Proceedings of 2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN 2018)

A Survey of Traffic Classification in Software Defined Networks

Jinghua Yan
National Computer Network Emergency Response Technical Team / Coordination Center of China
Beijing, China
yjh@cert.org.cn

Jing Yuan
National Computer Network Emergency Response Technical Team / Coordination Center of China
Beijing, China
Corresponding author: yuanjing@cert.org.cn

Abstract—Traffic classification has been widely used in network management, service measurements, network design, security monitoring and advertising. Software defined networks (SDN) is an newly-developing technology, which is capable of address problems in the traditional network by simplifying network management, introducing network programmability, and providing a global view of a network. Recent years, SDN has brought new opportunity to classify traffic. Traffic classification techniques in SDN have been investigated, proposed and developed. This paper looks at emerging research into the traffic classification techniques in SDN. We first introduce SDN and related work of traffic classification, and then review several representative works of traffic classification in SDN. These works are reviewed in line with the choice of classification strategies and contribution to the literature. Research challenges and future directions for SDN traffic classification are also discussed.

Keywords—Software defined networks, traffic classification, machine learning

I. INTRODUCTION

Traffic classification is an intelligent process which categorizes traffic into different classes (such as protocols). Accurate classification of traffic flows is highly beneficial for quality of service (QoS) schemes, Dynamic access control, lawful Interception and so on.

Software-Defined Networking (SDN) is an innovative networking architecture, which represents the direction of the future network [1]. In SDN, the control plane is decoupled from the data plane. The control plane plays a role of centralized device which is in charge of routing and management policies. The data plane only forwards packets, which is programmable through a standardized protocol, such as OpenFlow [2]. Figure 1 depicts the architecture of SDN.

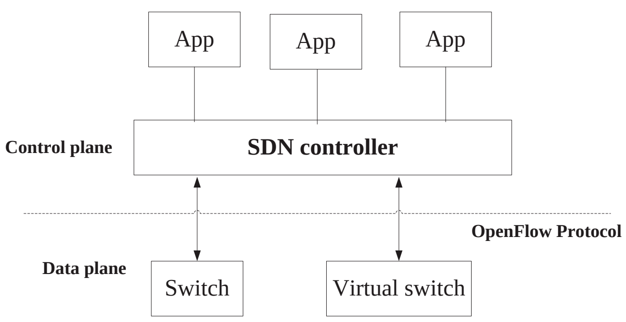

Fig. 1. SDN architecture

The SDN centralized architecture make the controller to have a central view of the network [3]. The characteristic of network programmability allows applications to program the network. In a word, the objective of SDN is to simplify network management, promote innovation, improve network resource utilization, and optimize network performance.

OpenFlow is the one of the most important southbound API protocols for SDN, which enables controller to communicate with the network switches[4]. It runs in software, and the data plane hardware. OpenFlow switches contain one or multiple flow tables with flow entries. Each flow entry includes match fields and actions and the tables are filled by the controller. The match fields consists of headers data of a packet, such as source and destination IP and MAC addresses, port numbers and other information. Each action determines instructions to the packet. The flow table of OpenFlow provides not only native flow features, such as IP and port number, but also a series of statistical features, such as number of packet and byte, duration and so on.

When a packet arrives at an OpenFlow switch, the switch looks up its flow table and executes the instruction such as dropping or forwarding to the packet. If there is no instruction in the flow table, the packet is sent to the controller, which is called packet-in. The controller decides the instruction for the packet based on its payload, header information and statistics. After the decision of instructions, the packet is sent back to the switch, which is called packet out. Meanwhile the flow table is updated. The succeeding packets of the same flow are processed according to the instruction.

The emergence of the SDN brought new possibilities to realize traffic classification and feature selection. The global view of controllers in SDN is simple to extract statistical features of network traffic from switches. Due to the mechanism design of OpenFlow protocol, it can be customized to gather flow features and indicates forwarding policies to each switch. Therefore it is expected to be very suitable for the implement of traffic classification. Nowadays, SDN is only capable of generate low-layer (Layer 2-4) policy. By means of identifying the packets generated by an application, it can implement higher-layer (Layer 7) policy. SDN allow users to construct software that carry out traffic analysis without any requirement to install physical hardware. The software can be deployed on the SDN controller layer as additional application. If the SDN controller learned traffic information after some initial number of packets, it can generate forward instruction to the packets of flow immediately.

The paper is organized as follows. Section II provides an overview of the related work of traffic classification. Section III reviews the related work of traffic classification in SDN

978-1-5386-4870-4/18/$31.00 ©2018 IEEE

in recent years. Section IV discusses the future directions. Section V concludes this paper.

