Byte Me: A Case for Byte Accuracy in Traffic Classification

Jeffrey Erman[¶], Anirban Mahanti[§], Martin Arlitt[‡¶]
[¶]Department of Computer Science, University of Calgary, Canada
[§]Department of Computer Science and Engineering, Indian Institute of Technology, Delhi, India
[‡]Enterprise Systems & Software Lab, HP Labs, Palo Alto, USA

ABSTRACT

Numerous network traffic classification approaches have recently been proposed. In general, these approaches have focused on correctly identifying a high percentage of total flows. However, on the Internet a small number of "elephant" flows contribute a significant amount of the traffic volume. In addition, some application types like Peer-to-Peer (P2P) and FTP contribute more elephant flows than other applications types like Chat. In this opinion piece, we discuss how evaluating a classifier on flow accuracy alone can bias the classification results. By not giving special attention to these traffic classes and their elephant flows in the evaluation of traffic classification approaches we might obtain significantly different performance when these approaches are deployed in operational networks for typical traffic classification tasks such as traffic shaping. We argue that byte accuracy must also be used when evaluating the accuracy of traffic classification algorithms.

Categories and Subject Descriptors

C.2.2 [**Computer-Communications Networks**]: Network Protocols

General Terms

Algorithm, Measurement, Performance

Keywords

Traffic Classification, Machine Learning

1. INTRODUCTION

The number of applications in use on the Internet is continuously increasing. To effectively monitor and manage their networks, network administrators need to accurately identify the type of applications and the impact of their corresponding traffic. This enables network administrators to create policies to restrict and reduce the amount of undesirable traffic and ensure business critical traffic is prioritized.

Since the early 2000s, traffic classification has become a difficult task. In particular some application types such as Peer-to-Peer (P2P) have been built with features specifically intended to avoid common traffic classification techniques. This includes using dynamic port numbers and payload encryption. At the University of Calgary, where strict payload-based traffic shaping is deployed, we have observed that P2P traffic still accounts for almost 40% of the bytes transferred on our network, even though identified P2P is relegated to a lower priority class.

Historically, port [7] and payload [6, 14, 17] based approaches have been used for traffic classification in the literature and by commercial vendors. However, these approaches are affected by several drawbacks such as being either increasingly ineffective or incurring high overhead hampering their deployment. While these drawbacks have received some attention [6, 10, 12], they have spurred new traffic classification techniques to be developed based on using the behaviours of hosts [8, 9, 20] and using machine learning [1–4, 15, 16, 19].

Although several classification techniques have been proposed, they have all been evaluated using different traces and metrics, which makes it difficult to effectively compare one technique with another. One of the main reasons is that there has been much focus on achieving only a high flow accuracy. However, on the Internet some large "elephant" flows have a much greater effect on the network than small "mice" flows. Elephant flows have been known to account for over 90% of the bytes transferred on typical networks [11]. In addition, some classes of traffic such as Web and P2P traffic are more likely to introduce these elephant flows than other types such as Internet Chat. These are important aspects that must be taken into consideration as they can considerably affect the performance of a classifier. In the machine learning literature, this is referred to as a *class imbalance problem* [18].

In this opinion piece, we argue that byte accuracy is an important measure for a classifier's performance. Byte accuracy better accounts for the class imbalances amongst applications and is a better indicator of the performance the classifier would obtain for typical traffic classification tasks such as traffic shaping, and traffic analysis. In most realtime traffic usages, byte accuracy is very important to the classifier. Ideally, we want to identify the majority of the flows as well as the bytes.

The rest of this opinion piece is structured as follows. Section 2 provides background information and pointers to related work. Section 3 describes the impact the class imbalance problem has in traffic classification. Section 4 discusses typical traffic classification uses. Section 5 gives our conclusions and recommendations.

2. BACKGROUND

Given the shortcomings of port-based and payload-based traffic classification techniques, researchers have looked for alternative solutions. A promising approach to traffic classification is the use of machine learning. This approach relies on the premise that a set of features for objects would be similar when objects are of the same class. In general, a feature can be any attribute that is rel-

MineNet'07, June 12, 2007, San Diego, California, USA.

