Towards Smart Phone Traffic Classification

Min Hur and Myung-Sup Kim
Dept. of Computer and Information Science
Korea University
Korea
{grrrhm, tmskim}@korea.ac.kr

***Abstract*— The appearance of smart phones and their continuing rapid uptake has large affects on our society in as much as they represent a paradigm shift in the traditional industrial structure. The Telecom market is changing day by day, networks with the complicated and varied traffic have almost reached capacity because of the rapid increase of user and the service releases on smart phones. Therefore, the necessity for smart phone traffic monitoring and analysis has increased. Traffic analysis is an essential element for efficient and reliable networks. In this paper, we propose a new smart phone traffic classification by application method. The proposed method is composed of several consecutive steps: grouping the HTTP User-Agent field, extracting common strings by the LCS algorithm and finally classifying the traffic. In addition, to classify unknown traffic from previous methods, we propose a process that extracts header signatures in grouped information to improve the classification completeness. We achieved about a 90% accuracy rate for the analysis by our proposed method in the target campus network.**

Keywords- Traffic classification; Traffic Identification; Smart phoene; User-agent;

I. Introduction

The appearance of smart phones and their continuing rapid uptake has large affects on our society in as much as they represent a paradigm shift in the traditional industrial structure. The Telecom market is changing day by day. Application program development of a new type is taking place in an open platform environment, there is a diversification of user requirements and the user is demanding a high quality of service and quality assurance. Providers are demanding low management costs for networks and want to explore new profitable areas by providing several high quality services to users. According to ISP, traffic is expected to surge by 20 to 40 times over the next three years.[1] Smart device traffic will increase exponentially every year until 2015. In particular, video and web traffic from smart phones are increasing at a higher rate than any other traffic. It is expected this traffic will increase more than 60 fold by 2015. Domestic smart phone traffic trends were found to be similar to CISCO's expectations. In SKT's case, 2010 smart phone traffic experienced a 12 fold increase compared to 2009. KT (Korea ISP) has seen a 300 fold increase during the same period.

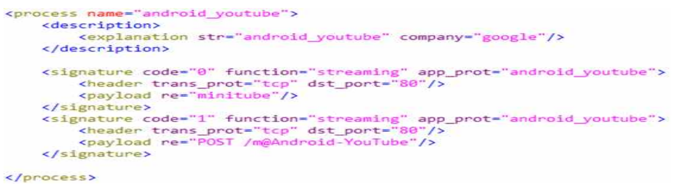
The rapid increase of smart phone traffic aggravates network problems; the quality of communication service is degraded because of the proliferation of users in a particular network section causing a bottleneck by time. Traffic analysis is required research for reliable network operations and service quality assurance. However, research into smart phone traffic analysis is in its early staged. Also, methods to collect traffic, analyses it and research the methodology for classification are lacking. In this paper, we propose an objective method for collecting smart phone traffic and classifying this traffic by application. We prove the feasibility of our proposed method in the target campus network.

This paper is organized as follows. Section 2 describes existing research related to this issue. Section 3 presents a signature extraction method for traffic classification by application. Section 4 describes the experimental environment and an analysis of the results. Finally, in Section 5, conclusions and future research directions are described.

II. Related Work

In this section, we look at the problems of previous methods for smart traffic classification.

The paper [2] described a payload signature method that was hand-made using the XML format, as in figure 1 and classified the traffic by smart phone application. It showed a classification rate of 15.37% for flow, 7.70% for byte and 10.52% for packet, we mentioned the difficulty to classify traffic due to the low classification rate when extracting signatures by hand because of the various kinds of applications for smart phone users.



```
<process name="android_youtube">
    <description>
        <explanation str="android_youtube" company="google"/>
    </description>

    <signature code="0" function="streaming" app_prot="android_youtube">
        <header trans_prot="tcp" dst_port="80"/>
        <payload re="minitube"/>
    </signature>
    <signature code="1" function="streaming" app_prot="android_youtube">
        <header trans_prot="tcp" dst_port="80"/>
        <payload re="POST /m@Android-YouTube"/>
    </signature>

</process>
```

Figure 1. Payload signature in hand-made

The method for extracting signature based on an Agent is very useful for creating automated-signatures but the mTMA [3] (mobile Traffic Measurement Agent), seen in figure 2, is not good at classifying smart phone traffic. This must develop a separate agent for the various kinds of operating systems. In other words, the signature depends on the operating system so can provide incorrect information for objective traffic classification, as such, we propose a method to classify traffic

This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (20100020728).

by service via HTTP traffic which doesn't depend on operating system.

