On the Cloud-based Network Traffic Classification and Applications Identification Service

Nen-Fu Huang , Gin-Yuan Jai, Chih-Hao Chen
Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan
{nfhuang@cs.nthu.edu.tw; billjair@gmail.com;
knight@totoro.cs.nthu.edu.tw}

Han-Chieh Chao
Institute and Department of Electronic Engineering
National Ilan University
Ilan, Taiwan
{hcc@ilan.edu.tw}

Abstract— **Recently, the need of traffic classification and applications identification has attracted numerous research efforts. Based on statistical attribute analysis, recent research studies employed machine learning algorithms for building traffic classifiers. The machine learning based traffic classification achieves high accuracy and becomes prominent scheme. This paper proposes the framework of cloud-based traffic classification service for sharing model and parallel classification. A training tool is designed for a PC to collect the mapping of "statistical information" and each application running in the PC. This statistical information is sent to the cloud for training. In the cloud, a database is designed to collect these information and a machine learning based training system is constructed. For traffic and applications classification service, the tool in the network of a company or a campus will also send the "traffic statistics" to the cloud, which then identifies these traffic by using virtual machines and returns the identified results. This service platform is scalable as the cloud platform is used and virtual machines can be rent and managed dynamically. Also based this training model, we have the opportunity to train/identify the network applications as complete as possible.**

Keywords- Applications Identification, Cloud Computing, Machine learning algorithm, P2P, Traffic classification.

I. INTRODUCTION

Application traffic classification has attracted numerous research efforts. As the preprocessing module of network management devices, application traffic classification plays an important role in traffic management since the proper decision policy of traffic shaping or network intrusion defense is based on classification results. Nevertheless, the emergence of P2P file transfer applications and the mechanisms for protecting communication flows, such as encryption algorithms and tunneling transferring profoundly influences the performance of network traffic management. To support the P2P or encrypted flows classification, the methods based on statistical analysis of flow attributes have been studied in recent years. To acquire the potential statistical behaviors of each application protocol, the supervised, unsupervised and semi-supervised machine learning algorithms have been employed in recent studies for offline and online based traffic classification. The recent studies show that the high accuracy can be achieved using the machine learning based traffic classifiers. The machine learning based classifying schemes become prominent techniques for network management.

While the schemes of offline or real-time classifiers have been extensively investigated, the cloud-based training/testing scheme for traffic classifier is relatively unexplored. Most proposed studies are based on "single classifier" architecture. Each classifier operates independently. The model built from training phase must be separately dispatched to each classifier. The network administrator of large scale network cannot easily maintain the training sets, test sets, and models.

This study developed the service framework of traffic classification on cloud computing platform. Adopting the cloud computing platform as the basic traffic classification platform provides scalability, flexibility, model sharing and improvement, and parallel classification for network administrators to easily maintain the training data sets, deploy the network management devices, and improve the accuracy of traffic classification. A training tool is designed for a PC to collect the mapping of "statistical information" and each application running in the PC. This statistical information is sent to the cloud for training. In the cloud, a database is designed to collect these information and a machine learning based training system is constructed. For traffic and applications classification service, the tool in the network of a company or a campus will also send the "traffic statistics" to the cloud, which then identifies these traffic by using virtual machines and returns the identified results. This service platform is scalable as the cloud platform is used and virtual machines can be rent and managed dynamically. Also based this training model, we have the opportunity to train/identify the network applications as complete as possible. The main contributions of this study are the following:

1. The presented service framework provides the integrated environment for researchers and network administrators for easily sharing and maintaining data sets and models.
2. The presentation of basic architecture for scalable traffic classifiers on cloud platform.

The rest of this paper is organized as follows. Section II introduces works related to traffic classification. Section III discusses the framework of traffic classification service on

978-1-4673-0937-0/12/$31.00 ©2012 IEEE

cloud computing platform. Finally, Section IV draws conclusions.

II. Related Work

To classify encrypted application flows, machine-learning algorithms have been applied to statistical attributes captured in transport layers to characterize differences among categories of applications. [1] developed a flow-level classifier that applied a Naïve Bayesian classifier and Fast Correlation-Based Filter (FCBF) to summarize attributes of each TCP flow. Furthermore, [2] applied a support vector machine (SVM) algorithm for traffic classification to improve the test error and accuracy.

