Demo Abstract: NeTraMark: A Network Traffic classification benchMark

Suchul Lee*, Sungryoul Lee*, Hyunchul Kim*, Chong-Kwon Kim*, Dhiman Barman†
Ted "Taekyoung" Kwon* and Yanghee Choi*
Seoul National University*, Juniper†

1. Introduction: Recent research on Internet traffic classification has produced a flurry of proposed approaches for distinguishing types of traffic. However, rigorous comparison of the proposed algorithms remains an almost impossible task, since nearly every paper considers a different benchmark for its experimental evaluation. To shed light on scientifically grounded traffic classification research and respond to the growing necessity for an objective way of comparing results from different research groups [4], [5], this demonstration introduces an extensible Internet traffic classification benchmark, NeTraMark. To the best of our knowledge, this is the first benchmark where all the state-of-the-art Internet traffic classification approaches are integrated; the payload-based classifier called crl_pay [4], [7], graphs-based classifiers like BLINC [7] and Traffic Dispersion Graphs (TDG) [6], the seven most oft-used machine learning algorithms[1] [4], and CoralReef's [1] ports-applications matching database.

2. Design Principles: We employ six design principles for developing NeTraMark. Those principles are: ***Comparable***: Experimental results of different approaches should be comparable based on the same performance metrics. ***Reproducible***: Experimental results should be reproducible for verification. ***Extensible***: A new classification method should be easily added and integrated into existing software or it should be easy to modify and enhance an existing method. ***Synergetic***: It is known that each traffic classification method has its own strengths and weaknesses (e.g., payload-based classifiers are not applicable to encrypted traffic data), and careful combinations can provide synergy [4]. Our benchmark allows users to test and obtain synergy by combining/applying multiple classification methods on a given dataset, in order to build a combined classifier which outperforms individual ones. ***Flexible in use***: to enable users to setup plugged-in classifiers in variable configurations determined at runtime, either interactively or via batch. (e.g., configuring which classifier will be used to establish a comparison reference point, which features will be selected and/or discretized in advance, which classifiers will be combined together to create a new one, ...) ***Easy-to-use*** : All functionalities derived from the above design principles should be easy-to-use, both for traffic classification researchers as well as network administrators.

[1]So far we have integrated the most oft-used seven machine learning algorithms by plugging in WEKA [2]; C4.5 Decision Tree, Naive Bayes, Naive Bayes Kernel Estimation, Bayesian Networks, k-Nearest Neighbors, Neural Networks and Support Vector Machines

3. Architecture: Fig. 1 shows the NeTraMark architecture, which consists of five parts: (1) Graphical and Command-Line User Interface for interactive and batch processing, (2) Data Pre/Post-Processing like traffic flow feature selection, discretization, and input data format conversion, etc., (3) Traffic Data Repository (currently off-line flow data only), (4) the Classification part where we proudly integrate most of the state-of-the-art traffic classification approaches for the first time, and (5) PostgreSQL-based [3] Database where (reusable) classification and/or benchmark results are stored and retrieved for faster (repetitive-)processing of voluminous traffic flow data.

4. Performance Metrics: To evaluate and compare the performance of different traffic classifiers, NeTraMark uses four metrics: *overall accuracy, precision, recall, F-measure*.

•*Overall accuracy* is the ratio of the sum of all True Positives to the sum of all the True Positives and False Positive for all classes.[2] We apply this metric to measure the accuracy of a classifier on the whole trace set. The latter three metrics are to evaluate the quality of classification results for each application class.

•*Precision* is the ratio of True Positives over the sum of True Positives and False Positives or the percentage of flows that are properly attributed to a given application category.

•*Recall* is the ratio of True Positives over the sum of True Positives and False Negatives or the percentage of flows in application category that are correctly identified.

•*F-Measure*, a widely-used metric in information retrieval and classification, considers both precision and recall in a single metric by taking their harmonic mean: $2 \times precision \times recall / (precision + recall)$.

5. Demonstration: Our demo will illustrate the NeTraMark capabilities by performing the following tasks: As shown in Fig. 2, we demonstrate the ease of evaluating and comparing the performance of the 10 built-in classifiers using our benchmark software, as well as adding users' own new classifiers to the benchmark, various interactive configuration and pre/post-processes like traffic flow feature selection, discretization, and training/testing classification algorithms with multiple (sampled) datasets collected from different locations. We also show that network administators obtain synergetic benefits by

[2]*True Positive* is the number of correctly classified flows, *False Positive* is the number of flows falsely ascribed to a given application, and *False Negative* is the number of flows from a given application that are falsely labeled as another application.

