Hardware Design for Statistical Network Traffic Classifiers

Chun-Nan Lu
Computer Science
National Chiao Tung University
Hsinchu, Taiwan
e-mail: cnlu@cs.nctu.edu.tw

Yuan-Cheng Lai
Information Management
National Taiwan University of Science and Technology
Taipei, Taiwan
e-mail: laiyc@cs.ntust.edu.tw

Chun-Ying Huang
Computer Science
National Chiao Tung University
Hsinchu, Taiwan
e-mail: chuang@cs.nctu.edu.tw

Ying-Dar Lin
Computer Science
National Chiao Tung University,
Hsinchu, Taiwan
e-mail: ydlin@cs.nctu.edu.tw

***Abstract*—Signature matching is commonly used in network traffic classification and can provide accurate and efficient results. However, it requires constant updates of signatures and can't be applied to encrypted traffic. Statistical behavior-based approaches can avoid the drawback of payload encryption. However, the computational complexity of related statistical features may prevent them being deployed in systems that are expected to respond in limited time. In this work, we combine the advantages of statistics-based classification approaches and hardware design techniques to develop a balanced classifier that can provide timely responses to. Two statistics-based solutions, a message size distribution classifier (MSDC) and a message size sequence classifier (MSSC) which depend on classification accuracy and real timeliness are proposed. The former aims to identify network flows in an accurate but not-so-fast manner, while the latter aims to provide a lightweight and real-time solution. Simulations showed that MSSC contributed 77.4% and MSDC contributed 22.6% of decision rounds. Furthermore, our design can achieve an accuracy of more than 94% while achieving a throughput of 80 Gbps.**

Keywords—traffic classification, packet size, message size distribution, sequence, hardware classifier

I. INTRODUCTION

Classifying a network flow by its source applications is essential for application-aware network management. However, it is not an easy task to correctly classify network flows into their corresponding applications because of obfuscation techniques such as port number randomization, payload encryption, and network tunneling, which are used to avoid detection. As a result, characterization of Internet traffic has become one of the major challenging issues in communication networks over the past few years [1].

In this work, we base our hardware design on a hybrid traffic classification solution composed of two statistical classifiers, message size distribution classifier (MSDC) [2] and message size sequence classifier (MSSC) [3]. MSDC provides good accuracy, but it has lower throughput because of its statistical computation overheads. By contrast, MSSC attempts to track the application states of flows to make classifications. As long as the states can be clearly identified, MSSC can rapidly make a decision. As a result, MSSC has better throughput. However, MSSC may not be accurate enough because it uses short common subsequences. Inaccurate classifications may occur when incomplete packets of a flow are captured or states of an application behavior are similar to the states of another application's behaviors. Therefore, a hybrid solution is developed to combine MSDC and MSSC classifiers, to provide a balanced solution in terms of classification accuracy and response latency.

The organization of this paper is as follows. In Section II, relevant past researches on network flow classification is reviewed. Section III briefly describes the two statistical classifiers, MSDC and MSSC. The proposed methodology and hardware architecture of the hybrid solution is presented in Section IV. Section V gives the simulation results. Finally, conclusions are presented in Section VI.

II. RELATED WORK

Various statistical-based network flow classification approaches have been proposed in recent years. The advantage of these methods is the ability to classify an application without the need to inspect the packet payloads. All these approaches could be classified into flow-level and session-level classes. The former classifies each flow independently while the latter attempts to group network flows as sessions by using heuristic rules and then classifies network flows in a session-based manner.

A. Flow-level and Session-level Classification

Many statistical techniques observe outer characteristics, like traffic volume, flow duration, flow burstiness, packet payload size, or the jitter of network flows, to classify network flows. Those techniques generally consist of training and classification phases. A representative model is first built using extracted statistical attributes of flows by learning the inherent structural patterns of datasets, and the model is then used to classify network flows [4, 5].