II. Review of Traffic Classification Techniques

In this section we outline three broad categories to review significant works of traffic classification.

A. Port Based Traffic Classification

In the early stage of Internet, most of protocols used well-known port numbers assigned by IANA (The Internet Assigned Numbers Authority)[5]. In this situation, the port based techniques can finish the classification tasks perfectly. However, more and more protocols and applications used random or dynamic port numbers to hide from network security tools. Thus the port based techniques are no longer useful. Recently years, several evaluation experiment results have shown the port based methods are not very efficient. Moore et al. observed that no more than 70% accuracy for port based techniques by using IANA list [6]. Madhukar et al. found that port-based techniques can't identify 30-70% of traffic flows they investigated.

B. Payload Based Traffic Classification

To overcome the limitation of port based classification techniques, payload based techniques were proposed. The payload approaches are also known as Deep Packet Inspection (DPI). They classify traffic by inspecting packet payloads and comparing with the known signatures of protocols [7~10]. Several DPI tools have been widely used such as L7-filter [11] and OpenDPI[12]. Techniques based on DPI can perform traffic classification accurately, but lead to significant complexity and high computation cost. And DPI techniques can be difficult when dealing with encrypted or proprietary protocols. Gringoli L. Salgarelli et. al evaluate the performance of DPI methods, and the experimental results show that DPI tools such as L7-filter only correctly classify 67.73% of bytes on UNIBS data set and 58.79% of bytes on POLITO data set respectively [13]. Furthermore, The DPI methods require manual maintain signature to keep up-to-date with application semantics, while sometimes it is hard or impossible to obtain signatures with the evolution of network applications.

C. Machine Learning Traffic Classification

Recently, Machine learning (ML) based traffic classification techniques have been widely used to alleviate limitations imposed by traditional traffic classification techniques [14~15]. The assumption of ML methods is that network traffic flows have statistics characteristics (such as the distribution of packet lengths, packet inter-arrival time and so on) that are unique for a specific kind of protocol or application. ML based approaches distinguish different applications by their statistics features [16]. The ultimate goal of ML based techniques is either classifying different applications or clustering traffic flows into groups that have similar patterns. ML based approaches do not require packet payload inspection, thus results in a lower computational cost than DPI based solutions and can identify encrypted traffic.

ML based techniques include several steps. First, features are extracted by calculating over multiple packets of flows (such as packet lengths, flow duration or inter-packet arrival times) [17]. Then features are refined by feature selection algorithms if possible. Then the ML model is trained to to generate classification rules, and apply the ML algorithm to classify unknown traffic flows using previously trained ML model.

Generally speaking, a unique flow is defined by five tuple information (IP source, IP destination, the source port, the destination port, transport layer protocol). Features of traffic flow can be divided into two classes according to their observation levels :(1) flow-level features; and (2) packet-level features. The flow-level features are usually calculate after the flow have completed, such as number of packets, flow duration, mean packet size of a flow. On the contrary, the packet-level features can be obtained by early stage of flow, such as packet length, inter-arrival time and the packet direction of first few packets of flow. The intuition of packet-level feature is that the first few packets constitute the negotiation phase of application, which are descriptive features. Peng et. al conduct extensive experiments, and the experimental results show that 5~7 are the best packet numbers for early traffic identification [18]. Recent years, it is a hot topic to classify traffic at its early stage. A timely classifier should draw a conclusions using as few packets as possible rather than waiting until the flow completes.

The quality of the feature is significant to the performance of ML algorithm. Using redundant features often reduces the accuracy of most ML algorithms, which can also increase the system computationally cost[19]. Consequently it is important to select optimal set of features to characterize different types of traffic which describes essential information about the classes of interest. Feature selection algorithms can be divided into filter method or wrapper method. A number of feature selections techniques have been widely used, e.g. Correlation-based Feature Selection (CFS) , Fast Correlation Based Feature Selection (FCFS), information gain (IG), principal component analysis (PCA) and so on [20].

ML algorithms can be divided into three categories: supervised ML, unsupervised ML, and semi-supervised ML . Supervised ML constructs models that classifying new instances into known classes. There are two phases in supervised learning: The training phase that builds a classification model by analyzing the training dataset (instances with features and ground truth of classes). The testing (or classifying) phase use the model built in the training phase to classify new instances. The framework of supervised ML is demonstrated in Figure 2.