From the alternate perspective of normal flow attributes, simple heuristics have been developed to classify traffic by investigating intrinsically different behaviors among different categories of applications [3]. [3] investigated different behaviors among categories at three levels: the social level (identifying community), functional level (identifying server/client role), and application level (constructing communication topology of each host to identify application categories of that host). Moreover, the classifier inspects traffic at the host level rather than at the flow or packet level.

The necessity for real-time traffic monitoring and classification in recent years has also attracted several studies [4]-[10] to develop traffic identification methods by applying supervised, as well as unsupervised, and semi-supervised machine learning algorithms.

The first type of online classification method adopts supervised machine learning algorithms. [4] proposed two novel online classifiers based on the Neyman-Pearson classification and learning satisfiability framework to classify the traffic. The false alarm/discovery rates could be controlled with thresholds defined by the user. Five attributes were selected by applying the "correlation-based with best first search (forward)" method.

The second type of classifiers applies an unsupervised clustering algorithm with the advantage that no pre-defined classes are required before building the model. By applying the unsupervised clustering algorithm to the vector of the first k-data-packet size ($a_1, a_2, a_3, \ldots, a_k$) of each TCP flow, [5] has provided 80 % to 95 % average accuracy to identify traffic at the protocol level because applications have different packet sizes for control flows. Here the vector is the tuple of k variables ($a_1, a_2, a_3, \ldots, a_k$), and each variable a_i refers to size of the i-th packet. By employing the expectation maximization (EM) [6], *K*-means, DBSCAN, and AutoClass [7] algorithms with all or part of attributes: "total number of packets, mean packet size, mean data packet size, flow duration, and mean inter-arrival time of packets", the overall accuracy of classifying TCP flows at the category level can achieve 91 % for EM, 84 % for *K*-means, 75.6 % for DBSCAN, and 92.4 % for AutoClass, respectively. [8] developed the classifier that adopts DBSCAN clustering and the best first search feature selection (BFS). The feature set includes "mean and variance of packet size, mean inter-arrival time (IAT), and mean jitter of the first few packets of TCP/UDP flows". Moreover, the overall accuracy is above 85 %.

The third type of identification mechanism employs semi-supervised clustering algorithms that allow the training set containing both unlabeled and labeled data instances. For semi-supervised algorithms, the training set can include both known and unknown instances that are not labeled. The main idea of semi-supervised clustering is to build the model with the labeled data instances, and the unlabeled data instances are then used for model extension. [9] has applied semi-supervised clustering and layer modeling to 11 flow size attributes and has achieved more than 95 % accuracy at the category level.

Our previous study proposed a method to characterize both TCP and UDP flows from an application layer perspective [10]. Characterizing flow from application layer perspective shows more potential characteristics than the transport layer perspective does. All discriminators are normal statistical attributes available at the early stage.

Cloud computing technique is one of the prominent computing paradigms in recent years. The software services can easily scale up and scale down the computing resources to support the service level agreement.

Based on the above discussions, we can see that existing research in traffic classification does not investigate the need of cloud-based traffic classification schemes. Hence, in order to bridge this gap, this study develops the framework of traffic classification service on cloud computing platform.

III. Traffic Classification Services on Cloud Platform

A. Architecture of Traffic Classification Service on Cloud Computing Platform

Adopting the cloud computing platform as the basic traffic classification platform provides several advantages. First, the network management devices that are responsible for computing the statistical attributes of each flow can easily integrate with the cloud-based traffic classification platform. The attribute transmission size is small because most machine learning classification schemes adopt the feature selection algorithms to decrease the number of statistical attributes for classifying each flow. Second, putting the classification function on the cloud computing platform decreases the additional computation loading of network management devices. Third, the network administrators can share the different models on cloud computing platform. Network administrators can launch model improvement algorithms to increase the classification accuracy. Fourth, the scalable computation resources allow building model from single large training data set and concurrently supporting more traffic classification requests from different devices.

