey and LimeWire, which were used to validate previous research, are not frequently observed in current Internet traffic. Thus, we should select target applications based on the network traffic we want to analyze because there is no guarantee that the applications analyzed in other research will appear frequently in the network under consideration.

The initial tasks and the challenges in traffic classification research are as follows:

(1) Surveying and listing popular applications

Prior to extracting traffic classifiers, the target applications that are to be observed in network traffic should be selected. The common approach to selecting applications is based on user popularity data provided by application markets and surveys. However, these market ranking and surveys of user popularity do not guarantee that the application traffic will appear in the managed network. As a result, it is required that the application traffic in the target network that we want to analyze be identified. This analysis of application traffic as the initial task in traffic classification research offers a basic insight into the target network even though we cannot identify all of the application traffic in the network.

(2) Applying traffic classifiers

When applying traffic classifiers generated by other research or products, we are faced with unfamiliar applications that are not popular or applications that have never been observed in our network. Network operators are also unwilling to apply the traffic classifiers due to the old-updated traffic classifiers. The commercial entities usually do not publish their classifiers for business reasons; hence, it is difficult for us to modify or add new application traffic classifiers. Even in cases where traffic classifiers are available to the general public, updating hundreds of classifiers of Internet applications requires the exhaustive extraction of application traffic classifiers.

(3) Extracting application traffic classifiers

Extracting application traffic classifiers from application traffic is an exhaustive task. To collect the target application traffic exclusively, the target application is usually installed on a machine that is equipped to capture traffic. Application traffic is then intentionally generated by using the application for several minutes. From the application traffic collected, distinguishing traffic characteristics such as port number, destination IP address, and common substring are then extracted.

In this paper, we propose a methodology that identifies applications and application traffic based on remote subnet grouping. Our focus is not on traffic classification accuracy, but rather on how to begin traffic classification research and how to obtain insight into the target network without any extra information, which is to identify application traffic in the target network using collected traffic itself. We assume that the traffic in a subnet is a collection of the same type of application traffic. Grouping the same type of application traffic makes it easy for network operators to identify application traffic existing in the target network. We also define local and global classifiers that are applied to each subnet and the entire traffic, respectively.

The remainder of this paper is organized as follows. Section 2 presents related studies on network traffic classification. Section 3 describes the proposed methodology in terms of how to identify application traffic, how to extract traffic classifiers, and how to apply traffic classifiers. Evaluation results of real network traffic traces are presented in Section 4. Finally, concluding remarks are given and future work discussed in Section 5.

II. Related Work

Traffic classification research is a major research area in the network management field. The traffic classification methodology begins from the port number-based traffic classification. Even though some traffic such as HTTP, SSH, and FTP still use their well-known default port numbers, the accuracy of port-based traffic classification is less than 70% [3].

Payload-based traffic classification techniques inspect packet payloads to find byte patterns (commonly called 'Signatures') that uniquely represent the application (or protocol) [1][4][5][6]. Payload-based traffic classification is a most reliable approach if the signature is defined well. For this reason, the payload-based approach is used as the ground-truth for other traffic classification methodologies. However, one drawback of the payload-based traffic classification approach is the exhaustive process of classifier extraction. Recently, automated signature generation studies have been proposed [1][2][5]. These studies are based on the extraction of a common substring from the payload of the application traffic that we want to classify, however, the input traffic has to be generated accurately, according to each application and its functionality.

As an alternative to the payload-based traffic classification approach, connection pattern-based traffic classification approaches have been proposed [7][8]. BLINC [7] attempted to classify Internet traffic flows according to application type and without the use of port numbers and payloads. However, our research focuses on application traffic identification rather than traffic classification according to the type (protocol) of application traffic. In other words, we are focusing on which applications are existing in the network to help selecting target applications we will classify. This forms the initial step of the traffic classifier generation process.

Recently, statistics-based traffic classification approaches have also been proposed [9][10]. Applying machine learning (ML) algorithms to traffic classification results in high classification accuracy, but training ML algorithms consumes high computing resources. In addition, because of the over-fitting problem, the accuracy is in question when the same methodology is applied to different networks.

Park et al. [11] proposed a fine-grained traffic classification technique that classifies network traffic according to the functionalities in an application. This study categorizes traffic classification approaches according to the level of the traffic classification. Our methodology aims to provide 'application breakdown'. The application functional-level classification is also available if we can extract different traffic classifiers according to the functions of an application.

In this paper, we propose a traffic identification methodology as the initial step of the classifier generation procedure. The proposed approach is based on remote subnet grouping—the collecting of the same type (or similar) application traffic. We also validate the proposed methodology in terms of completeness, feasibility, and accuracy using the traffic in a campus network.