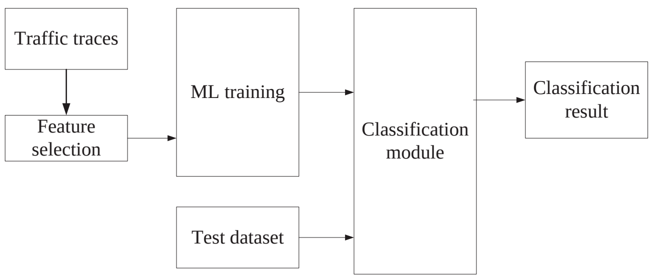

Fig. 2. Framework of supervised ML algorithm

There exist a number of supervised ML algorithms such as Naive Bayes, C4.5 Decision Tree, Bayesian Network, Support Vector Machine(SVM), Neural Networks and so on. Recent years, supervised ML algorithms have been widely used for traffic classification and achieve promising results [21~26]. Both of the training and testing phases of supervised ML must be executed using datasets that have been labeled accurately. The supervised ML approach can achieve high classification accuracy for known applications in this context, while it cannot detect unknown applications. However sometimes it is difficult in obtaining accurately annotated data.

To overcome these drawbacks of supervised ML, unsupervised ML techniques are proposed. Unsupervised ML is used for clustering tasks, in which the algorithms group the unlabeled data into different clusters according to similarities, and then uses the clusters to build a classifier. Frequently used unsupervised ML algorithm includes KMEANS, DBSCAN and so on. In the traffic classification fields, a number of related works take use of unsupervised ML techniques [27~31]. Unsupervised ML schemes can automatic discovery the patterns with unlabeled dataset. However, the constructed clusters still need to map to applications. The number of clusters is always much larger than the number of applications, which leads to challenges for traffic classification task.

A combination of supervised and unsupervised ML methods is semi-supervised ML methods, which can be used with both labeled and unlabeled data. This approach can perform with a dataset where the majority of the instances are unlabeled and the minority instances are labeled. Semi-supervised ML methods have been proposed for several years, and have been applied for traffic classification [32~35]. The semi-supervised ML methods are promising for traffic classification, and worth in-depth study.

The results of ML algorithms can be divided into two categories according to classification level: coarse-grained and fine-grained. The coarse-grained classification identifies traffic flow into several classes based on rough traffic type (web browser, bulk transfer, VOIP and so on). However, it is increasingly important to classify network traffic with fine-grained results. Fine-grained classification aims to distinguish individual application or function model rather than classifying them into a rough traffic class, which is in favor of analyzing traffic composition and user profiles.

III. REVIEW OF TRAFFIC CLASSIFICATION IN SDN

With the rapid development of SDN, more and more researchers study traffic classification in SDN. This section first gives review of related works and summarize in TABLE I, then introduced works based on optimizing existing architecture to improve scalability of SDN classification.

In this subsection, we first introduce a general framework of traffic classification in SDN, which is demonstrated in Figure 3. The framework comprises the following steps: (1) The switches collect network traffic flows, and send to controller by OpenFlow protocol. (2) The controller compute the features. If it is possible, a feature selection algorithm is used to extract the most meaningful features. Then the controller sends the flow features to the classifier. (3) The classifier build classification module and classifies the flows. The classifier can support the implementation of a variety of classification algorithms. When the classification results have been generated, the classifier sends flow tables to switches. The classification results can also be shared with other modules or applications through northbound API.

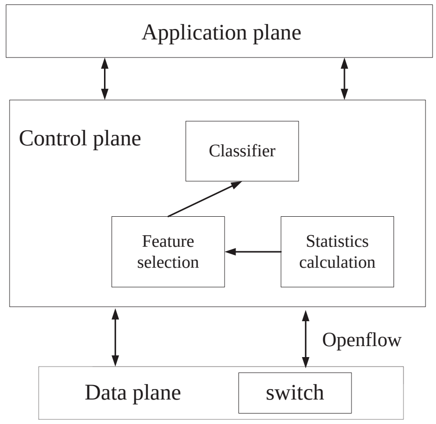

Fig. 3. General framework of traffic classification in SDN

A. Traffic Classification solutions in SDN

Anderson Santos da Silva et al. proposed a framework to collect and select a set of flow features for accurate traffic classification in SDN [36]. The features are generated from native OpenFlow counters. They extend higher level features by calculating mathematical statistics and Fourier Transform. The new features and can be classified into three categories: (1) statistical features which are mean, variance of traffic characteristics; (2) scalar features which are minimum and maximum values, flow duration; and (3) complex feature which are the Discrete Fourier Transform (DFT) of packet characteristics. To find an optimal subset of features for traffic classification, they adopt principal component analysis (PCA) and genetic algorithm (GA). The classification algorithm used in this work is SVM. To obtain dataset for validation, they set up experimental environment with three levels comprising switches and hosts. The experimental results show that the classification accuracy rates of PCA and GA schema is above 91% and 88% respectively, and all of the performance with optimized traffic features is superior to that of using complete features. The main contribution of this paper is that it provides an plentiful set of traffic flow features in SDN and refines features to find the optimal features by PCA and GA algorithms. However, the features are flow-level features, and required complicated computation because of DFT. Therefore leads to failure to handle traffic classification real-time. Meanwhile the proposed method studies DDoS, FTP, and Video Streaming traffic which doesn't verify its efficiency on individual applications of internet.