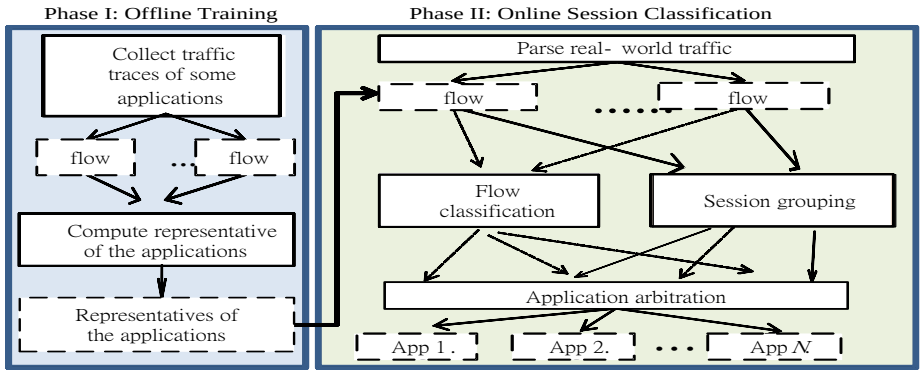

Fig. 1. Components and operation flows of MSDC

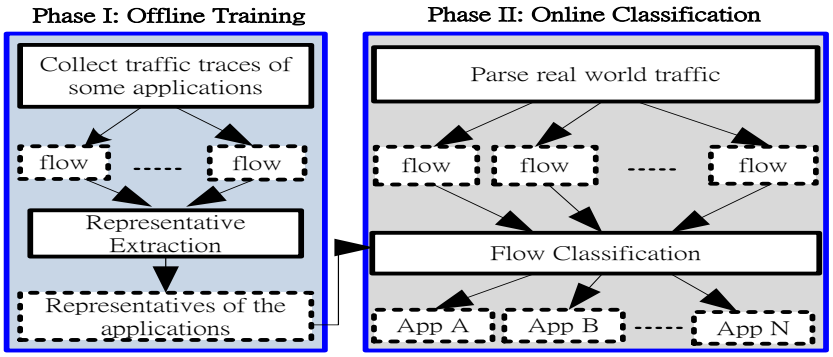

Figure 2. Components and operation flows of MSSC

A few works analyze traffic at a level other than flow level. Kannan et al. [6] used a flow-level trace to derive abstract descriptions of the session-structure for different applications present in the trace. Based on flows' statistical information, Kannan's approach can discover and characterize flow/session causality relationship and further infer applications' internal session structures. Karagiannis et al. [7] developed a traffic classification approach based on the analysis of host behavior. It associates Internet host behavior patterns with one or more applications, and refines the association by heuristics and behavior stratification.

B. Hardware design

Hardware is often employed to handle the computation-intensive part to accelerate the throughput. SnortOffloader [8] and Shunting [9] offloaded the subset of traffic that is large in volume but of little interest to intrusion detection systems.

III. STATISTICAL METHODOLOGIES

Two statistical classifiers, MSDC and MSSC, are discussed in this section. The former aims to provide an accurate but not-so-fast solution while the latter aims to provide a lightweight and real-time solution. Both MSDC and MSSC have to collect application traffic to develop application representatives and then use the representatives to classify network flows.

A. Message size distribution classifier (MSDC)

MSDC runs in two phases: an offline application representatives training phase, and an online session classification phase. Figure 1 shows an overview of MSDC. The left block shows the steps of the training phase and the right block shows the online classifier, which includes three modules, flow classification, session grouping, and application arbitration modules.

With packet size distribution (PSD), each flow is transformed into a set of points in a two-dimensional space. The goal of the offline training phase is to find out application representatives, which should be unique to or different from other applications. Hence, the training phase collects a set of traffic traces and extracts the representatives from the five-tuple information (source IP, source port, destination IP, destination port, transport layer protocol) and the PSD of all captured flows.

The online session classification phase first extracts the five-tuple information (source IP, source port, destination IP, destination port, protocol) and the PSD from all real-world flows. Next, the flow classification module compares the incoming flows with application representatives and classifies them into the application with minimum similarity distance. Meanwhile, the session grouping module attempts to group flows as a session based on port locality. After the above phases, each flow should be classified into some application and flows having adjacent ports should be grouped into the same session. If two or more flows of a session are classified as different applications, the application arbitration module, majority vote, is invoked to solve the conflict and make the correction. If flows of two or more different applications are grouped together, all flows of the session will be treated as the application with the largest amount of flows in this session.

