ijecs
open access
www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 4 Issue 3 March 2015, Page No. 11066-11071

Traffic Classification By Using: TIE (Traffic Identification Engine)

P.Raj kumar, P.Prasanna
MCA Final Year
Vel tech Technical University
Avadi,Chennai-62
E-Mail-rajkumarkutty26@gmail.com
Asst.Prof MCA dept
Veltech Technical University
Avadi,Chennai-62
E.Mail:prasannavelit@gmail.com

Abstract

The availability of open source traffic classification[1][2][3][4][5] systems designed for both experimental and operational use can facilitate cooperation, union on standard definitions and events and trusted evaluation of methods. In this paper, we describe Traffic Identification Engine (TIE), an open source tool for network transfer classification, Investigating the optimal combination strategy and set of classifiers to generate reliable ground truth while preserving privacy Extending the support for sharing labeled traffic with anonym zed traces Investigating strategies for multi-threaded classification, exploiting: off-load techniques focuses by recent traffic capturing engines such as multi queue adapters. Comparing the accuracy of different classifiers and classification performance. Investigating multi-classification and combination strategies.

Keywords: patterns, accuracy of classifier, garbage collector, plug-in

Keywords:

Privacy Preserving Data Mining; Distributed Computation; Frequent

I. Introduction

The evolution of Internet applications has made traditional methods for classifying network traffic progressively less effective. Port based perspective can easily misclassify traffic flows, mostly because of new applications reuse port numbers registered at IANA with other applications, at random selecting port numbers, or let users choose a in port. Payload-based approaches which inspect packets content to identify peculiar patterns are considered more reliable, but pose privacy, technical, and economic challenge, and cannot be applied to encrypted and obfuscated traffic. The increasing use of protocol encapsulation and multi channel applications has further hindered the ability to classify Internet traffic. One of the main issues when novel classification perspective is presented is the inability to properly evaluate and compare them. While the main obstacle to performing such tasks is the actual lack of available implementations, new difficulties derive from intrinsic differences in the types of objects to be classified (flows, TCP connections, etc.), in the considered traffic classes (specific applications, function categories, etc.), as well as in the metrics used to price classification accuracy. In the existing scenario we see the Inability to properly evaluate and compare them. There is a for seen lack of available implementations. The system cannot be applied to encrypted and obfuscated traffic

II. Problem Statement

The evolution of Internet applications has made traditional methods for classifying network traffic progressively less effective. Port based perspective can easily misclassify traffic flows, mostly because of new applications reusing port numbers registered at IANA with other applications, randomly selecting port numbers, or letting users choose a preferred port. Payload-based approaches which inspect packets content to identify peculiar patterns are considered more reliable, but pose privacy, technological, and economic challenges, and cannot be applied to encrypted and obfuscated traffic.

The increasing use of protocol encapsulation and multi channel applications has further hindered the ability to classify Internet traffic. One of the main issues when novel classification perspective is presented is the inability to properly evaluate and compare them. While the main obstacle to performing such tasks is the actual lack of available implementations, other difficulty derive from intrinsic differences in the types of objects to be classified (flows, TCP connections, etc.), in the considered traffic classes (specific applications, function categories, etc.), as well as in the metrics used to estimate classification accuracy

The disadvantages are as following.

- Inability to properly evaluate and compare them
- lack of available implementations
- cannot be applied to encrypted and obfuscated traffic

objective: The objective of the proposed project that I am going to develop will get involved in comparing the accuracy of different classifiers. It will also compare their classification performance. The aim of the result which I will be acquiring is to produce better results in investigating multi-classification and combination strategies

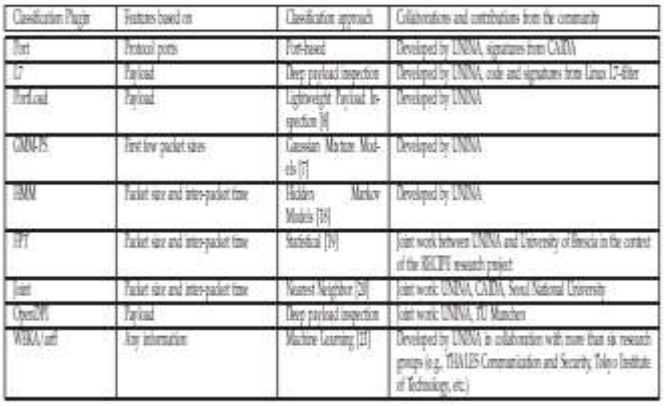
III. Proposed System

To compare different classification a perspective, TIE recommends a unified representation of classification results. It defines IDs for application classes and associates them with group classes, which include applications offering related services. Such mapping enables the comparison of technique working at different granularities or, for instance, the comparison of traffic classifiers which have application-level protocol classes using a coarser granularity. Moreover, several function sub classes are associated with both applications, in order to discriminate linked traffic flows serving different purposes. These briefly describe TIE's components and functionalities by detailing some of the devise choices, focused on multi-classification, relationship of perspective, and online traffic classification.