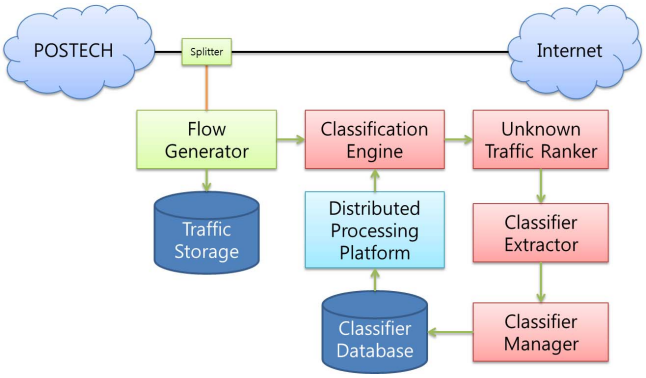

Figure 1. Architecture overview of the proposed traffic identification system.

III. Proposed Methodology

Figure 1 gives an overview of the architecture of our proposed traffic identification system. The proposed architecture is not dependent on any specific traffic monitoring system. Currently, our focus is on integrating the proposed methodology with the POSTECH Internet traffic monitoring system, which captures Internet traffic generated from both wired and Wi-Fi networks at 1 Gbps Internet junction links. The major components of the proposed traffic identification system are as follows:

- ***Unknown Traffic Ranker:*** gives the rank of remote subnets according to the volume of unclassified traffic. Because network operators analyze subsets of traffic that they can handle in practice, it is required that each group that is created contains the same type of application traffic. Assuming the group contained in a subnet has the same application traffic, we provide manageable sample traffic to network operators for application traffic identification.
- ***Classifier Extractor:*** extracts application traffic classifiers such as common substring and destination address. Currently, we extract classifiers manually using several extraction guidelines. Because of the efficient grouping of the application traffic, extracting

distinguishable traffic characteristics as application traffic classifier is not complex.

- ***Classifier Manager:*** provides a user interface for network operators to manage the traffic classifiers. The main functionalities are addition, deletion, and scoping.
- ***Distributed Processing Platform:*** analyzes traffic in parallel according to the remote subnet. Our remote subnet-based traffic grouping can divide a large dataset into small sub-datasets that are independent of each other. We plan to apply a multi-processing platform, such as Hadoop [12] or Condor [13], to analyze traffic.

A. Identifying Application Traffic

We assume that all traffic directed to the same subnet is generated by the same application or service. This assumption is reasonable if the traffic is generated by a server-client application. The servers for a service are assigned to IP addresses that are also assigned to an organization; hence, the same application traffic is observed in a single subnet. On the other hand, most peer-to-peer (P2P) applications are used by desktop PC as both client and server; hence, traffic generated by a P2P application at a host is distributed to dozens of destinations belonging to different subnets. Even though P2P traffic is not grouped well, P2P application traffic is usually not mixed in the subnet where server-client application traffic is observed. In addition, because due to the characteristics of P2P traffic the same traffic pattern, such as flooding for searching a file, frequently appears network operators can easily detect P2P traffic patterns.

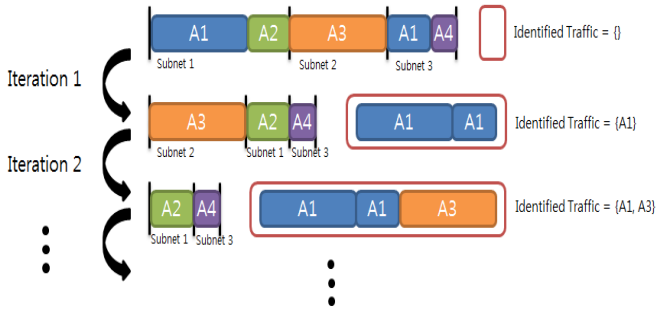

Figure 2. Subnet-based application traffic grouping and identification.

Algorithm 1. Pseudo code of subnet-based traffic identification algorithm.

```
01  While (completeness < th_comp)
02      Pick n subnets containing the largest unidentified traffic
03      For each destination subnet
04          Pick m flow
05          Extract common characteristics as traffic classifier
06      End for
07      Update identified traffic
08  End while
```

Figure 2 illustrates the proposed application traffic grouping and identification approach. The proposed approach is based on subnet grouping. We identify application traffic from the subnet where the largest traffic is directed. In Figure 2, *An* denotes the traffic generated by an application *n*. In *subnet 1*, application traffic from *A1* and *A2* exists. In the first iteration, if *A1* is identified, then *subnet 2* is the next target for analysis in the second iteration. Even though *A2* is not identified in the first iteration, it will have a chance to be analyzed when *subnet 1* is the subnet containing the most unidentified traffic.