Table 1: Proposed Traffic Classification Approaches

Technique	Flow Accuracy	Byte Accuracy
Class-of-Service [16]	✓	
Naive Bayes [15]	✓	✓
BLINC [9]	✓	✓
Early Application Identification [1]	✓	
Supervised Machine Learning [19]	✓	
Statistical Fingerprinting [2]	✓	
Semi-Supervised Learning [3,4]	✓	✓

evant to the prediction of the target set of classes. In the case of traffic classification, the objects under consideration are flows and the classes are the different applications or traffic types (e.g., P2P, Web, Email) the flow is attempted to be classified as.

In machine learning there are generally two stages when developing a classifier. The first stage "learns" a mapping between the objects and the desired classes. This mapping is done using a labelled training data set. Subsequently, in the second stage this learned mapping is used by the classifier to label new objects.

There have been several proposals using machine learning for traffic classification [1–3, 5, 15, 16, 19]. Some approaches have focused on using features that are based on aggregate flow statistics such as average packet size [3,5,15,16,19]. Other approaches have used per-packet statistics such individual packet sizes and interarrival times [1, 2].

In addition to machine learning approaches, there has been work on leveraging the communication patterns of hosts to classify traffic, for example, BLINC [9]. However, for this piece we focus our discussion on the machine learning approaches. Table 1 summarizes the proposed approaches and the metrics they use to evaluate their classifiers. Many of the recent papers [1, 2, 19] have ignored byte accuracy even though some early works [9, 15] have used it.

3. CLASS IMBALANCE PROBLEM

One of the invariants of the Internet has been characterized as the "elephants and mice phenomenon" [11]. This phenomenon is that the majority of the flows on the Internet are small and only a small portion of the total bytes and packets in the network are the result of these flows. These small flows are referred to as mice. The majority of the traffic is the result of a small number of large flows (e.g., the elephant flows). The elephant traffic can be responsible for over 50% of the traffic in some networks. (See references therein [11] for additional details).

There are several different definitions available for identifying what are elephants and mice flows (See [11]). However, we believe a simple definition suffices for our ensuing discussion. In particular, we utilize a threshold to distinguish between elephant and mice flows; i.e., elephant flows are larger than x KB of data transfer and mice are less than x KB of data transfer.

In our own traffic classification experience, we collected traces from the University of Calgary for a 6-month period of time [3, 4]. When we analyzed these traces we found as expected this elephant and mice phenomenon existed in them as well. Figure 1 shows the fraction of total bytes transferred by the largest flows in our traces. As can be seen, the top 1% of flows account for over 73% of the traffic and the top 5% of flows account for 83% of the traffic in terms of bytes. The value of $x = 288$ KB and $x = 57$ KB would place the top 1% and 5% of flows into elephants, respectively. The top 0.1% of flows account for 46% of the traffic with $x = 3.7$ MB.

This is an important consideration to take into account when designing a network traffic classification approach. Otherwise, a classifier naively optimized to identify all but the top 0.1% of the flows would attain a 99.9% flow accuracy but would still result in 46% of the bytes in this trace being misclassified.

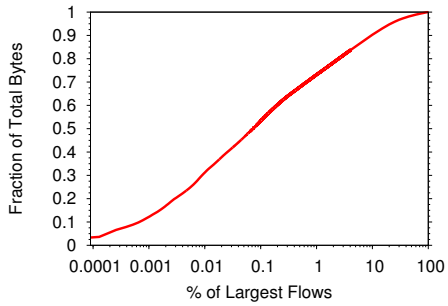

Figure 1: CDF of the Bytes in Flows

In the machine learning literature, this type of data set has what is called a class imbalance problem. This problem is not specific to network traffic classification and is quite common in other real world classification problems. A few examples include the detection of insurance fraud or the diagnoses of a rare medical condition. Due to the common occurrence of class imbalance problems there has been research done in the machine learning literature to overcome some of its challenges.

One of the methods to overcome this elephants and mice problem would be to use the idea of cost sensitivity when measuring the performance of a traffic classifier. Cost sensitivity makes the misclassification of more "expensive" objects more costly. For traffic classification, a simple measure of the cost of a flow to the network is the total number of bytes it transfers. This can be represented aggregately for a trace as *byte accuracy*.

This idea of cost sensitivity can be taken much further with the incorporation of cost-sensitive learning methods into the design of the network classifier such as Bayesian decision theory. This type of classification technique attempts to improve "accuracy" by minimizing the total cost of the misclassification from the classification model it generates. The decision criteria of the more expensive classes are relaxed to be more inclusive. For traffic classification, this means the classifier would try to increase the accuracy on the elephant flows while potentially decreasing its accuracy on the mice flows. This can be accomplished in several ways such as choosing the set of features that best maximizes some desired function of flow and byte accuracy.