Peng Xiao et al. introduced a traffic classification method in SDN, which is based on an unsupervised ML technique named spectral clustering [37]. The solution extracts flow features by scanning the flow tables in SDN controller. And the flow features are clustered with spectral analysis. They conduct their experiments on WIDE data set and data center data set , and the traffic protocols mentioned in this paper include http, smtp, ssh, p2p, dns. They compare their spectral clustering method with traditional KMENAS method on WIDE dataset, and the experimental results demonstrate that the proposed method outperforms KMEANS in term of accuracy and class recall. This paper also evaluated the classification time of these techniques on data center data set, and results show that spectral clustering method spends less time than KMEANS. However, the work doesn't describe the detailed features used for classification.

Pedro Amaral et al. present a traffic classification architecture based in SDN that allows the collection of data both in legacy networks and in SDN networks [38]. This study gather data by OpenFlow protocol and the extracted features include packet size, interval arrival time, source and destination IP/MAC/Port, flow duration, byte count, packet count of the first five packets,. To reduce the redundancy of features, they perform principal component analysis (PCA) algorithm to find the optimal components. The ML method used in this work is ensemble learning method, in which several basic classifiers take a weighted vote of classification result. They adopt three supervised classification methods: Random Forests (RF), Stochastic Gradient Boosting (SGB) and Extreme Gradient Boosting (EGB). They deploy this platform in a enterprise network, and collect dataset which include the following applications: Youtube, Vimeo, Facebook, Linkedin, Skype, Bittorrent, Web Browsing (HTTP) and Dropbox. The experimental results are promising, and each classification approach obtained accuracy above 90% on average with the each application. This method gathers information from the first five packets, which guarantees real-time traffic classification.

Li et al. introduce a classification architecture in SDN called MultiClassifier, which combined DPI and ML techniques [39]. Multilassifier takes advantages of these two classifiers to maintain satisfactory accuracy rate while achieve a high classification speed. MultiClassifier is deployed on the controller, and one of the kernel modules of MultiClassifier is named selector. When a new flow arrives, the selector first sends it to ML classifier. According to the ML result, selector will decide whether choose DPI for further classification. The ML result is compared to a threshold value. If the result exceeds the threshold, MultiClassifier adopts the ML result. On the contrary, if the result is under the threshold, they will take use of DPI approach for further classification. To balance between accuracy and efficiency, there is a policy to update the threshold value. They capture traffic data from the campus network and perform MultiClassifier on this data set. The Experimental results reveal some conclusions: (1) DPI method obtains the highest accuracy, and the performance is the worst. (2)ML method has relatively low accuracy rate but maintain the best performance.(3)By combining DPI and ML method, Multiclassifier can obtain a accuracy of more than 85% and achieve high classification speed . This work proposed a novel classification framework in SDN by combining DPI and ML classifier. But it doesn't point out the features and the specific ML algorithm, and doesn't mention the applications contained in experimental data set.

Zafar Qazi et al. present a framework, Atlas, which enables fine-grained application classification in SDN [40]. They design an automated method for obtaining ground truth to avoid manual labor cost. It also leverages OpenFlow protocol to gather flow data. Atlas employs ML based classification approach to incorporate fine-grained application-awareness in SDN. The proposed method collects the packet size of the first N packets of a network flow, which means that it can classify network traffic real time. Atlas is deployed in wireless network and gathers network traffic through Android OS. They adopts C5.0 decision tree algorithm for classification, and the experimental results show the accuracy of Atlas is 94% on average for top 40 Android applications. This work proposed an intelligent framework, which not only leverages Openflow protocol but also label the ground truth of data automatically. They evaluate their fine-grained classification approach for several Android applications, and achieve satisfactory results. We think it is expected to implement Atlas with other applications based on different platform.

Mohammad Reza Parsaei et al. proposed application recognition of network flows with the help of ML techniques in SDN [41]. They used four Neural Network estimators to classify traffic application. The proposed four classification methods are: feedforward, Multilayer Perceptron (MLP), NARX (Levenberg-Marquardt) and NARX (Naïve Bayes). This method calculates features based on full-flow, which includes five tuple information, average packet size, mean number of packets, mean number of bytes of a flow. This work collects data set with the following protocols: instant messaging, video streaming, FTP, HTTP and peer to peer protocol. These four classification scenarios respectively obtain accuracy of 95.6%, 97%, 97% and 97.6%. This method has to collect information from the whole flow. Such requirements might prevent the application of early traffic classification.