Algorithm 1 depicts pseudo code for the proposed traffic identification methodology. To reduce the amount of time taken for analysis, we iteratively analyze *n* subnets until the identified traffic is greater than a specified threshold (the outer *while* loop in line 1). We need not analyze the traffic in its entirety; thus, we can identify significant application traffic and ignore irrelevant application traffic. After identifying application traffic in a subnet, we compare *m* flows whether the flows are generated from the same application or not. If a common byte pattern can be extracted from the *m* flows, then that common byte pattern can be used as the application traffic classifier. If only some of the *m* flows have a common byte pattern, the common pattern is used as the classifier and the remaining flows are ignored. The ignored flows are reanalyzed in subsequent iterations.

B. Extracting Traffic Classifiers

Because the proposed methodology uses subnet-based grouping, the traffic classifiers are also dependent on the subnet. This means that the classifiers have an effect on the subnet from which they are extracted, which is an advantageous property for applying classifier extraction and application. The most significant challenge when extracting traffic classifiers is the occurrence of miss-extracted classifiers. These miss-extracted classifiers have an effect on increasing both the false positive and false negative rates. The classifiers that are not convincing intimidate network operators as miss-classified traffic can cause service disruptions.

In this paper, we focus on the common substring in the payload (called the signature) as it is the most reliable traffic classifier. Traffic classifiers are used to filter out the identified application traffic. Thus, the classifiers cover entire or some part of the application traffic. To extract application traffic signature, we manually analyze payload streams after summarizing traffic according to flows in the same remote subnet. Previous studies have shown that the application traffic signature usually appears within the first several packets [1]. We use the first 10 packets in flows to analyze packet payloads.

When extracting traffic signatures, we usually capture traffic at the host at the end while running the target application. We then carefully analyze the traffic to avoid extracting ambiguous signatures. In the proposed approach, however, we do not require the traffic generation by the target application. Because our traffic classifier has an impact only on the subnet where the classifier is extracted, we can reduce the risk of ambiguous traffic classifiers. This means that common substrings such as company name and host name can be classifiers even though the common substring is contained in a packet that is generated by an incorrect application and is directed to the other subnet. Thus, this loose constraint makes it easy for network operator to extract application traffic

signature. We categorize application traffic into formatted application traffic and unformatted application traffic. Most Internet applications use their own application-level protocols, but reversing the protocols to extract signatures is difficult.

The guidelines for signature extraction for application traffic identification are as follows:

- For HTTP traffic, HTTP *User-Agent* is the primary classifier and the hostname can be used to label the application name.
- Service-level and device-level identification are recommended if the service and the type of access device can be identified.
- For SSL traffic, analyze the SSL header of the hello packet and use any noticeable string as much as possible.
- Any noticeable substrings such as application name, company name, and well-known byte pattern are good candidates for local classifier (available for one subnet).
- Summarize and update protocols of popular P2P applications.

Our methodology for application traffic identification is based on subnet grouping; hence, the common substring is the signature for the application if we find or infer the application name. For example, in a subnet, if most HTTP traffic (flow) contains 'Host: facebook.com\r\n', which can appear in other Web content as HTTP-host, we can infer that the subnet is the server farm for Facebook. Even if there are other application servers that are usually located in the CDN or Web hosting center, 'Host: facebook.com\r\n' is a unique substring that distinguishes *Facebook* traffic in this subnet. In other words, we can focus on finding application names or company names in a single subnet, which is not as complex as extracting signatures from the application traffic captured directly at the end host. Any common characteristics of application traffic, such as destination address and port number, can be applied as an application traffic classifier and the proposed approach can be easily extended to support these classifiers.

C. Applying Classifiers

Because we extract traffic classifiers from a single subnet, it is reasonable to suppose that applying the classifiers to the subnet determines the boundary of the classifiers. This boundary prevents the mixing of the entire application traffic, which means that the classifiers we extract are required for classifying traffic only for the mixed traffic of several applications. When we apply the classifiers, we can choose whether the classifiers are applied to a target subnet where the classifiers are extracted or to all subnets. If we manage traffic classifiers separately according to subnets, there are two main benefits to be derived. The first benefit is a reduction in the amount of time consumed in matching traffic classifiers. In fact, the traffic that appears in a single subnet was generated by only a few applications; hence, it is wasteful to match all application traffic classifiers to all of the traffic. The second benefit is to prevent limiting the impact of erroneous classifiers. Extracting the perfect classifier for application traffic is almost impossible because the classifiers can be the same as millions of other applications and we cannot generate application traffic containing all the protocols. In general, network operators always apply traffic classifiers with uncertainty and worry about the potential impact of erroneous classifiers. In the proposed approach, even if a traffic classifier is extracted with an error, the classifier is not matched to the traffic in the other subnets.

In this paper, we define two types of classifiers: local classifier and global classifiers. A local classifier is one that is applied to a target subnet such as that discussed in the previous section. On the other hand, a global classifier is one that is applied to all the subnets, which helps to define classifiers whose application traffic appears frequently in multiple subnets. The analyzer decides whether a classifier is local or global when he or she maps the label (application name) to the classifier. For example, if a Web browser or P2P application is identified and it appears frequently in more than twenty subnets, then it is better to apply a global classifier.