Another way to influence the classifier is to use a sampling based approach. This type of approach would train the classifier with more of the rare but expensive cases. One example could be to train the classifier with a training data set that contained an equal number of elephant and mice flows. This can be accomplished using either an undersampling or an oversampling technique. Undersampling would choose fewer of the mice flows and oversampling would replicate additional elephant flows for the training data. In our own work [3, 4], we have found this type of approach to work very well for improving byte accuracy. Initially, we found that achieving high flow accuracy was quite easy and that high byte accuracy was as expected much more difficult. To improve our byte accuracy we trained our classifier with a data set that contained 50% of flows below the 95% percentile of the flow sizes and 50% of flows above the 95% percentile of flow sizes. This allowed our classifier to substantially improve its byte accuracy for the classification models it generated with only a marginal reduction in flow accuracy.

Finally, some traffic types such as Web, P2P, and FTP are more likely to contain these large elephant flows. Thus, it is important to attempt to classify such traffic types when evaluating any traffic

classification approach. In addition, P2P traffic is currently especially important to classify, as it purposely has been designed to be elusive to traditional traffic classification approaches.

As a final example of the effect of rare elephant flows on classifier performance, we draw on our own experience. When we tested the classifier we designed over a 6-month period we found that on two days the byte accuracy dropped significantly (by 25% [4]). When we investigated these pronounced drops we found this was due to the misclassification of FTP flows. In general, for our traces FTP was not captured well by any of the sampling techniques we used because it accounts for only a small fraction ($< 0.01\%$) of the total flows, and thus, is unlikely to be captured in a small-sized training data set. Typically, FTP accounted for less than 5% of the bytes but on those two days a few large FTP transfers accounted for 21.6% and 26.6% of the bytes, respectively.

4. DISCUSSION

The lack of in-depth analysis of byte accuracy and the focus on high flow accuracy largely ignores many of the common uses for traffic classification. The typical uses of traffic classification can be divided into two categories: offline and realtime. In the realtime case, byte accuracy plays a much more important role than flow accuracy as many of the potential uses are for traffic shaping and flow prioritization. In both of these uses the misclassification of one large elephant flow can substantially outweigh the benefit of better flow accuracy.

In addition to byte accuracy, we could also use other metrics to evaluate our classifiers. These could include measuring the precision and recall of elephant flows, and using receiver operating characteristic (ROC) curves to tune our classifier's performance.

One of the foremost challenges of traffic classification currently is effectively comparing between the many proposed approaches. This is especially crucial for the many machine learning approaches that have been put forward.

We believe there are several reasons why comparing classification approaches currently is quite difficult. First, the performance metrics used to evaluate techniques vary widely. As we have discussed, byte accuracy has been ignored in the results of several studies. Second, publicly available data sets with a reliable "base truth" (i.e., not from port-based analysis) are not available. While this can not easily be changed, solutions to this challenge have been proposed at (See [13] and last year's MineNet forum discussion). Third, many of the currently proposed approaches only attempt to classify a small subset of applications and this subset differs between studies. Some applications such as P2P specifically attempt to disguise their traffic from classification and essentially are what have prompted this interest in traffic classification. Thus, it is essential that these approaches are tested against these types of traffic. Otherwise, standard port and payload classification techniques are sufficient. Finally, many of the techniques have different tuning parameters and use different features.

Of these challenges, when evaluating a classification approach what can easily be controlled is the traffic classes (i.e., the applications) the approach is tested upon and the performance metrics used for evaluation of the approach.

5. CONCLUSIONS

In this opinion piece we have discussed challenges faced by traffic classification techniques. Specifically, we have described how a small number of elephant flows contribute to a large portion of the overall network traffic volume. This is important to consider when designing a classifier as it greatly affects the performance of the classifier for common traffic classification tasks such as traffic shaping.

The machine learning literature has already studied similar class imbalance problems. We argue that these should be considered for traffic classification. For example, traffic classification approaches could make use of cost-sensitive learning approaches. The incorporation of byte accuracy into the evaluation of classifiers will also help in comparing proposals and enable network administrators to choose which approach best suits their performance requirements.