The cloud-based traffic classification service consists of several parts: the machine learning virtual machine (LVM), the learner VM management server, the classifier VM (CVM), and the classifier VM management server. The architecture and relationships among management servers, VMs, users, and network management devices are illustrated in Figure 1:

1. Machine learning virtual machine (LVM): the main tasks of machine learning virtual machine are to build and test model using training and test sets. Users can

select supervised, unsupervised and semi-supervised machine learning algorithms to build and verify models. For each VM, several server instances are simultaneously executed to concurrently serve multiple users.

2. Classifier virtual machine (CVM): each classifier virtual machine receives the flow classification requests from one or more network management devices.

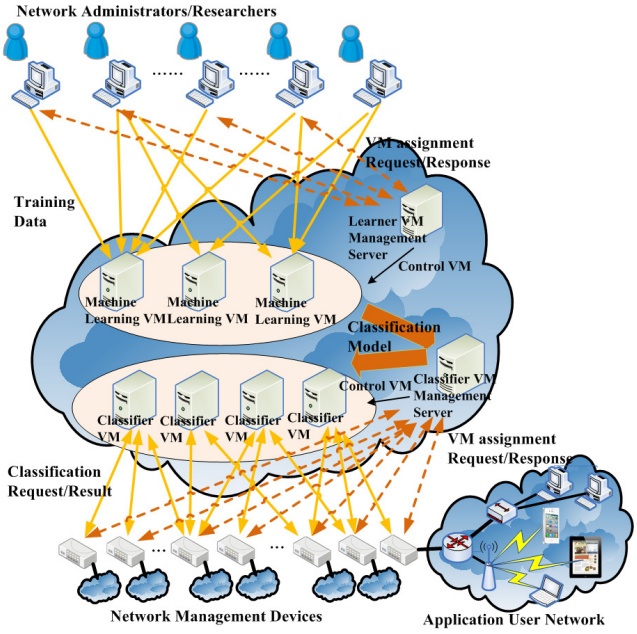


Figure 1. Basic architecture of traffic classification service on cloud computing platform.

3. LVM management server: the management server is responsible for monitoring, turning on and off the LVMs according to the computing resource needed for executing training requests. The training requests are first sent to the LVM management server, and then the server assigns one or more service instances on the LVMs to the network administrators/researchers for running the training tasks.

4. CVM management server: the tasks of classifier management server are monitoring, turning on and off the CVMs according to the computing resource needed for processing classification requests. The network management devices send the CVM assignment request to the CVM management server in the first step. The CVM management server selects one or more service instances on the CVMs for processing the flow classification requests, then responds to the network management device with the IP addresses of the selected VMs. After receiving the IP addresses of CVMs from CVM management server, the network management device can then send the flow attributes to the CVMs for classifying the flows.

5. Users: the users consist of network administrators and researchers. Each user can upload, maintain, and share data sets for training and model improvement. The client program can directly compute the attributes of flows of the client host and label the original application id of each flow by searching the process to port mapping information maintained by client host OS. To preserve the privacy, users can decide the sharing policy of data sets.

6. Network management devices: each network management device is responsible for computing the attributes of each flow, send the attributes to CVMs, and receive the result.

The parallel machine learning mechanisms have been investigated in Mahout [11] and GraphLab [12], so this study focuses on developing the framework of scalable classifiers and VM management.

B. The Training Service

The cloud-based traffic platform provides two traffic classification services: training service and classification service. To build the traffic classification rules, users employ the client program to send the labeled flow data to the training service. The basic operation steps of the training service are shown in Figure 2 (a) :

1. Requesting the training service: In the first step, the client program sends the training service request to the LVM management server to ask for the training service.

2. LVM assignment: after the LVM management server receives the training service request, the management server will check the loading and status of each service instance on the LVM. Then the LVM management server selects one training service instance on the LVM for serving the client.

3. Sending the reply message with the IP address and port number of the service instance: in the third step, the management server sends the IP address and port number of the selected service instance to the client.

4. Sending the training data: in the fourth step, the client connects to the selected service instance and uploads the training data.

5. Training phase: after the client uploads the training data, the service instance will begin the training phase.

6. Storing the rules and sending back the training result: when the service instance finishes the training phase, the generated classification rules are stored in the rule database.