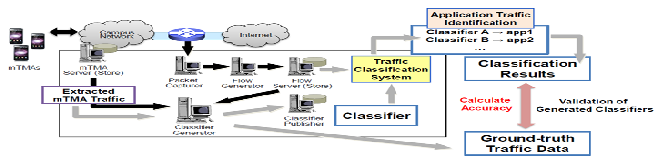


Figure 2. Mobile signature extraction system

III. Method of Extracting Smart Phone Signatures

In this section, we outline a method to improve the classification rate of smart phone by application, especially for traffic which can't be classified through grouping User-Agents, for the classification, extracting the common strings with LCS algorithm and organizing header signature.

Figure 3 shows the overall traffic classification process. After extracting only smart phone traffic using OS fingerprinting [4] from the entire trace, we extract the HTTP and Non-HTTP traffic by protocol examination. We examine the user-agent field to analyze the HTTP traffic from smart phones. The user-agent string is grouped based on the server IP, port and string size. We extract common string using LCS algorithm. After the system extracts common strings, it converts them to regular expression in order to enhance their semantic meaning. Extracting the common strings allows us to get the name of the application or engine that is using service. In this way, we are able to classify application traffic. Also, we produce a header signature using an extracted common string for analysis of Non-HTTP and Non User-Agent traffic.

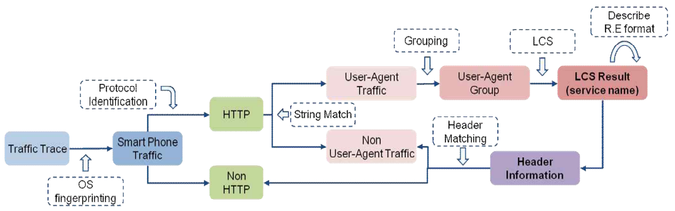


Figure 3. The classification process of smartphone traffic

A. User-agent Grouping for Extraction of Common Strings

This chapter describes the User-Agent for HTTP traffic grouping method for smooth common string extraction.

We group user-agent strings by extracting HTTP traffic of smart phones from the entire trace. This consists of preprocessing for extracting common strings using the LCS algorithm and creating a header signature for further analysis. The header signature is used to generate flow information (server IP, server port and user-agent size) when grouping. Figure 4 shows the user-agent set from one server IP that is not grouped.

```
========================= User-Agent 1 =========================
Mozilla/5.0 (Linux; U; Android 2.2; ko-kr; LG-LU3700 Build/FRF91)
AppleWebKit/533.1 (KHTML, like Gecko) Version/4.0 Mobile Safari/533.1

========================= User-Agent 2 =========================
Mozilla/5.0 (Linux; U; Android 2.2; ko-kr; LG-LU3700 Build/FRF91)
AppleWebKit/533.1 (KHTML, like Gecko) Version/4.0 Mobile Safari/533.1
```

Figure 4. The user-agent set that is not grouped

We can know various browser and connection devices to the server through the user-agent information of figure 4. When we extract common strings using the LCS algorithm from the set of a user-agent, we would extract 'Mozilla/' from the meaningless character string. We need preprocessing to get the correct extractable form and to prevent extraction of meaningless character strings.

The grouping algorithm checks the HTTP pattern that contains the flow information and its payload. Also, it checks the user-agent string. If a user-agent string exists, it checks the creation of the group, and if not, it generates a new group and inserts a string into the group. Figure 5 shows the schematically grouped user-agent. Initially, the server IP is grouped and then the server port and string size of the user-agent is connected to that IP.