This paper has also highlighted some of the open problems in this area. While traffic classification can seem like an unending game of cat and mouse, we believe that many of the challenges can be overcome. We believe that this will help guide future researchers when designing the next generation of traffic classification approaches.

6. ACKNOWLEDGMENTS

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada and Informatics Circle of Research Excellence (iCORE) of the province of Alberta.

We thank Ira Cohen of HP Labs for his comments and suggestions which helped improve this paper.

7. REFERENCES

[1] L. Bernaille, R. Teixeira, and K. Salamatian. Early Application Identification. In *CoNEXT'06*, Lisboa, Portugal, December 2006.

[2] M. Crotti, M. Dusi, F. Gringoli, and L. Salgarelli. Traffic Classification through Simple Statistical Fingerprinting. *Computer Communications Review*, 37(1):7–16, 2007.

[3] J. Erman, A. Mahanti, M. Arlitt, I. Cohen, and C. Williamson. A Semi-Supervised Approach to Network Traffic Classification. In *SIGMETRICS'07 (Extended Abstract)*, San Diego, USA, June 2007.

[4] J. Erman, A. Mahanti, M. Arlitt, I. Cohen, and C. Williamson. Offline/Online Traffic Classification Using Semi-Supervised Learning. Technical report, University of Calgary, 2007.

[5] J. Erman, A. Mahanti, M. Arlitt, and C. Williamson. Identifying and Discriminating Between Web and Peer-to-Peer traffic in the Network Core. In *WWW'07*, Banff, Canada, May 2007.

[6] P. Haffner, S. Sen, O. Spatscheck, and D. Wang. ACAS: Automated Construction of Application Signatures. In *SIGCOMM'05 MineNet Workshop*, Philadelphia, USA, August 2005.

[7] IANA. Internet Assigned Numbers Authority (IANA). http://www.iana.org/assignments/port-numbers.

[8] T. Karagiannis, A. Broido, M. Faloutsos, and k. claffy. Transport Layer Identification of P2P Traffic. In *IMC'04*, Taormina, Italy, October 2004.

[9] T. Karagiannis, K. Papagiannaki, and M. Faloutsos. BLINC: Multilevel Traffic Classification in the Dark. In *SIGCOMM'05*, Philadelphia, USA, August 2005.

[10] S. Kumar, S. Dharmapurikar, F. Yu, P. Crowley, and J. Turner. Algorithms to Accelerate Multiple Regular Expressions Matching for Deep Packet Inspection. In *SIGCOMM '06*, Pisa, Italy, September 2006.

[11] K. Lan and J. Heidemann. A Measurement Study of Correlations of Internet Flow Characteristics. *Computer Networks*, 50(1):46–62, 2006.

[12] J. Ma, K. Levchenko, C. Krebich, S. Savage, and G. Voelker. Unexpected Means of Protocol Inference. In *IMC'06*, Rio de Janeiro, Brasil, October 2006.

[13] J. C. Mogul and M. Arlitt. SC2D: An Alternative to Trace Anonymization. In *SIGCOMM'06 MineNet Workshop*, Pisa, Italy, September 2006.

[14] A. Moore and K. Papagiannaki. Toward the Accurate Identification of Network Applications. In *PAM'05*, Boston, USA, March 2005.

[15] A. Moore and D. Zuev. Internet Traffic Classification Using Bayesian Analysis Techniques. In *SIGMETRIC'05*, Banff, Canada, June 2005.

[16] M. Roughan, S. Sen, O. Spatscheck, and N. Duffield. Class-of-Service Mapping for QoS: A Statistical Signature-based Approach to IP Traffic Classification. In *IMC'04*, Taormina, Italy, October 2004.

[17] S. Sen, O. Spatscheck, and D. Wang. Accurate, Scalable In-Network Identification of P2P Traffic Using Application Signatures. In *WWW'04*, New York, USA, May 2004.

[18] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[19] N. Williams, S. Zander, and G. Armitrage. A Preliminary Performance Comparison of Five Machine Learning Algorithms for Practical IP Traffic Flow Classification. *Computer Communication Review*, 30:5–16, October 2006.

[20] K. Xu, Z. Zhang, and S. Bhattacharyya. Profiling Internet Backbone Traffic: Behavior Models and Applications. In *SIGCOMM '05*, Philadelphia, USA, August 2005.