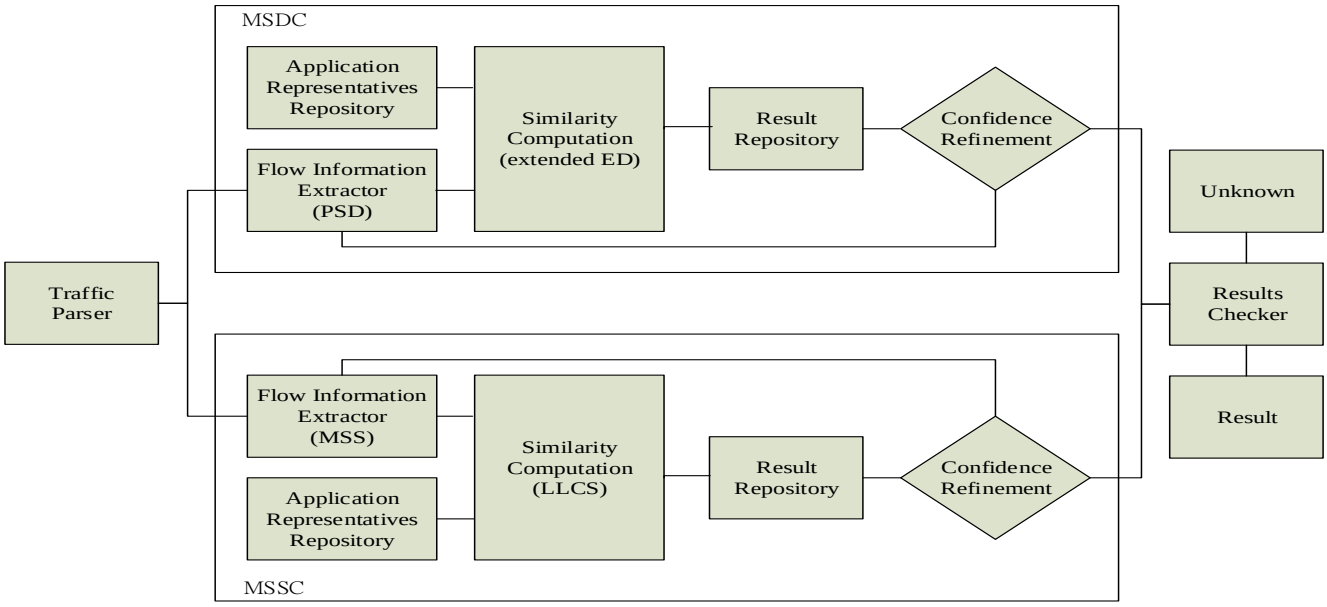

Figure 3. The overview of the hybrid solution architecture

B. Message size sequence classifier (MSSC)

MSSC also runs in two phases: an offline application representatives training phase and an online flow classification phase. Figure 2 gives an overview of the MSSC. The left and the right blocks represent the steps of the offline training phase and the online classification phase respectively.

The offline training phase uses a set of traffic traces and extracts applications' representatives from the five-tuple information, the size and the direction of each packet, and the message sequences (MSes) of all captured flows. Normally, a protocol/application message is sent by a packet, and hence packet sequences are another form of MSes.

The online flow classification mechanism compares the flows with pre-selected application representatives based on the message size sequences (MSSes) and classifies them into the application with maximal likelihood. The similarity distance is computed by finding a common subsequence in which the entries in the common subsequence appear in each of the two sequences; these entries must appear in the same order, but not necessarily consecutively. The longer the common subsequence we can find, the more similar the two sequences are.

IV. HARDWARE ARCHITECTURE AND METHODOLOGY

MSDC provides good accuracy, but it has a lower throughput because of its statistical computation overheads; MSSC has better throughput, but MSSC may be not accurate enough because of the occurrence of short common subsequences. Therefore, MSSC and MSDC are combined to seek a balanced solution in terms of classification accuracy and performance.

A. Methodology

Figure 3 illustrates the overview of the hybrid solution. Initially MSDC and MSSC run in parallel and the related flow information, the PSD, and the MSS are extracted and preserved. MSSC compares the MSS against all pre-defined application representatives, and MSDC computes the similarity distance against all application representatives. A metric, confidence, is defined as the ratio of the number of current received packets to the length of a flow representative. In later experiments, the value of confidence was set to 90%.

MSSC usually makes a decision in a very short time, but if it fails to make a classification, the decision is made by MSDC instead. If MSSC and MSDC both can't make a decision, the flow would be regarded as an unknown application flow.

The functionalities of each modules are described as follows.

- *Flow Information Extractor (FIE)* module, which is used to collect the number and the size of packet payload and output the PSD and MSS of the incoming flows.
- *Application Representative Repository (ARR)* module, which is used to store all representatives of possible behavior flows of pre-defined applications.
- *Similarity Computation (SC) module*, which is used to compute similarity distance between a flow and each representative stored in memory.
- *Result Repository (RR) module*, which is used to store all immediate valid results.
- *Confidence Refinement (CR) module*, which is used to check if the distinct sizes of packet sizes is equal to or larger than a user-defined threshold, 90% here, of the number of a representative. If yes, CR will output the final decision to the next module; otherwise, CR will continue to restart the similarity calculation from FIE by involving more incoming packets.
- *Result Checker (RC) module*, which is used to wait for the results coming from MSSC and MSDC. If both MSDC and MSSC can't make a decision, the flow would be labeled as unknown.