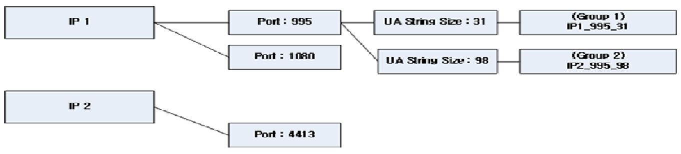
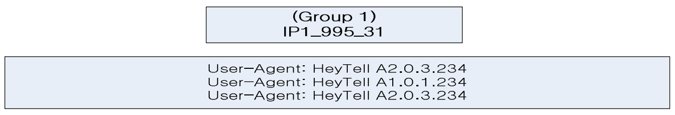


Figure 5. The structure of user-agent

Figure 6 shows group 1 of a user-agent. By looking at the name of the group, server IP 1 is generated and 995 port is used. The user-agent size is 31 bytes. The grouped user-agent in figure 6 is the iPhone's phonetic alphabet 'Heytell'. It's a service for voice messages between users.



(Group 1)
IP1_995_31

User-Agent: HeyTell A2.0.3.234
User-Agent: HeyTell A1.0.1.234
User-Agent: HeyTell A2.0.3.234

Figure 6. The grouped user-agent(IP, port, user-agent size)

When looking at the process in figure 6, each group will be chosen based on the user-agent string. One of the groups has several user-agents containing the name of service and the version information. It will be used for the extraction of common string using the LCS algorithm.

B. The Extraction of Common String Using the LCS Algorithm

There are several algorithms for finding the LCS (Longest Common Subsequence). The Brute Force algorithm needs Θ (n2m) of time and requires a lot of storage space. This algorithm is not suitable for extraction because of the large input data set and traffic from applications is complex. In this paper, we apply the Dynamic Programming for the LCS algorithm. This algorithm has low time complexity and uses less storage space.

We removed the common 'User-Agent' substring, because it is extracted to the HTTP user-agent. We are removed 'Darwin' and 'Mozilla' occurring in normal web traffic. Strings of less than 2 bytes string are removed because of their low evaluation factors.

Figure 7 shows the LCS result for the iPhone's 'Face Worldcup'. LCS result for the 'Face Worldcup' Service contains version information only one version, 2.0, appeared in the extraction traffic. On the other hand, in the 'HeyTell' extraction in figure 8 the version number is deleted because several versions of the application appeared in the target network.

FaceWorldcup/2.0.0 CFNetwork/485.12.7/Darwin/10.4.0

Figure 7. The LCS result of 'FaceWorldcup'

HeyTell A.0.234

Figure 8. The LCS result of 'HeyTell'

C. *Production of Header Signature for Further Analysis*

Traffic classification by application through user-agent applies the traffic analysis extracted from the user-agent with being created in the point that its applicable name of HTTP traffic or the using engine specified in it. However, it is important not to only analyze the traffic in which the applicable pattern appears when the relevant method applies but also when it occurs in any traffic except for HTTP protocol. In this paper, in order to overcome this weakness and improve the analysis rate, we construct a header signature through applicable common strings in order to classify and analyze the flows without the same user-agent as its service for non-HTTP traffic. Figure 9 shows our method to analyze non-HTTP traffic and non-user-agent traffic by matching the header information with the header signature formed by the header information extracted from the traffic grouped in its service through the user-agent.

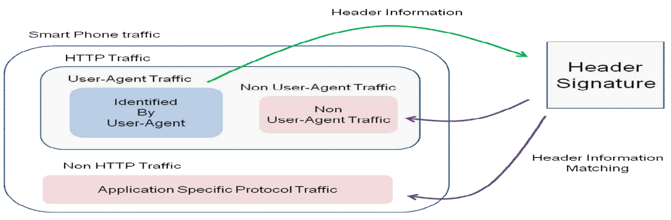


Figure 9. The traffic classification using header signature

Figure 10 shows a method to analyze unique protocol traffic by constructing a header signature with the IP of the server extracted user-agent of the Melon service and apply it via IP matching.