C. The Classification Service

The basic operation steps of the training service are introduced as follows and illustrated in Figure 2 (b)-(c):

1. Requesting the classification service: In the first step, the client program sends the classification service request to the CVM management server to ask for the classification service.

2. CVM assignment: after the CVM management server receives the service request, the management server will check the loading and status of each service

instance on the CVM. Then the CVM management server selects one classification service instance on the CVM for serving the client.

3. Loading the classification rules: in the third step, the selected service instances load the rules from rule database for preparing to serve the client.

4. Sending the reply message with the IP addresses and port numbers of the service instance: in the third step, the management server sends the IP address and port number of the selected service instance to the client.

5. Sending the flow attributes and receiving the classification result (application id): in the fifth and sixth steps, the client sends the attributes of each flow to the service instances and receives the classification results from the service instances.

D. The program modules of training and classification client

The block diagrams of program modules for training and classification client are introduced as follows and depicted in Figure 3 (a)-(b), respectively:

1. Packet decoding module: the module is responsible for decoding the packet header fields. Then the decoded data fields are passed to the flow processing module.

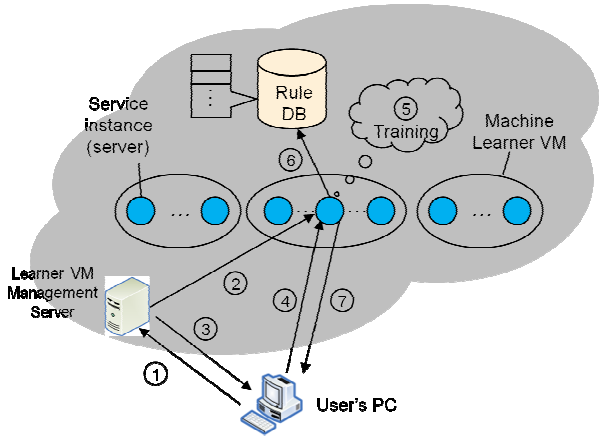
2. Flow processing module: the tasks of flow processing module are computing flow attributes and managing the flow table for storing the flow attributes.

3. Signature/port matching module and process-port mapping module: to construct the training labels (the "true" application class of each flow), the training client employed signature/port matching and process-port mapping methods to identify the original background truth. The process to port mapping information are maintained by the host OS, so the original application class can easily be retrieved for each flow from the client host OS. User can choose the signature/port matching or process-port mapping as the module to find the original application class for each flow.

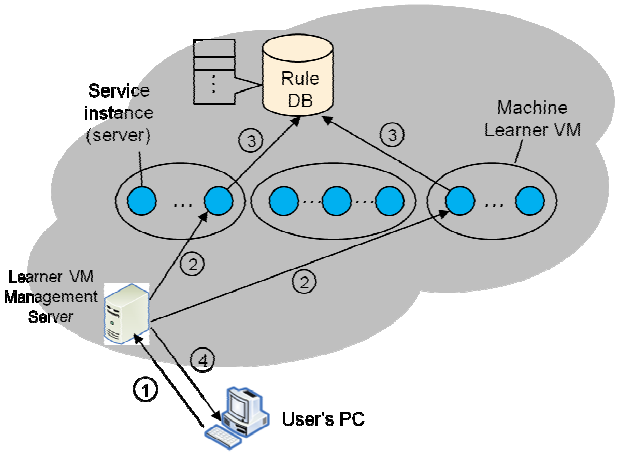
4. Sending the reply message with the IP addresses and port numbers of the service instance: in the third step, the management server sends the IP address and port number of the selected service instance to the client.

5. Training data management module: the task of training data management module is collecting the attributes of flows to form the training data set. Then the training data set is sent to the cloud communication module for uploading to cloud-based traffic classification platform.