Pu Wang et al. proposed a framework to classify the network traffic into different QoS classes [42], such as interactive video gaming or bulky data transfer. The proposed architecture includes two components: (1) the local traffic classifier at switches. (2) the global traffic classifier at the network controller. The local traffic classifier is used for detecting "elephant" flows which occupy more than K% bandwidth of network. The latter component performs the traffic classification to divide into different QoS classes. The framework combines DPI and semi-supervised ML algorithm name Laplacian SVM to achieve accurate classification. They take use of DPI to identify known applications, while the semi-supervised algorithm provides classification for unknown applications. The features adopted in this paper include: statistical value of inter-arrival time, packet length, five tuple of first 20 packets in a flow. To evaluate the proposed approach, they perform traffic classification on the real internet data. The traffic flows of this data set were categorized into 4 QoS classes such as voice/video conference, streaming, interactive data, and bulk data transfer. The simulation results show the proposed classification approach can provide promising performance. The classification accuracy exceeds 90%, which is better than other semi-supervised ML techniques using KMEANS. And the communication costs between switch and controller are acceptable. This work focuses on coarse-grained traffic classification for QoS instead of fine-grained application classification. Because the features are derived from the first 20 packets of flow, the framework can identify traffic in a real-time manner.

Ng, B et al analyzed SDN technique from the perspective of traffic classification [43]. They developed an SDN based traffic classification platform of enterprise network. The platform named "nmeta" which is built with openly SDN controllers and commercially commodity hardware. Similar to ref [39], Nmeta combines multiple classifiers which can be depicted as a multiclassifier. Classifiers can be general or specific. Meanwhile nmate employs the combination of both software classifiers and hardware classifiers with the programmability through SDN. They also outline the practical experiences and lessons learned from this work.

.

TABLE I. SUMMARY OF REVIEWED PAPER

Work	Classification algorithms	Features	Traffic considered	Comments
Santos da Silva et al.[36]	Support Vector Machine	Packets , packet length, inter-arrival time, flow duration, flow size, Fourier transform of inter-arrival time of full flow Adopt PCA and GA feature selection algorithm	DDos attack, FTP traffic, Video streaming	Coarse grained classification Doesn't support real-time classification
Peng Xiao et al.[37]	Spectral Clustering	Not mentioned detailed features	WIDE traces, data center traces http, smtp, p2p, ssh, dns, ssl	Coarse grained classification
Pedro Amaral et al.[38]	Random Forests ,Stochastic Gradient Boosting, Extreme Gradient, Boosting	Packet size, inter-arrival time, src/dst MAC, src/dst IP, src/dst port, flow duration, byte, packets of full flow	Enterprise network traces Youtube, Video, Facebook, Linkedin, Skype, Bittorrent, Web Browsing(HTTP) and Dropbox.	Fine grained classification Doesn't support real-time classification
Y. Li et al.[39]	Combination of DPI and ML method	Not mentioned detailed features	Not mentioned	N/A
Zafar Qazi et al.[40]	C5.0 Decision tree	First N packet size	Wireless network traces Android applications such as facebook, Google+,twitter and so on	Fine grained classification Support real-time classification
Reza Parsaei et al.[41]	Feedforward, Multilayer perceptron, Levenberg-Marquardt ,Naive Bayes	Packets size, packets, bytes, src/dst ip, src/dst port, protocol of full flow Adopt PCA feature selection algorithm	Stream, P2P, http, ftp, instant message	Coarse grained classification Doesn't support real-time classification
Pu Wang et al.[42]	DPI and Laplacian SVM	Inter arrival time, packet length, src/dst port,, hurst parameter of the first 20 packets	Broadband Communication Research Group in UPC traces, skype, ftp, gaming, ppstream...	Coarse grained(Qos classes: streaming, voice, data transfer) Support real-time classification

B. *Scalable traffic classification methods*

Traffic classification in SDN send packets to a control plane and requires packet inspection, which may lead to resource exhaustion on control plane, and degrade network performance, thus limit scalability. To address these limitations, several researchers proposed related works to enhance the scalability of traffic classification in SDN.

Matthew Hayes, et al propose a scalable traffic classification architecture based on OpenFlow [44].The kernel component of this architecture is a device called DPAE(a data plane auxiliary engine). The main advantage of DPAE is that it resides on the data plane and offloads traffic classification workload from the control plane. The scalability of DPAE is decoupled from the controller. The authors further leverage OpenFlow multiple flow table (MFT) feature to improve performance. The experimental results demonstrate the proposed architecture can improve the performance and scalability of traffic classification in SDN.