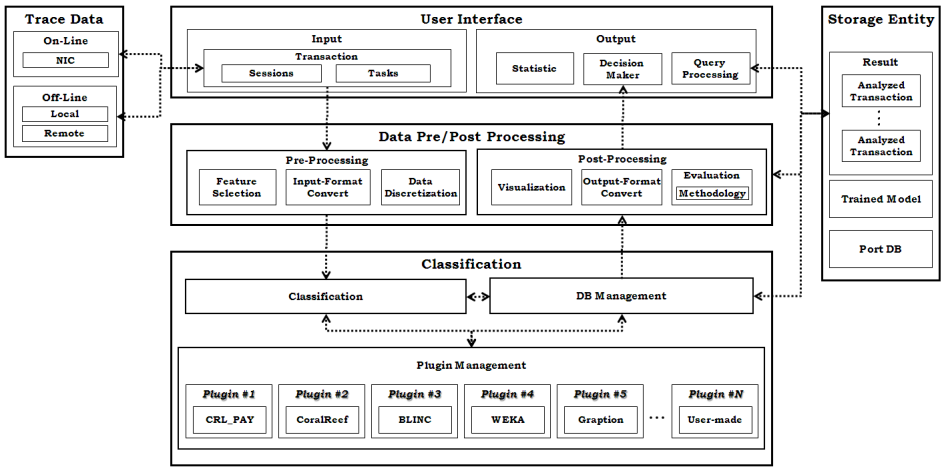

Fig. 1. The NeTraMark Architecture

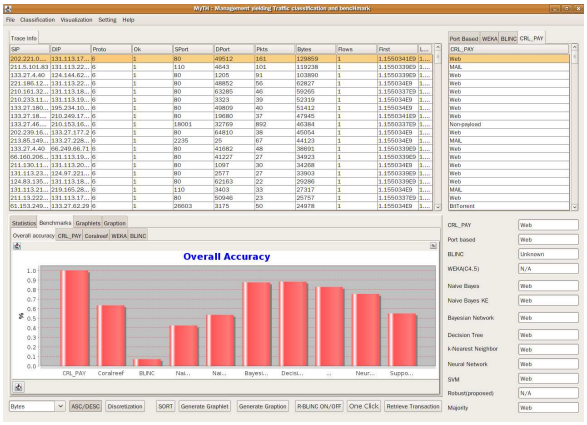

Fig. 2. Screenshot : GUI and Benchmark results of 9 traffic classifiers

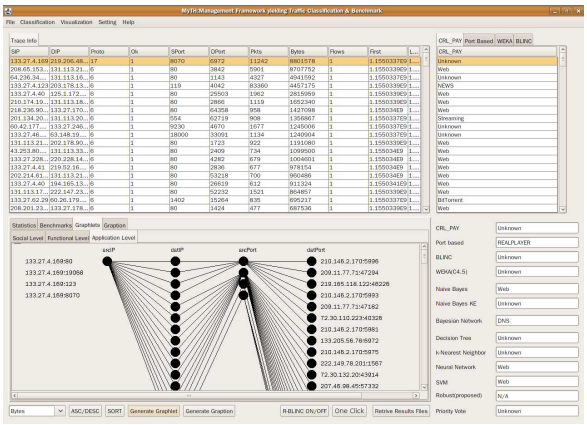

Fig. 3. screenshot: The use of BLINC host-graphlets to identify heavy hitters

using multiple methods in classifying traffic; e.g., using graph-based methods like BLINC or TDG to identify heavy-hitters (either in terms of the number of flows or transferred bytes), malicious attacks, and new P2P application flows previously classified as "unknown' by other methods.

Acknowledgment

We thank Thomas Karagiannis, Marios Iliofotou, and Michalis Faloutsos for sharing their BLINC and TDG codes with us. This work was supported by NAP of Korea Research Council of Fundamental Science and Technology and the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency) (NIPA-2010-(C1090-1011-0004)).

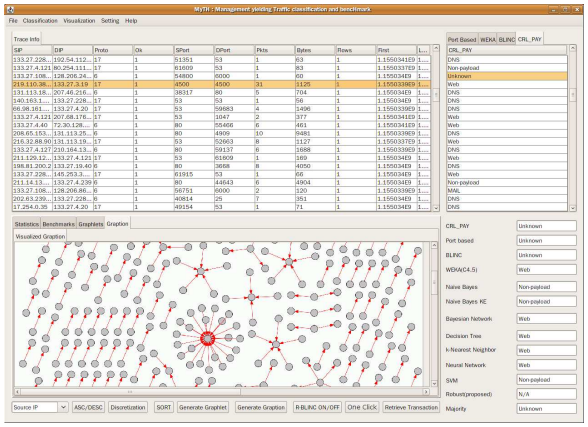

Fig. 4. screenshot: The use of Network-wide communication behavior of hosts (TDG; Traffic Dispersion Graphs) to address Unknown flows

REFERENCES

[1] *CoralReef.* http://www.caida.org/tools/measurement/coralreef/
[2] *WEKA: Data Mining Software in Java.* http://www.cs.waikato.ac.nz/ml/weka/
[3] *Postgresql* http://www.postgresql.org
[4] H. Kim, K. claffy, M.Fomenkov, D.Barman, M. Faloutsos and K. Lee, Internet Traffic Classification Demystified:Myths, Caveats and the Best Practices. In ACM CoNEXT 2008.
[5] L. Salgarelli, F. Gringoli and T. Karagiannis, Comparing Traffic Classifiers. In ACM SIGCOMM CCR Vol 37, Num 3, July 2007.
[6] M. Iliofotou, P. Pappu, M. Faloutsos, M. Mitzenmacher, S. Singh, and G. Varghese. Network monitoring using traffic dispersion graphs. In ACM IMC, October 2007.
[7] T. Karagiannis, K. Papagiannaki, and M. Faloutsos. Blinc: Multilevel traffic classification in the dark. In ACM SIGCOMM, August 2005.