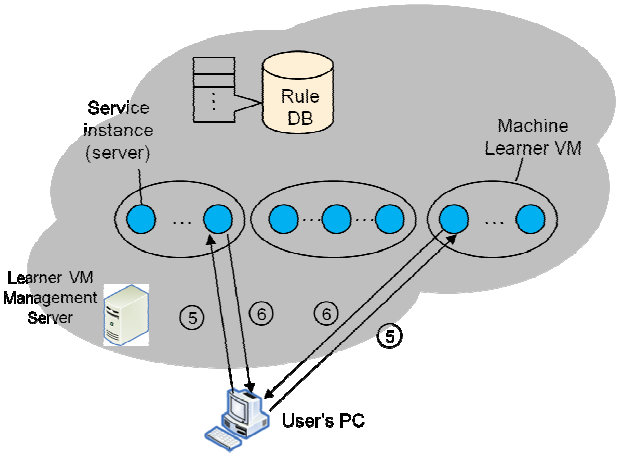
6. The cloud communication module in the training client: the cloud communication module in the training client is responsible for sending the training request to and receiving the IP address and port number of service instance of the LVM from the LVM management server. Besides, this module is also responsible for uploading the training data to and receiving the training result from the LVM.



(a) The operation steps of training service.



(b) The first four operation steps of classification service.

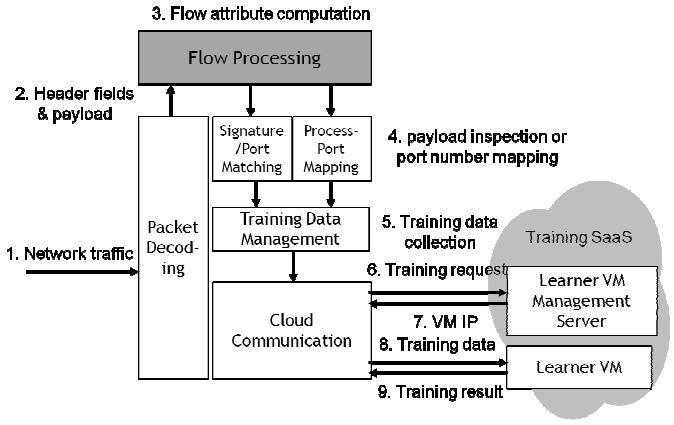


(c) The fifth and sixth operation steps of classification service

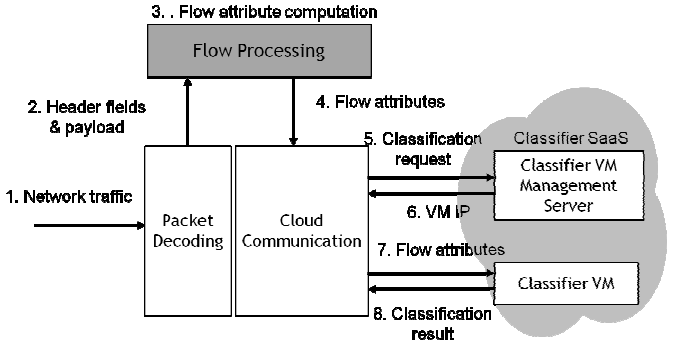
Figure 2. The operating steps of training and classification services.

7. The cloud communication module in the classification client: the cloud communication module in the classification client is responsible for sending the classification request to and receiving the list of IP addresses and port numbers of service instances of CVMs from the CVM management server. Besides,

this module is also responsible for sending the flow attributes to and receiving the classification result from the CVM.



(a) Training client



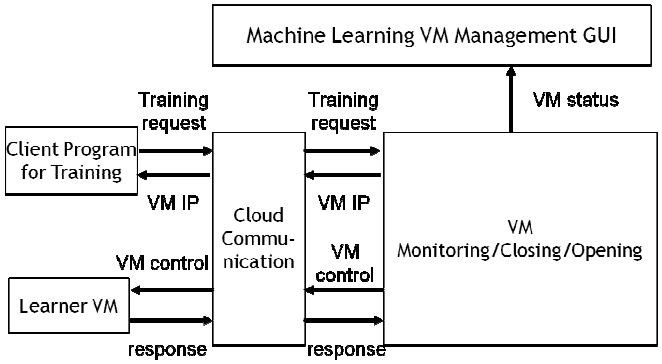
(b) Classification client

Figure 3. The program modules of training and classification client.