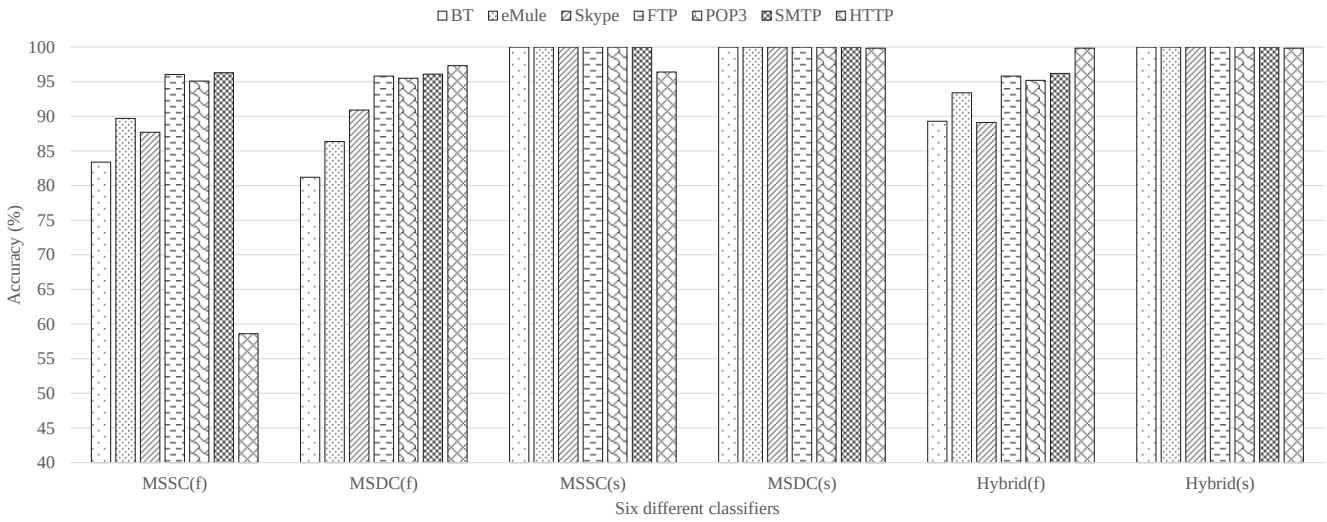

Figure 4. Accuracy rates for six classification methods

TABLE I
SUMMARIZED PROFILE OF PRE-SELECTED APPLICATION TRACES

Application Name	Application-Layer Protocol	TCP flows (training)	TCP packets (training)	TCP flows (testing)	TCP packets (testing)
BitTorrent	P2P	4172	194036	2241	104481
eMule	P2P	18569	920951	9994	453607
Skype	P2P	941	11943	508	5889
FTP	FTP	1965	361302	1308	230997
POP3	POP3	210	24158	140	15479
SMTP	SMTP	210	24407	140	14335
HTTP	HTTP	150	129866	100	93267

V. EVALUATIONS

Two different data sets were used; both were captured from the operational instances running in campus networks, not from a traffic generator or a lab. Data sets were split into two parts. One was for training and another for testing. The training data contained all pre-selected application traffic and was only used to develop application representatives. The testing data was used for the purpose of application classification. Table I shows the profile of the two data sets of each application. Individual and pure application traffic, marked as training data, was used to develop application representatives while the traffic, marked as testing data, were mixed together to evaluate the accuracy.

A. Parameters

The parameter tolerant threshold (*TT*) required by MSDC and MSSC affects the length of common subsequences and the accuracy of application classifications. Here, the value of *TT* and other two parameters of MSDC, port locality range and flow inter-arrival time are set to 4 and 500 seconds according to suggestion provided by [6].

B. Classification Accuracy

For the session-level classification, we further classified an unknown flow into a classified network flow by using the rules introduced by [6]. Figure 4 shows the classification accuracy for the six classification configurations: MSSC(f), MSDC(f), Hybrid(f), MSSC(s), MSDC(s), and Hybrid(s), which represents the flow-level classification with MSSC, flow-level classification with MSDC, flow-level classification with the hybrid solution, session-level classification with MSSC, session-level classification with MSDC, and session-level classification with the hybrid solution, respectively.