TABLE I. TRAFFIC TRACES CAPTURED AT POSTECH

	Time (YYYY-mm-dd-HH-MM)	Volume (MB)	Duration (s)
Dataset1	2012-03-01-19-00	2651	60
Dataset2	2012-03-01-20-00	833	60

TABLE II. TRAFFIC TRACES CAPTURED AT KOREA UNIVERSITY

	Time (YYYY-mm-dd-HH-MM)	Volume (MB)	Duration (s)
Dataset3	2011-10-19-14-00	83.5	60
Dataset4	2011-10-19-14-01	73.6	60
Dataset5	2011-10-19-14-02	76.1	60

IV. EVALUATION

In this section, we evaluate the proposed methodology to confirm its feasibility and completeness. We implemented the prototype that currently supports offline analysis on the POSTECH traffic monitoring system. To validate the completeness of the proposed approach, we captured POSTECH Internet traffic as sample traffic traces (TABLE I) containing both wired and Wi-Fi traffic of the residential area. In addition, to validate the accuracy of the proposed approach, we used traffic traces captured at Korea University (TABLE II). We used a 24-bit subnet mask (which is generally used to divide subnets) in all the experiments.

A. Extracted Traffic Classifiers for Application Traffic Identification

While identifying application traffic, we extracted and applied traffic classifiers to filter out identified application traffic. Even though we can use unique patterns as the application traffic classifier, we focused on extracting the common payload pattern—the most complex task for traffic analyzers. TABLE III shows the extracted payload classifiers for representative applications in dataset1 and dataset2.

We categorized application traffic into three categories: HTTP, P2P, and SSL (encrypted traffic). HTTP header is an example of formatted and human-readable payload; hence, extracting payload signatures was relatively easy. We analyzed *User-Agent, Host,* and *URL* to identify the application name

and the signature. Even though filling in the fields in the HTTP header is not a fundamental requirement for developers, the fields contain very helpful information that can be used to identify application traffic. In addition, we can extract OS information from HTTP *User-Agent* if OS name and version is explicitly written in the field. Identifying and extracting P2P traffic was more complex. The best approach to identifying P2P traffic is to analyze its connection pattern and the protocol. BitTorrent is one of the most famous P2P protocols in the world. The protocol has already been analyzed; hence, we can easily identify the BitTorrent protocol in subnet traffic. In addition, BitTorrent traffic was observed in 63,471 subnets among 110,853 monitored subnets during the monitoring period. This result means that applying the BitTorrent classifier as the global classifier is efficient in terms of reducing both redundant classifier and analyzer's effort. In contrast, we do not have any background on the Pandora TV protocol (a multimedia streaming service that uses the P2P protocol). We could identify the frequent appearance of Pandora TV and extract a common substring that is not contained in the other application traffic within the same subnet. You can be confident that the extracted Pandora TV classifier has not been confused with classifiers of the other applications after observing the same traffic pattern in several subnets. Otherwise, we can let the extracted Pandora TV classifier remain as the local classifier. Extracting classifiers from SSL payload is almost impossible, which is the common drawback of payload-based traffic classification methodologies. As a benefit of the subnet-based traffic identification, we can use payload patterns if the pattern is unique within a single subnet. Thus, we can extract 'google\.com' as the local classifier from Google SSL 'Hello packet' traffic if other traffic in the same subnet does not contain the substring.

TABLE III. EXAMPLES OF EXTRACTED PAYLOAD CLASSIFIERS.

Category	Application	Payload Classifier
HTTP	Daum Cloud	User-Agent: DaumCloud.*\r\n.*android
	iTunes	User-Agent: iTunes-iPhone.*\r\n
	NateOn	User-Agent: NateOn.*\r\n
P2P	BitTorrent	\x13BitTorrent protocol;^d1:rd2:id20:;^d1:ad2:id20:;
	PandoraTV	^\x0ePando protocol; PeerSvrHost=p2prtmp1.pandora.tv
SSL	Google SSL	google\.com
	Facebook SSL	facebook.com; fbcdn-sphotos-a.akamaihd.net

B. Completeness of Application Traffic Identification

Even though our research goal is not focused on accurate traffic classification, the completeness (how much traffic we can identify) is the major evaluation metric for the proposed approach.

Figure 3 shows the accumulated traffic volume of identified application traffic in the 10,000 subnets sorted by each traffic volume in descending order. We observed that the number of subnets where approximately 70% of all the traffic is identified ranges from 200 to 1,000. This result shows that the operators are only required to analyze 1.5% of the total number of subnets to identify 70% application traffic (because the number of the entire subnets is 2^{24} = 16,777,216 if we define each subnet using 24-bit netmask). After identifying 70% of the application traffic, the completeness of the identification seems to have become saturated. This phenomenon occurred as a result of the popup applications and the missing application traffic. We also analyzed the unidentified traffic generated by applications that appears infrequently. We also noticed that we missed some application traffic because we did not analyze the protocol of the application as we were not able to collect the target application traffic that covers its entire traffic patterns. From these observations, we conclude that it is better for us to focus on analyzing application protocols that are used by identified applications rather than try to find popup applications.