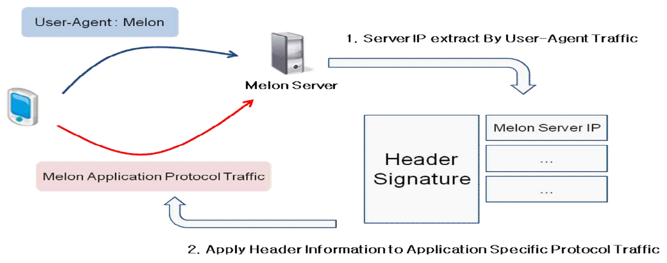

User-Agent : Melon
Melon Server
1. Server IP extract By User-Agent Traffic
Header Signature
Melon Server IP
...
...
Melon Application Protocol Traffic
2. Apply Header Information to Application Specific Protocol Traffic

Figure 10. Apply header signature to Melon service

Figure 11 shows the method to construct header signature. The header signature is created by information from each user-agent grouping and the result of the LCS. Each user-agent group has IP and Port information so that comes in handy when the header signature is formed. The header signature can classify the application traffic of its header signature because it is consists of a server IP formed as a type of linked-list in an IP hash table, of the engine used or the application name.

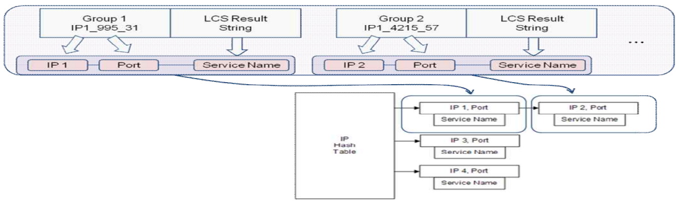


Figure 11. The conversion to header signature from user-agent group

The approach of linked-list in hash table means it can get close to each element according to the server IP or the creation of hash key. When constructing a header signature, the common strings receive each group as an input and inspect the existence of its group in header signature to prevent it from being duplicated elsewhere; they change the record type of the header signature and insert their service name. The result from the LCS is registered as the service name but sometimes user input is required to get an accurate application name. For example, Safari strings are needed when the whole user-agent gets the result from LCS, like in figure 12. This part is determined by user decision. In order to decrease the problem by making a decision like this afterward.

Mozilla/5.0 (iPad; U; CPU OS 4_3_2 like Mac OS X; ko-kr) AppleWebKit/533.17.9 (KHTML, like Gecko) Version/5.0.2 Mobile/8H7 Safari/6533.18.5

Figure 12. The extraction of whole user-agent

IV. EXPREIMENT RESULT

This section explains about the results of the analysis with the experimental conditions constructed for the test.

A. *Experimental Conditions and Traffic Trace*

This chapter describes the overall system environment for extracting and analyzing smart phone traffic occurring on an enterprise network.

The environment for the experiment is organized as shown in figure 13, packets are collected from the top router linking the school network and the outside internet. An OS fingerprintor checks whether the traffic comes from smart phones or not, if so, it records the time and IP in the database. The smartphone traffic extractor only picks up traffic from smartphones. User-agent grouping, LCS and the service identification system extract the user-agent field with the methods suggested in 3.A and 3.B and gathers them in a group. They also extract the name of the application or engine and classify the traffic by application. The header signature generation and analysis system form the header signatures with the method outlined in 3.C and classify the traffic which cannot be analyzed on the basis of the User-Agent header signature.

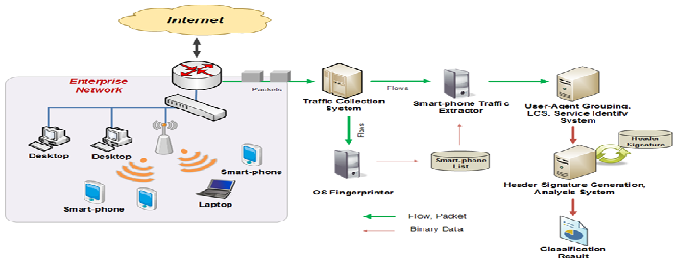


Figure 13. The experimental environment

TABLE 1. TRAFFIC TRACE

Duration	Flows	Packets	Bytes
2011-11-07 ~ 13	366M	57,927T	93,246TB
2011-12-05 ~ 11	407M	64,041T	103,121TB

Table 1 show the traffic trace information collected from the target network. It displays all the information from the traffic trace occurring in about three thousand hosts, including smart phones, and consists of service traffic from various kinds of application.

B. *Result of Experiment and Analysis*

In this chapter, we talk about the additionally analyzed results in the case of applying the header signature with classified traffic outputs from the LCS algorithm and user-agent grouping that we suggested.