We found that some applications have similar accuracies regardless of the use of session grouping and application arbitration. This might be caused by those applications usually using only a single flow to communicate with other applications, or that the correlations among the flows of those applications are low or obscure. Based on our experiments, MSSC contributed 77.4% of decision rounds and MSDC contributed 22.6%.

C. Throughput

We simulated the hardware architecture of the hybrid classifier on an FPGA platform. The target device was Xilinx Virtex 5 XC5VLX50T with -3 speed grade. The simulator used was ISim and the simulation results were from Xilinx ISE 14.7 place and route reports. Our design was able to meet the timing constraints to achieve 250 MHz clock rate and the throughput obtained was 250 million packets per second, i.e. 80 Gbps for minimum size (40 bytes) packets.

D. Discussion

Based on these implementation and simulations, some interesting observations are discussed here.

- Throughput

This hardware design aimed to verify the feasibility and performance of a hybrid statistical classifier. More than 4000 2- or 4-bit gates were used, and no acceleration or optimization design was applied. Other than circuit optimization, parallelization seems another good candidate because incoming flows could individually compute the similarity distances among application representatives. Further, the number of memory access increased to 2000+ because the decision of a flow was changed whenever the flow statistics changed during

the life time. A cache-based mechanism could be applied to raise the throughput by avoiding an intensive memory access overhead.

- Table size vs. classification accuracy

The table size and the accuracy were dominated by the number of application representatives and how precise one representative could be. The more precise preliminary sketch for distinct types of behaviors of an application, the more accurate the final decision. In order to achieve the highest possible accuracy, the variance of a representative was almost exhaustively listed. If the table size is limited, the number of application representatives or the variance of a representative could be reduced, based on the expected accuracy.

- Encrypted traffic vs. unencrypted traffic

Compared to our results of unencrypted traffic, encrypted traffic was a little less accurate (88.18%). Because of the limited information on the effect of applied encryption techniques, the representatives were computed and derived blindly, and were difficult to verify. However, some interesting clues were found where *TT* should be refined because the sizes of unencrypted and encrypted packet payloads were different.

VI. Conclusion

A hybrid solution of combined MSSC and MSDC can provide a balanced solution for flow classification. A flow classification is by default made by MSSC. However, if MSSC is not able to make a decision, classification would be postponed until MSDC is able to make a decision. The session-level hybrid solution therefore achieves a classification accuracy of 99.97% and an overall system throughput of 723 Mbps. Simulations show that MSSC contributed 77.4% of decision rounds and MSDC contributed 22.6%. Our design can also an accuracy of more than 94% while achieving a throughput of 80 Gbps.

References

[1] N. B. Azzouna, F. Guillemin, "Analysis of ADSL Traffic on an IP Backbone Link," Proc. GLOBECOM'03, pp. 3742-3746, Dec. 2003.

[2] C.-N. Lu, C.-Y. Huang, Y.-D. Lin, Y.-C. Lai, "Session Level Flow Classification by Packet Size Distribution and Session Grouping," Computer Networks, vol. 56, no. 1, pp. 260-272, Jan. 2012.

[3] C.-N. Lu, C.-Y. Huang, Y.-D. Lin, Y.-C. Lai, "High Performance Traffic Classification based on Message Size Sequence and Distribution," submitted to Journal of Network and Computer Applications, 2015.

[4] M. Roughan, S. Sen, O. Spatscheck, N. Duffield, "Class-of-Service Mapping for QoS: A Statistical Signature-Based Approach to IP Traffic Classification," Proc. ACM SIGCOMM Conf. Internet Measurement (IMC'04), pp. 135-148, Oct. 2004.

[5] A. Moore, D. Zuev, "Internet Traffic Classification Using Bayesian Analysis Techniques," Proc. ACM SIGMETRICS Conf. Measurement and Modeling of Computer Systems (SIGMETRICS'05), pp. 55-60, June 2005.

[6] J. Kannan, J. Jung, V. Paxson, C. E. Koksal, "Semi-Automated Discovery of Application Session Structure," Proc. Sixth ACM SIGCOMM Conf. Internet Measurement (IMC'06), pp. 119-132, Jan. 2006.

[7] T. Karagiannis, K. Papagiannaki, M. Faloutsos, "BLINC: Multilevel Traffic Classification in the Dark," Proc. Conf. Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM'05), pp. 229-240, Aug. 2005.

[8] H. Song, et al., "Snort Offloader: A Reconfigurable Hardware NIDS Filter," Proc. FPL'05, 2005.

[9] J. Gonzalez et al., "Shutting: A Hardware/Software Architecture for Flexible, High Performance Network Intrusion Prevention," Proc. CCS'07, 2007.