Andrea Bianco et al. exploit a stateful SDN approach for traffic classification [45]. They leveraged OpenState to realize a state machine in the OpenFlow switches in order to mirror just the minimum number of packets to the packet classifier. OpenState is an extension of OpenFlow. The contribution of proposed approach is minimizing the interaction between the switch, the controller and the traffic classifier. They design two stateful solutions named SCD and CCD respectively. Both of them can minimize the total memory occupancy of flow tables of the switches. Finally, they validate these solutions and estimate the memory required for the flow tables, and experiment results demonstrate CCD solution outperforms SCD solution in terms of memory occupancy.

Shota Ogasawara et al. propose a mathematical model to evaluate the performance of traffic classification in SDN [46]. The model is analyzed by using queuing theory under approximation assumptions. They derive the mean and the Coefficient of Variation of setup delay of a flow, which is time from the arrival of a flow until the instruction is updated, and the mean number of flows stored for classification. The experiment show analytical results agree well with simulation results

IV. CHALLENGES AND FUTURE DIRECTIONS

Traffic classification in SDN has obtained a lot of achievements, but there are still several challenges to be solved. In this section, we would like to analyze some general challenges and recommendation of traffic classification in SDN.

A. *Large scale traffic classification*

With the great progress of network, network bandwidth increased year by year. The traffic classification systems should process Gigabits per second in some cases, which is a great challenge for traffic classification. There is still a lot of room for further research in the field of large scale traffic classification in SDN. We suggest a promising solution for this challenge is to employ specific hardware or parallel processing architecture. Currently, several related works have taken use of Field Programmed Gate Array (FPGA) and Graphics Processing Units (GPUs) to deal with large scale traffic classification. The parallel processing is a efficient solution for real time classification to process lots of traffic flows simultaneously. Another solution is to reform the architecture of SDN to improve the scalability of SDN classification, and several related works have proposed practical techniques [44~45].

B. *Fine-grained traffic classification*

Nowadays, internet applications became more sophisticated with the development of network. More and more protocols carry different applications. For example, QQ is not a simple instant messaging protocol, but also employs Chat, Video, VoIP, File transmission,, Email, game and so on. It is quite essential to perform fine-grained classification of these specific applications. Traditional approaches tend to classify traffic based on protocol, which is coarse-grained classification. For example, these techniques can classify P2P traffic flows, but are unable to distinguish specific P2P applications such as BitTorrent and eMule. We think the fine-grained classification is a further research direction which is beneficial to analyze the composition of network, understand user profile, and provide better QoS.

C. *Network traffic evolution and new coming protocols*

As network technologies continue to be mature and evolve, protocols and applications are developed and evolved, and more and more new coming protocols emerged. Recently, most of existing techniques classify network traffic based on static analysis model (static features and static training model).. To identify new coming protocols, it is suggested to find the new traffic mode automatically. An adaptive traffic classification method is required, which should update classification model dynamically to classify evolution network traffic.

D. *Robust traffic classification*

A robust traffic classification solution can execute perfectly in various kinds of network. However, network always encounter abnormal phenomenon such as packet loss, retransmission, duplication, fragmentation and jitters. Almost all of related work assumes the network condition is perfect and can obtain whole information of traffic flow. For example, the packet size, inter-arrival time, payload of the first packets. These work will lose efficiency when deal with the above abnormal conditions. We think a robust traffic classifier is quite important to deploy on network.

V. CONCLUSIONS

Traffic classification is an important and traditional research area. With the rapid development of SDN, more and more researchers pay attention to traffic classification in SDN. In this paper, we first present an overview of the existing traffic classification techniques, then survey the related works of traffic classification in SDN. We also present some general challenges from the perspective of traffic classification, and finally outline some recommendations.

ACKNOWLEDGMENT

This work was supported by National Natural Science Foundation of China (NO. 61771410)

REFERENCES

[1] J. A. Wickboldt, W. P. de Jesus, P. H. Isolani, C. B. Both, J. Rochol, and L. Z. Granville, “Software-Defined Networking: Management Requirements and Challenges”, IEEE Communications Magazine, vol. 53, no. 1, pp. 278-285, Jan 2015.

[2] Nunes, Bruno, et al. "A survey of software-defined networking: Past, present, and future of programmable networks." Communications Surveys & Tutorials, IEEE 16.3 (2014): 1617-1634.