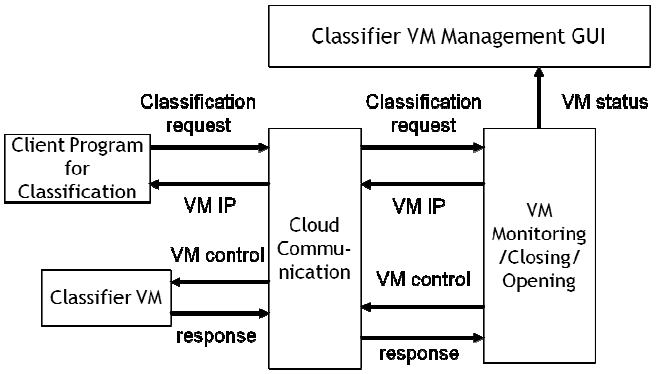
E. The program modules of LVM management server, CVM management server, LVM, and CVM

The block diagrams of program modules for LVM management server and CVM management server are illustrated in Figure 4 (a)-(b), respectively:

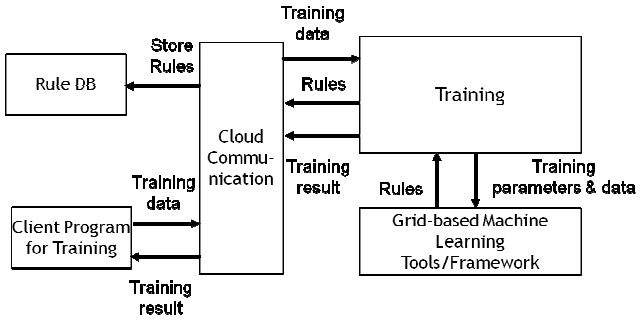
1. Machine learning/classifier VM management server graphic user interface (GUI): the main tasks of the VM management graphic user interface are to show the service status and statistical information to user, and to receiving the user's command.
2. VM monitoring/closing/opening module: to simultaneously serving multiple users with the limited sources, the VM monitoring/closing/opening module are responsible for monitoring status and loading of each LVM/CVM, creating the learner/ classifier based on the loading of VMs, and closing the VMs to collect the idle computing resources.
3. The cloud communication module in the LVM management server: the main tasks of the cloud communication module in the LVM management server are receiving the training request from and sending the IP address and port number of service instance of LVM to the training client. Besides, this module is also responsible for sending the control messages to and receiving the response messages from the LVM.



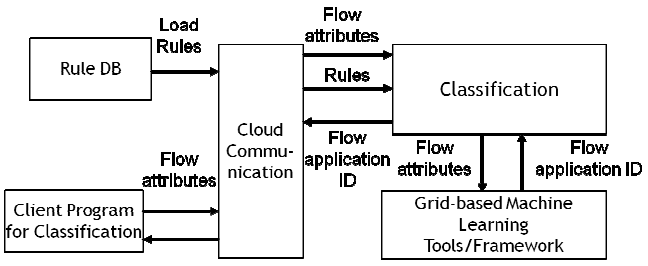
(a) LVM management server



(b) CVM management server



(c) LVM



(d) CVM

Figure 4. The program modules of LVM management server, CVM management server, LVM, and CVM.

4. The cloud communication module in the CVM management server: the cloud communication module in the CVM management server is responsible for receiving the classification request from and sending the list of IP addresses and port numbers of service instances of CVMs to the classification client. This module is also responsible for sending the control messages to and receiving the response messages from the CVM.

The block diagrams of program modules for LVM, and CVM are demonstrated in Figure 4 (c)-(d), respectively:

1. The training/classification modules: the training and classification modules employed the machine learning algorithms provided in Grid-based machine learning tools/framework to build the rules or classify the traffic flows.
2. The cloud communication module in the LVM or CVM: for the LVM, the main tasks of cloud communication module are to receiving the training data from training client, sending the training result to the training client, and storing the rules into the rule database. By contrasts, for the CVM, the main tasks of cloud communication module are to receiving the flow attributes from classification client, sending the classification result (application id of the classified flow) to the classification client, and loading the classification rules from the rule database.
3. Grid-based machine learning tools/framework: this module is the implementation of the machine learning algorithms. There are several tools or frameworks such as Gridweka [13], Gridweka2 [14], Mahout [11] and GraphLab [12] for grid-based environment. However, these frameworks don't monitor computing resources and dynamically open or close VMs. The VM management server is responsible for monitoring, opening and closing the VMs.