Figure 4 is the distribution of identified and unidentified traffic in terms of the number of flows and traffic volume. Most unidentified application traffic exits in subnets containing small number of flows and small traffic volume, which means the proposed methodology identifies the most used and important application traffic with high priority.

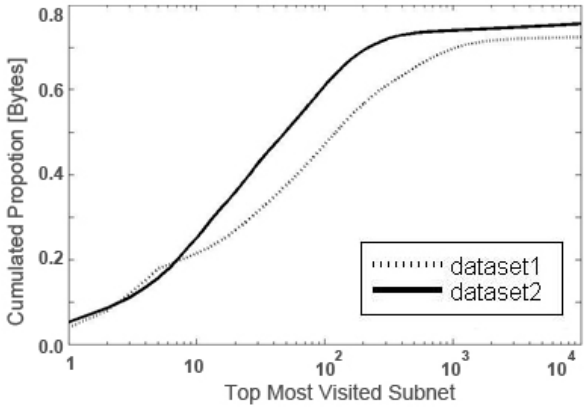

Figure 3. The accumulated traffic volume of identified application traffic in the most-visited remote subnets.

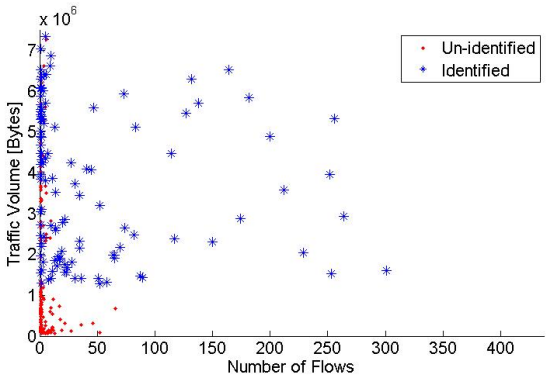

Figure 4. Distribution of identified and unidentified traffic in subnets in terms of the number of flows and traffic volume.

TABLE IV shows the traffic volume of identified application traffic in dataset1 and dataset2. The proportion for eleven of the most popular applications is 72.9%. The proposed subnet-based methodology identifies not only well-known

applications (e.g., BitTorrent) but also domestic applications (e.g., CoFile file share and Melon music player).

TABLE IV. THE VOLUME OF IDENTIFIED APPLICATION TRAFFIC

Application	Dataset1		Dataset2		Total	
	Vol. [MB]	*Prop. [%]*	*Vol. [MB]*	*Prop. [%]*	*Vol. [MB]*	*Prop. [%]*
BitTorrent	1332.81	50.3	91.21	10.9	1424.02	40.9
Web Browser for Windows	342.65	12.9	376.47	45.2	719.12	20.6
Pandora TV	113.30	4.3	0.00	0.0	113.30	3.3
Apple Core Media Player for iPhone	56.31	2.1	10.78	1.3	67.09	1.9
Kpeer Client	15.85	0.6	30.54	3.7	46.40	1.3
Web Browser for Android	26.02	1.0	18.91	2.3	44.93	1.3
Apple Core Media Player for iPad	10.09	0.4	34.33	4.1	44.42	1.3
Kpeer Engine	17.61	0.7	13.80	1.7	31.41	0.9
Webhard for Windows	0.00	0.0	19.45	2.3	19.45	0.6
CoFile File Share	12.28	0.5	6.35	0.8	18.62	0.5
Melon Music Streaming	8.19	0.3	4.07	0.5	12.26	0.4
Etc.	715.85	27.0	227.45	27.3	943.30	27.1
Total	2650.97	100.0	833.34	100.0	3484.31	100.0

C. *Feasibility of Subnet Grouping*

The research goal of this paper is to identify application traffic, rather than to classify application traffic accurately. As the prior task of traffic classification, we focus on identifying application traffic in the dark. In this section, we show how much application traffic can be identifiable using the proposed methodology.

To show how many applications were observed in the same subnet, Figure 5 gives a histogram of the number of subnets. To simplify the histogram, we took one hundred subnets containing a large volume of traffic. There were only a few subnets containing more than three application traffic classifiers. We assigned more than two application traffic classifiers to only 46 subnets in dataset1 and 42 subnets in dataset2 whose traffic volume are 7.62% and 6.89% of entire traffic respectively. This result shows that the proposed method does not require a complex procedure to extract application traffic classifier because most subnets contain the small number of application traffic. We also noticed that subnets containing many application traffic classifiers were those for CDN services or Web services. In the subnet for CDN services such as Akamai, we observed SNS services, mobile application market, Web browser plugins, etc. Observing multiple traffic classifiers in the subnets for Web services is the effect of subdividing the traffic classifiers according to Web browsers and operating systems.