[3] T. Huang, F. R. Yu, C. Zhang, J. Liu, J. Zhang, J. Liu, “A Survey on Large-scale Software Defined Networking (SDN) Testbeds: Approaches and Challenges”, IEEE Communications Surveys & Tutorials, vol. 19, no. 2, pp. 891-917, 2016.

[4] F. Hu, Q. Hao, K. Bao, “A survey on software-defined network and openflow: From concept to implementation”, IEEE Communications Surveys & Tutorials, vol. 16, no. 4, pp. 2181-2206, 2014.

[5] IANA. Internet assigned numbers authority. http://www.iana.org/assignments/port-numbers.

[6] Moore A, Papagiannaki K. “Toward the accurate identification of network applications”, Proceedings of Passive and Active Measurement Workshop (PAM2005). Boston(USA), 2005.

[7] Finsterbusch M, Richter C, Rocha E, et al. “A Survey of Payload-Based Traffic Classification Approaches”, Communications Surveys & Tutorials IEEE, 2014, 16(2):1135-1156.

[8] S. Sen, O. Spatscheck, and D. Wang. “ccurate, scalable in-network identification of p2p traffic using application signatures”. In WWW 2004, pages 512-521. ACM, 2004.

[9] Goo Y, Shim K, Lee S, et al. “Payload signature structure for accurate application traffic classification” , Asia Pacific Network Operations and Management Symposium, 2016: 1-4.

[10] Zhe Fu, Zhi Liu and Jun Li. “Efficient Parallelization of Regular Expression Matching for Deep Inspection.”, Proc. of the 26th International Conference on Computer Communication and Networks (ICCCN), 2017.

[11] L7 Filter. http://l7-filter.sourceforge.net.

[12] OpenDPI: https://github.com/thomasbhatia/OpenDPI

[13] F. Gringoli, L. Salgarelli, M. Dusi, N. Cascarano, F. Risso, K. Claffy, “GT: picking up the truth from the ground for Internet traffic”, ACM SIGCOMM Compute. Communication. Review. 39 (2009) pp.13-18.

[14] Alberto Dainotti, Antonio Pescap ́ e, and K.C. Claffy, “Issues and future directions in traffic classification”, Network, IEEE, 26(1), 35-40, January-February 2012.

[15] Xue, Yibo, Luoshi Zhang, and Dawei Wang. "Traffic Classification: Issues and Challenges." Journal of Communications 8.4 (2013): 240-248.

[16] I. Witten and E. Frank, “Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations (Second Edition).” Morgan Kaufmann Publishers, 2005.

[17] Hongli Zhang, Gang Lu, Mahmoud T. Qassraweni, Yu Zhang, Xiangzhan Yu. "Feature selection for optimizing traffic classification." Computer Communications 35.12, 2012: pp.1457-1471.

[18] Lizhi Peng, BoYang,Yuehui Chen. “Effective packet number for early stage internet traffic identification”. Neurocomputing. (2015)252-267.

[19] Shi H, Li H, Zhang D, et al. “Efficient and robust feature extraction and selection for traffic classification”, Computer Networks, 2017: 1-16.

[20] Fahad A, Tari Z, Khalil I, et al. “Toward an efficient and scalable feature selection approach for internet traffic classification”. Computer Networks, 2013, 57(9): 2040-2057.

[21] A.Moore and D. Zuev, “Internet traffic classification using Bayesian analysis techniques”, in ACM International Conference on Measurementand Modeling of Computer Systems (SIGMETRICS) 2005, Banff.

[22] J. Zhang, Y. Xiang, Y. Wang, W. Zhou, Y. Xiang, Y. Guan, “Network traffic classification using correlation information”, IEEE Trans. Parallel Distrib. Syst. (2012).pp1‑15.

[23] Chun-Nan Lua, Chun-Ying Huanga, Ying-Dar Lina, Yuan-Cheng Laib. “High performance traffic classification based on message size sequence and distribution”. Journal of Network and Computer Applications. (2016) 60‑74.

[24] Zhang J, Chen C, Xiang Y, et al. “Internet Traffic Classification by Aggregating Correlated Naive Bayes Predictions”. IEEE Transactions on Information Forensics and Security, 2013, 8(1): 5-15.

[25] Camacho J, Padilla P, Garciateodoro P, et al. “A generalizable dynamic flow pairing method for traffic classification” . Computer Networks, 2013, 57(14): 2718-2732.

[26] Qiong Liu, Zhen Liu, Ruoyu Wang, Changqiao Xu. “Large Traffic Flows Classification Method”. International Conference on Communications, 2014: 569-574.

[27] L. Bernaille, R. Teixeira, and K. Salamatian, “Traffic Classification on the Fly”, ACM SIGCOMM Computer Communication Review,Vol.36, 2006, No.2, 23-26.