The advantages are as following.

- Involved in Comparing the accuracy of different classifiers
- Involved in Comparing their classification performance
- Produced better results in Investigating multi-classification and combination strategies

IV. Algorithm as Proposal



Classification Plugin	Features based on	Classification approach	Collaborations and contributions from the community
Port	Protocol ports	Port-based	Developed by UNINA, signatures from CAIDA
L7	Payload	Deep payload inspection	Developed by UNINA, code and signatures from Linux L7-filter
PortLoad	Payload	Lightweight Payload Inspection [6]	Developed by UNINA
GMM-PS	First few packet sizes	Gaussian Mixture Models [7]	Developed by UNINA
HMM	Packet size and inter-packet time	Hidden Markov Models [18]	Developed by UNINA
IPT	Packet size and inter-packet time	Statistical [19]	Joint work between UNINA and University of Brescia in the context of the RECIPE research project
Joint	Packet size and inter-packet time	Nearest Neighbor [20]	Joint work: UNINA, CAIDA, Seoul National University
OpenDPI	Payload	Deep payload inspection	Joint work: UNINA, TU Munchen
WEKA/arff	Any information	Machine Learning [21]	Developed by UNINA in collaboration with more than six research groups (e.g., THALES Communication and Security, Tokyo Institute of Technology, etc.)

Algorithms, as reported in Table 1.b. whenever a new packet associated with an unclassified session is processed by the mark extractor, if all the classifiers are ready to be invoking on that sitting, the DC combines their results according to the configured algorithm[6] in order to take the final conclusion. A confidence value between 0 and 100 represent the overall reliability of such decision. Since most combination algorithms require additional information (a sort of training of the combiner), a set of utilities extracts from a reference file (i.e., a previously generated TIE output file) the confusion matrix and the BKS table necessary to train them. By default the PRI combination is used, where the classifier with higher priority determines the final result.

V. System Architecture

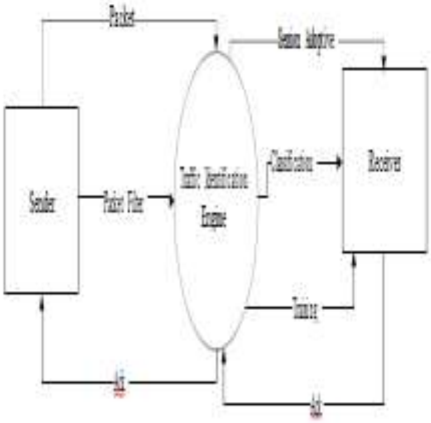


Fig2: System Architecture

We have designed a system architecture to enhance and meet the objective of our project. In my system the packet filter This stage captures link-layer frames or reads them from a file and filters them according to configurable rules. It is based on the well-known LINPAC library, and its filtering capabilities are implemented using both Berkeley Packet Filters and additional user-space filtering rules. The work of the session builder is to organizes network traffic into sessions we defined a generic concept of session to support the various types of traffic flow objects adopted in literature. Except for the first session type, this stage differentiates traffic flowing in two opposite directions by taking as reference the first observed packet. Counters, features, and state information are kept separately for each way. Although flows can be considered a computationally efficient approximation of TCP connections (they only require a lookup on a hash table for each packet), some applications may need more accurate identification of their lifetime. Hence, TIE implements computationally-light heuristics based on TCP streamer that, applied to flows, yield to a better estimate of TCP connections, avoid the segmentation of TCP relations into several flows in presence of long periods of silence. This stage keeps track of sessions using a chained hash table, and to properly work with high traffic volumes it includes a Garbage Collector component responsible for periodically releasing the resources related to classified and expired sessions. The work of the feature extractor is to be responsible for collecting the features required by the classification and is triggered by the session builder for every incoming packet. For each session it provides Basic features (always available to

This stage keeps track of sessions using a chained hash table, and to properly work with high traffic[7][8]volumes it includes a Garbage Collector component responsible for periodically releasing the resources related to classified and expired sessions. The work of the feature extractor is to be responsible for collecting the features required by the classification, and is triggered by the session builder for every incoming packet. For each session it provides Basic features (always available to

classifiers) and Advanced features (extracted on demand). In order to optimize computational efficiency, advanced features are collected only if specified by a command line option and if a skip session flag is not set. While we included support for features based on the most common classification techniques, TIE can easily be extended to extract new features based on definitions already published in the literature or to support original techniques. In order to rapidly experiment with techniques implemented by outdoor tools, this stage can optionally dump for each session the corresponding classification features along with the label assigned by a classifier. TIE supports dumping features directly in some common formats, such as the reformat use by WEKA, 10 one of the most used tools in the field of machine-learning classification. Traffic classification techniques are implemented in TIE as plug-in exposing a standard interface through which their functionalities can be activate. Each plug-in is enabled only if the skin texture it requires are existing and, once enabled, its classification knowledge base is loaded. We currently distribute TIE along with a skeleton plug-in and two basic classification plugins respectively implement traditional approaches: port- and payload-based. Since 2009, several additional plug-in have been developed, also through collaborations with other explore groups, implementing techniques based on machine-learning and statistical approaches.