Figure 6 depicts scatter plots that illustrate the number of classifiers and the number of flows in the top-100 subnets that contain the large volume of traffic. The subnets that contain the large number of flows were the subnets assigned to popular Web services. The subnets that contain the large number of classifiers were the popular Web services including mobile Web services. Interestingly, we did not find a subnet containing multiple classifiers of non-Web traffic. In other words, few subnets exist in which multiple application servers are located, which exactly matches our assumption that the subnet-based traffic grouping makes a group of the same application traffic.

Figure 7 shows the relationship between the number of classifiers and traffic volume among the top-100 subnets. The subnets containing the large number of classifiers are also the subnet for Web services, and the subnets containing the large volume of traffic is usually file transmission of BitTorrent that generates much larger traffic than Web-based applications.

TABLE V. TOTAL APPLICATIONS REPORTED BY TMA AND IDENTIFIED APPLICATIONS IN DATASET3, 4, AND 5.

TMA (GT)	Identified App.
Internet Explorer	Web Services[1]
Google Chrome	
AL Song	ALTools
AL Yac	AL Yac
Naver Tools	NaverAXGuide
BitTorrent	BitTorrent
Drop Box	Drop Box
Melon Music Player	Melon music streaming
NateOn	NateOn
Naver Mini Challendar	Naver Calendar
Outlook	Naver Mail
nProtect	nProtect
AhnLab MyV3	AhnLab
My People	Daum Mobile
	Daum Webmail for Windows
-	Android Market
Pandora Media	-
PotPlayer	-
Skype	-
Yahoo Messenger	-

1: We differentiate Web services in TABLE VI

D. *Accuracy of Application Traffic Identification*

Because the aim of this research is to identify application traffic at the beginning of the classifier generation procedure, it is difficult to validate the result of the proposed methodology without knowledge of the application usage in the network. To validate the application traffic that we identify, we use Traffic Monitoring Agent (TMA)—the agent program generating traffic logs labeled with the process name at the end hosts [1]. From TMA records, we obtained information as to which flow is exactly generated by which application. We applied the proposed methodology to the traffic trace from Korea University and obtained the ground-truth data from TMAs that are installed on the public PCs. In the evaluation of the accuracy of the proposed method, comparing the identified applications with the ground-truth data is the most significant experiment.

TABLE V compares the identified applications and the ground-truth reported by TMA. We identified 15 applications and TMA reported 18 applications in dataset 3, 4, and 5. The proposed approach can identify application traffic accurately even though we do not use any extra information. However, we missed several applications in the traffic. There are three reasons why we missed the applications. First, we could not

find application traffic classifier from application traffic. For example, identifying Skype traffic without knowledge of the protocol is difficult. Second, we missed the application's traffic because the application does not fill the *User-Agent* field correctly. Even though filling HTTP header fields is recommended for application developers, we noticed that the same *User-Agent* value with the Mozilla Web browser is used for some applications. For example, Yahoo Messenger uses the same *User-Agent* as the Mozilla Web browser; hence, we identified Yahoo Messenger as a Web browser. The last reason is the small segmented traffic transmission of P2P traffic. We noticed that Pandora Media receives small multimedia data from dozens of hosts that are scattered across multiple subnets. Because our application traffic grouping is based on the remote subnet, if an application breaks data into very small fragments and receive them from multiple subnets then the subnets are not reported as high volume and important subnets. In this case, by performing more iterations (outer loop of Algorithm 1), we can finally identify the small fragmented traffic. However, we identified most application traffic, that is, a large proportion of the analysis traffic.

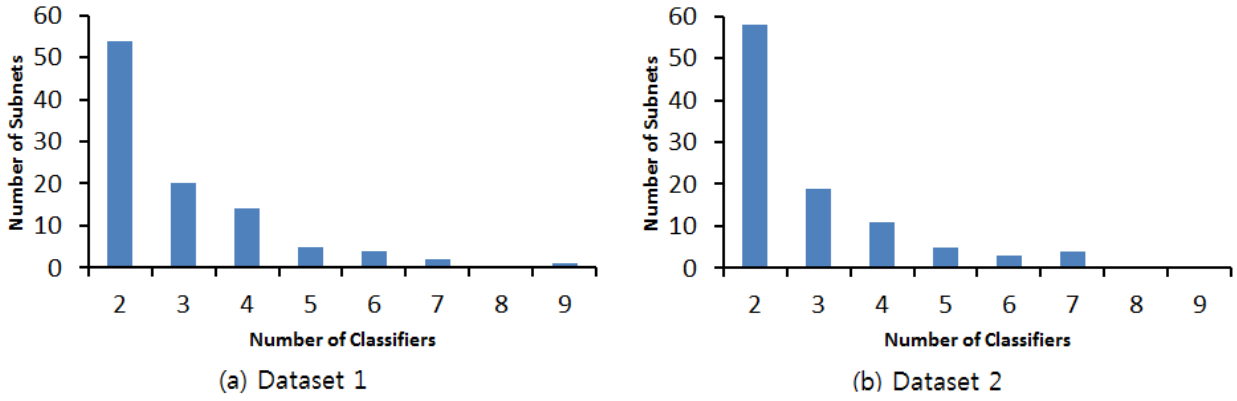

Figure 5. Histogram of the number of classifiers that are assigned to the top-100 subnets.