Table 2 shows the names of the engines and applications extracted by user-agent groups using the LCS algorithms Entries in the table below number 5 are the names of the services originated from HTTP traffic where the names of Korean applications in the user-agent field are encoded and transmitted as UTF-8.

TABLE 2. THE SIGNATUER OF USER-AGENT

No	***User-Agent***
1	MozilgeToeic
2	SpeedTest
3	Safari/
4	DaumCloud
5	%ED%95%A0%EC%9D%B8%EC%9D%
…	…
2325	iRollerCoastr

Figure 14 is the graph analyzing user-agent signatures and header signatures in a group unit. User-agent signatures success rates on average were 75.7% for flow, 81.3% for packet and 81.5% for byte data. The header signature provided additional analysis for the traffic which the user-agent signature could not analyze improving the overall success rate by 5.7% for flow, 5.8% for packet and 7.2% for byte data on average . The reason the analysis rate for byte is higher compared to flow is that the formed header signature has analyzed the flows that have high generation of bytes i.e. heavy flows. HTTP traffic basically generates more bytes than other traffic because it is a protocol based on TEXT.

We performed verification using a key made by hand and comparing our results with the results of the classifications only in matched traffic. We made a key for four services: Afreeca, Facebook, Kakaotalk and Melon using a method where a smartphone connects to the designated sharer, uses the service and collects the traffic from the sharer.

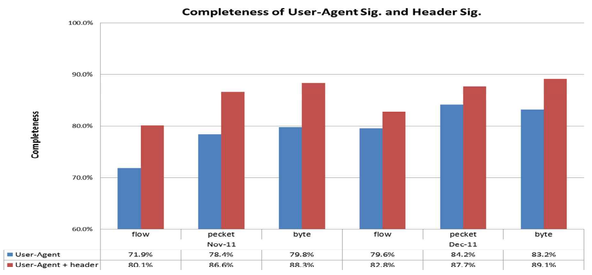


Figure 14. The Completeness of Flow, Packet and Byte

Table 3 shows the accuracy of the classification in each service via real verification, showing 90% accuracy.

TABLE 3. ACCURACY OF THE PROPOSED METHOD

Service Name	***Recall %***	***TP Flow***	***FN Flow***
Afreeca	96.0	461	19
Facebook	91.7	134	12
Kakaotalk	82.6	382	80
Melon	89.7	711	81
Total	***90.0***	***1,688***	***192***

V. CONCLUSION AND FUTURE WORK

In this paper we proposed a methodology that classifies smart phone traffic by application to understand the behavior of smartphone applications, an area that has become an issue lately. The proposed method improves the classification performance using several consecutive steps: grouping traffic flows by using the user-agent field in the HTTP header, extracting common strings via LCS and IP header information, classifying the traffic by application.

We are going to continue our research to find a method that extracts a signature from other fields, except the user-agent field, of smart phone traffic. In addition, we are planning to study the application occurring for each service from smart phones and the dependency of the engine in order to define that relationship.

REFERENCES

[1] Cisco, "Cisco Visual networking Index: Global Mobile Data Traffic Forecast Update," White Paper, Feb. 1, 2011

[2] Sang-Woo Lee, Jun-Sang Park, Hyun-Shin Lee, and Myung-Sup Kim, "A Study on Smart-phone Traffic Analysis," Proc. of the Asia-Pacific Network Operations and Management Symposium (APNOMS) 2011, Taipei, Taiwan, Sep. 21-23, 2011.

[3] Yeong-Rak Choi, Jae-Yoon Chung, Byung-Chul Park and Won-Ki Hong,"A Study on System Architecture for Application-Level Mobile Traffic Monitoring and Analysis", KNOM Review, vol. 14, no. 2, Dec. 2011, pp. 10-21.

[4] Hyun-shin Lee, Myung-sup Kim "Research on OS fingerprinting method for real-time analysis system", in proceedings of Korea Information and Communication Society(KICS)

[5] Jun-Sang Park, Sung-Ho Yoon, and Myung-Sup Kim, "Software Architecture for a Lightweight Payload Signature-based Traffic Classification System," Proc. of the Traffic Monitoring and Analysis (TMA) Workshop 2011, LNCS6613, Vienna, Austria, Apr. 27, 2011, pp. 136-149.