VI. Experimental Results

When TIE is used to teach classifiers, the fourth stage of the TIE engine pre-loads the labels associated with each session from a view truth file, which can be obtained as production by running TIE on the same traffic communication trace with a ground truth classifier

When TIE is used to prepare classifiers, the last stage of the TIE engine is in charge for invoking the signature-collection functions implemented by each enabled plug-in to let them collect the necessary per-packet information – and to trigger their training at the end of the TIE execution

Classifier	Accuracy of application		Accuracy of different classifiers and classification	
	Session	Byte	Session	Byte
Port load	74.24%	97.83%	73.85%	97.68%
Port	19.57%	25.12%	18.75%	24.21%

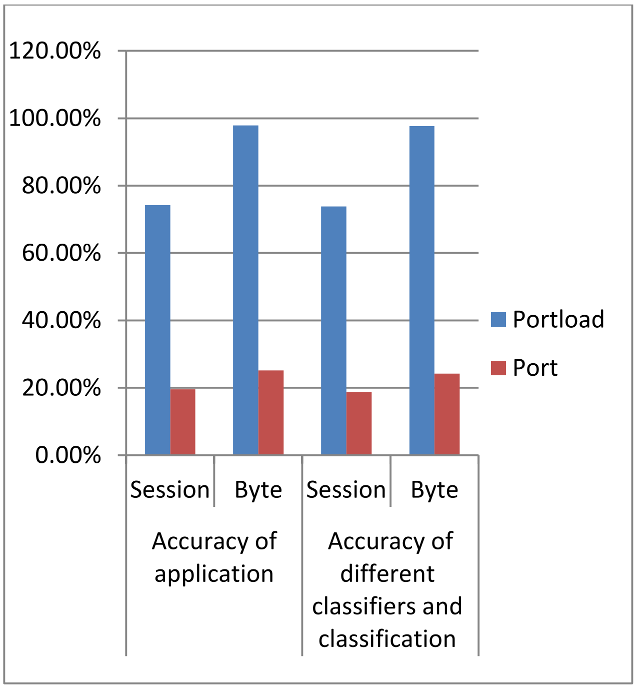


Fig2: Overall accuracy of Port load and Port

VII. CONCLUSION:

This developing in 2008 to help researchers to tackle unsolved challenges in traffic classification. Thanks to the support of the open source community and scientific collaborations, the platform has gradually evolved during the past five years, enabling the production of significant scientific results. In the first quarter of 2014, we plan to release a new version of the platform based on feedback and contributions from users collected in the past two years. Thereafter, we plan to further extend TIE by: Investigating the optimal combination strategy and set of classifiers to generate reliable ground truth while preserving privacy Extending the support for sharing labeled traffic with anonym zed traces Investigating strategies for multi-threaded classification, exploiting: Offloading techniques offered by recent traffic capturing engines such as multi queue adapters and multi-line buses between NICs and CPU cores GPU extensions NUMA capabilities, and so on.

VIII. References

[1]. Karagiannis, T., Papagiannaki, K., Faloutsos, and M.: BLINC: multilevel traffic classification in the dark. In: Proc. of ACM SIGCOMM (2005)

[2] Kim, H., et al.: Internet traffic classification demystified: myths, caveats, and the best practices. In: Proc. of ACM CoNEXT (2008)

[3]. Moore, A., Zuev, D.: Internet traffic classification using bayesian analysis techniques. In: Proc. of ACM SIGMETRICS (2005)

[4]. Nguyen, T., Armitage, G.: A survey of techniques for internet traffic classification using machine learning. IEEE Communications Surveys and Tutorials 10(4) (2008)

[5]. Roughan, M., et al.: Class-of-service mapping for qos: a statistical signature-based approach to ip traffic classification. In: Proc. of ACM SIGCOMM IMC (2004)

[6]. Williams, N., Zander, S., Armitage, G.: Evaluating machine learning algorithms for automated network application identification. CAIA Tech. Rep. (2006)

[7]. Zander, S., Nguyen, T., Armitage, G.: Automated traffic classification and application identification using machine learning. In: Proc. of IEEE LCN Conf. (2005)

[8]. G. Aceto, A. Dainotti, W. de Donato, and A. Pescap, “PortLoad: Taking the Best of Two Worlds in Traffic Classification,” in IEEE INFOCOM 2010 – WIP Track, 2010.