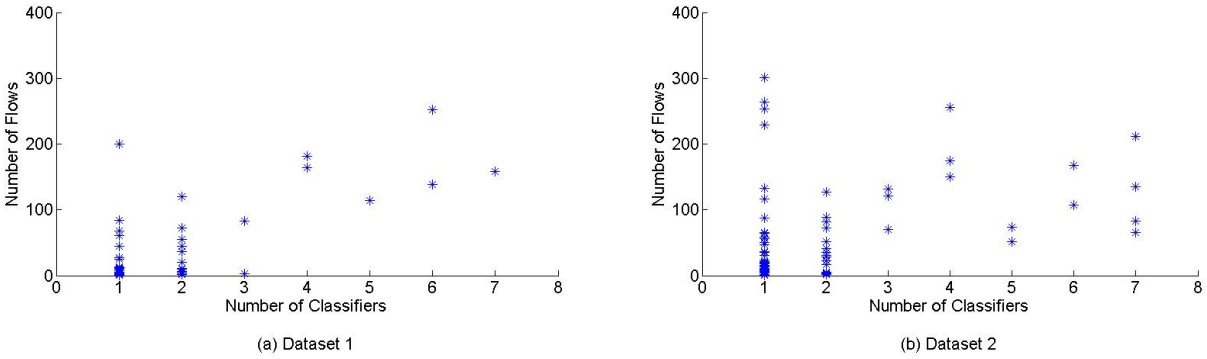

Figure 6. Scatter plot of the number of classifiers and the number of flows that are observed in the top-100 subnets.

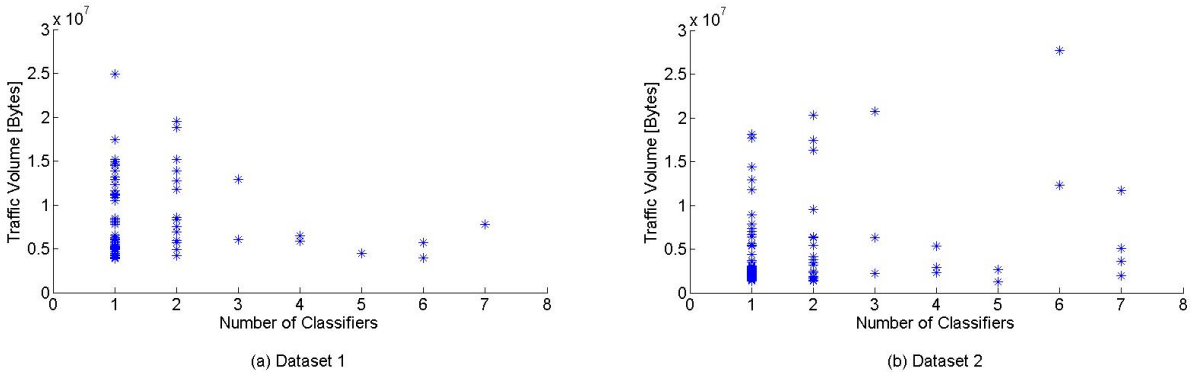

Figure 7. Scatter plot of the number of classifiers and traffic volume that are observed in the top-100 subnets.

TABLE VI. WEB APPLICATION DIFFERENTIATING FOR WEB SERVICE IDENTIFICATION.

Web Service	Web Browsing	Mobile	Multimedia
Web Browser for Android	O	O	
Web Browser for iPad	O	O	
Web Browser for iPhone	O	O	
Web Browser for Linux	O		
Web Browser for Macintosh	O		
Web Browser for Windows	O		
Web Browser for Windows Media PC	O		
Youtube for iPad		O	O
Youtube for iPhone		O	O
AppleCoreMedia		O	O
Facebook for iPhone		O	
Google Check Internet			
Google for iPhone		O	
Google Mail			
Google SSL			
Google voice search		O	
Twitter (image)			
Naver search for Android		O	
NaverMusic			O

Because TMA is based on the socket event on operating systems, TMA records the process name that handles the socket and flow information. Thus, TMA cannot distinguish Web services offered by Web browsers. In TABLE V, TMA reported two Internet Web browsers: Internet Explorer and Google Chrome browsers. However, we identified Web-based services according to service name and a more detailed application breakdown scheme. We differentiated Web services into 24 services, including Web browsers for different operating systems (Windows, iPhone, Android, etc.), email, search, and multimedia streaming (TABLE VI). In other words, our traffic identification can differentiate Web-based application traffic and support fine-grained traffic breakdown even though we analyzed traffic traces without knowledge of the popularity of applications.