[28] J. Erman, M. Arlitt, and A. Mahanti. “Traffic Classificaton Using Clustering Algorithms”,. In ACM SIGCOMM MineNet Workshop, September 2006.

[29] Baer A, Casas P, Alconzo A D, et al. “DBStream: A holistic approach to large-scale network traffic monitoring and analysis”. Computer Networks, 2016: 5-19.

[30] Binfeng Wang,Jun Zhang,Zili Zhang, Lei Pan,Yang Xiang,Dawen Xia. “Noise-resistant Statistical Traffic Classification”, IEEE Transactions on Big Data, 2017.

[31] [31]Zhang J, Chen X, Xiang Y, et al. “Robust network traffic classification”. IEEE ACM Transactions on Networking, 2015, 23(4): 1257-1270.

[32] Li X, Qi F, Xu D, et al. “An internet traffi c classifi cation method based on semi-supervised support vector machine” .2011 IEEE International Conference on Communications (ICC).

[33] Jeffery Erman, Anirban Mahanti, Martin Arlitt, Ira Cohen, Carey Williamson. “Semi-supervised network traffic classification”,. Proceedings of the ACM SIGMETRICS, 2007.

[34] J. Yan, X. Yun, Z. Wu, H. Luo, S. Zhang, S. Jin, and Z. Zhang, “Online traffic classification based on co-training method”, in Parallel and Distributed Computing, Applications and Technologies (PDCAT), 2012 13th International Conference on. IEEE, 2012, pp. 391‑397.

[35] H. He, C. Che, F. Ma, J. Zhang, and X. Luo, “Traffic classification using en-semble learning and co-training”, in WSEAS Proc. of the 8th AIC’ 08, pp. 458‑463.

[36] S. da Silva, C. C. Machado, R. V. Bisol, L. Z. Granville, and A. Schaeffer-Filho, “Identification and selection of flow features for accurate traffic classification in SDN”, in Proc. IEEE 14th Int. Symp. Netw. Comput. Appl., Cambridge, MA, USA, Sep. 28‑30, 2015, pp. 134‑141.

[37] Peng Xiao, Na Liu, Yuanyuan Li, Ying Lu, Xiao-jun Tang, Hai-wen Wang, Ming-xia Li. “A Traffic Classification Method with Spectral Clustering in SDN”, International Conference on Parallel and Distributed Computing, Applications and Technologies(PDCAT).2016. pp. 391~394.

[38] Pedro Amaral,Jo˜ao Dinis,Paulo Pinto, et al. “Machine Learning in Software Defined Networks: Data Collection and Traffic Classification”, .IEEE 24th International Conference on Network Protocols (ICNP) Workshop on Machine Learning in Computer Networks (NetworkML 2016).pp.1-5.

[39] Y. Li and J. Li, “MultiClassifier: A combination of DPI and ML for application-layer classification in SDN”, in 2nd Int. Conf. on Systems and Informatics (ICSAI), 2014, pp. 682‑686.

[40] Zafar Qazi, Jeongkeun Lee,Tao Jin. “Application-Awareness in SDN”,.SIGCOMM’ 13, August 12‑16, 2013, Hong Kong, China.pp 487-488.

[41] Mohammad Reza Parsaei, Mohammad Javad Sobouti, Seyed Raouf khayami, Reza Javidan “Network Traffic Classification using Machine Learning Techniques over Software Defined Networks”, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 8, No.7, 2017 .PP.220~225.

[42] Pu Wang,Shih-Chun Liny, and Min Luo, ”A Framework for QoS-aware Traffic Classification Using Semi-supervised Machine Learning in SDNs”, 2016 IEEE International Conference on Services Computing, PP.760~765.

[43] Ng, B., M. Hayes, and W. K. G. Seah. "Developing a traffic classification platform for enterprise networks with SDN: Experiences &lessons learned." IFIP NETWORKING Conference IEEE, 2015.

[44] Matthew Hayes, Bryan Ng, Adrian Pekar, and Winston K. G. Seah. “Scalable Architecture for SDN Traffic Classification”,. IEEE Systems Journal, 18 April 2017.

[45] Andrea Bianco, Paolo Giaccone, Seyedaidin Kelki, Nicolas Mejia Campos, Stefano Traverso, Tianzhu Zhang. “On-the-fly Traffic Classification and Control with a Stateful SDN approach”,.IEEE ICC 2017 SAC Symposium SDN & NFV Track. May. 2017.Paris. France.

[46] Shota Ogasawara, Yutaka Takahashi. “Performance Analysis of Traffic Classification in an OpenFlow Switch”, 1st International Conference on Internet of Things, CIoT 2016. November. 2016. Thailand..