IV. CONCLUSION

A service framework of traffic classification on cloud computing platform has been proposed in this paper. The service framework includes training and scalable classifying traffic. A training tool is designed for a PC to collect the mapping of "statistical information" and each application running in the PC. This statistical information is sent to the cloud for training. In the cloud, a database is designed to store these information and a machine learning based training system is also constructed. For traffic and applications classification service, the tool in the PC of a company/campus network sends the "traffic statistics" of the collected flows to the cloud, which then performs the traffic and application identifications by using virtual machines and returns the identified results. We show that this service platform is scalable as the cloud platform is employed and virtual machines are rent and managed dynamically. Based this scalable training model, we have the opportunity to train/identify the network applications as complete as possible. The future work of this study includes the implementation of the scalable classifier architecture and traffic classification service on cloud computing platform.

Acknowledgements

This work was supported by National Science Council (NSC) of Taiwan under the grant numbers NSC-98-2221-E-007-060-MY3, NSC-98-2219-E-007-013, and NSC-99-2219-E-007-007.

REFERENCES

[1] A.W. Moore, D. Zuev, Internet traffic classification using Bayesian analysis techniques, in: Proc. ACM SIGMETRICS international conference on Measurement and modeling of computer systems, Banff, Alberta, Canada, 2005, pp. 50-60.

[2] Z. Li, R. Yuan, X. Guan, Accurate Classification of the Internet Traffic Based on the SVM Method, in: Proc. IEEE Int. Conference on Communications (ICC '07), Glasgow, Scotland, 2007, pp. 1373-1378.

[3] T. Karagiannis, K. Papagiannaki, M. Faloutsos, BLINC: multilevel traffic classification in the dark, in: Proc. ACM SIGCOMM conference on Applications, technologies, architectures, and protocols for computer communications, Philadelphia, Pennsylvania, PA, 2005, pp. 229-240.

[4] D. Nechay, Y. Pointurier, M. Coates, Controlling False Alarm/Discovery Rates in Online Internet Traffic Flow Classification, in: Proc. 28th IEEE Int. Conference on Computer Communications (IEEE INFOCOM 2009), 2009, pp. 684-692.

[5] L. Bernaille, R. Teixeira, K. Salamatian, Early application identification, in: Proc. ACM Conference on emerging Networking EXperiments and Technologies (CoNEXT 2006), Lisboa, Portugal, 2006, pp. 1-12.

[6] J. Erman, A. Mahanti, M. Arlitt, Internet Traffic Identification using Machine Learning, in: Proc. IEEE Global Telecommunications Conference (GLOBECOM '06), San Francisco, CA 2006, pp. 1-6.

[7] J. Erman, M. Arlitt, A. Mahanti, Traffic classification using clustering algorithms, in: Proc. ACM SIGCOMM workshop on Mining network data, Pisa, Italy, 2006, pp. 281-286.

[8] J. Zhang, Z. Qian, G. Shou, Y. Hu, Online automatic traffic classification architecture in access network, in: 9th International Conference on Electronic Measurement & Instruments (ICEMI '09.), Beijing, China 2009, pp. 3-24.

[9] J. Erman, A. Mahanti, M. Arlitt, I. Cohen, C. Williamson, Offline/realtime traffic classification using semi-supervised learning, Performance Evaluation, 64(9-12) (2007) 1194-1213.

[10] N.-F. Huang, G.-Y. Jai, H.-C. Chao, Early Identifying Application Traffic with Application Characteristics, in: Proc. IEEE Int. Conference on Communications (IEEE ICC '08), Beijing, China, 2008, pp. 5788-5792.

[11] Mahout project, http://mahout.apache.org

[12] Y. Low, J. Gonzalez, A. Kyrola, D. Bickson, C. Guestrin and J. Hellerstein. GraphLab: A New Framework for Parallel Machine Learning. In the 26th Conference on Uncertainty in Artificial Intelligence (UAI), Catalina Island, USA, 2010.

[13] GridWeka, http://userweb.port.ac.uk/~khusainr/weka/index.html

[14] GridWeka2,http://www.andreas-hess.info/projects/gridweka2/index.html