V. CONCLUDING REMARKS

In this paper, we proposed a methodology for application traffic identification to utilize traffic classification research from the beginning. The proposed methodology is based on remote subnet grouping—the collection of the same or similar application traffic. As the initial task for traffic classifier generation, the proposed methodology contributes by helping the analyzer to identify popular applications in the network.

We also performed experiments to validate the proposed methodology. We identified over 70% of the application traffic even though we are not focusing on exacting accurate traffic classifiers and classification accuracy. By showing the most subnets containing only one or two types of application traffic, we also showed that traffic in a subnet is the collection of the same or similar application traffic. To compare the identified application traffic with the actual application traffic in the network, we used process labels generated by TMA (an agent installed at the end hosts). We also had fine-grained Web-based application traffic identification according to services and operating systems.

In the future, we plan to develop an integrated system based on the current traffic monitoring system and the proposed approach, which will facilitate the generation and management of classifiers as well as the classification of campus network traffic. We also plan to utilize a distributed processing platform to reduce the processing time by analyzing subnet traffic in parallel.

ACKNOWLEDGEMENT

This research was supported by World Class University program funded by the Ministry of Education, Science and Technology through the National Research Foundation of Korea (R31-10100) and the MKE(The Ministry of Knowledge Economy), Korea and Microsoft Research, under IT/SW Creative research program supervised by the NIPA(National IT Industry Promotion Agency) (NIPA-2011-C1810-1102-0053).

REFERENCES

[1] B.C. Park, Y.J. Won, M.S. Kim, and J.W. Hong. "Towards Automated Application Signature Generation for Traffic Identification", Proc. of the IEEE/IFIP Network Operations and Management Symposium (NOMS) 2008, Salvador, Brazil, Apr. 7-11, 2008, pp. 160-167.

[2] M. Ye, K. Xu, J. Wu, and H. Po, "AutoSig-Automatically Generating Signatures for Applications", IEEE 9th International Conference on Computer and Information Technology, Xiamen, China, Oct. 11-14, 2009, pp. 104-109.

[3] A.W. Moore, and K. Papagiannaki, "Toward the Accurate Identification of Network Applications", Passive and Active Measurement Conference, Boston, MA, USA, Mar. 31-Apr. 1, 2005, pp. 41-54.

[4] S. Sen, O. Spatscheck, and D. Wang, "Accurate, Scalable In-Network Identification of P2P Traffic using Application Signatures", International World Wide Web Conference, NY, USA, May 19-21, 2004, pp. 512-521.

[5] P. Haffner, S. Sen, and O. Spatscheck, "ACAS: Automated Construction of Application Signatures", ACM SIGCOMM Workshop on Mining Network Data, Philadelphia, PA, USA, Aug. 26, 2005, 197-202.

[6] F. Risso, M. Baldi, O. Morandi, A. Baldini, and P. Monclus, "Lightweight, Payload-Based Traffic Classification: An Experimental Evaluation", International Conference on Communications, Beijing, China, May 19-23, 2008, pp. 5869-5875.

[7] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: Multilevel Traffic Classification in the Dark", ACM SIGCOMM 2005, Philadelphia, PA, USA, Aug. 21-26, 2005, pp. 229-240.

[8] T. Karagiannis, A. Broido, M. Faloutsos, and Kc claffy, "Transport Layer Identification of P2P Traffic", Internet Measurement Conference, Taormina, Sicily, Italy, Oct. 25-27, 2004, pp. 121-134.

[9] A.W. Moore and D. Zeuv, "Internet Traffic Classification Using Bayesian Analysis Techniques", International Conference on Measurements and Modeling of Computer Systems, Banff, Alberta, Canada, Jun. 6-10, 2005, pp. 50-60.

[10] J. Erman, M. Arlitt, and A. Mahanti, "Traffic Classification Using Clustering Algorithms", SIGCOMM Workshop on Mining Network Data, Pisa, Italy, Sep. 11-15, 2006, pp. 281-286.

[11] B. Park, Y.J. Won, and J.W. Hong, "Toward Fine-grained Traffic Classification", IEEE Communications Magazine, vol. 49, no. 7, July, 2011. pp. 104-111.

[12] Hadoop, https://hadoop.apache.org/.

[13] D. Thain, T. Tannenhaum, and M. Livny, "Distributed computing in practice: the Condor experience: Research Articles", Journal of Concurrency and Computation: Practice and Experience, vol. 17, issue 2-4, Feb. 2005, pp. 323-356.