Revealing the Unknown ADSL Traffic Using Statistical Methods

Marcin Pietrzyk[1], Guillaume Urvoy-Keller[2], and Jean-Laurent Costeux[1]

[1] Orange Labs, France
{marcin.pietrzyk,jeanlaurent.costeux}@orange-ftgroup.com
[2] Institute Eurecom, France
{urvoy}@eurecom.fr

Abstract. Traffic classification is one of the most significant issues for ISPs and network administrators. Recent research on the subject resulted in a large variety of algorithms and methods applicable to the problem. In this work, we focus on several issues that have not received enough attention so far. First, the establishment of an accurate reference point. We use an ISP internal Deep Packet Inspection (DPI) tool and confront its results with state of the art, freely available classification tools, finding significant differences. We relate those differences to the weakness of some signatures and to the heuristics and design choices made by DPI tools. Second, we highlight methodological issues behind the choices of the traffic classes and the way of analyzing the results of a statistical classifier. Last, we focus on the often overlooked problem of mining the unknown traffic, i.e., traffic not classified by the DPI tool used to establish the reference point. We present a method, relying on the level of confidence of the statistical classification, to reveal the unknown traffic. We further discuss the result of the classifier using a variety of heuristics.

1 Introduction

Knowledge about the applications that generated a traffic mixture in the network is essential for ISPs and network administrators. It can be used as the input for a number of network planning, charging and performance studies. The objective of traffic classification is to automatically and accurately find out what classes or precise applications are run by the end users. This task, recently becomes more and more challenging. The reason for this lies in the TCP/IP protocol stack design, which is not providing explicit information about the application that generated traffic. Performance of classically used methods, e.g., port based classification, is diminishing due to the development of new applications, which purposely try to evade traffic detection. We observe this behavior on our platform, where half of the traffic on the ftp legacy ports is generated by peer-to-peer applications. The research community reacted with a number of works proposing solutions or possible lines of further inquiry to solve this important problem [1], [3], [5], [7].

In this paper, we put the emphasis on a number of key issues that need to be addressed while performing statistical classification of traffic. A first key issue is the calibration of the statistical classifier. This task is in general addressed using some DPI tool. Considering an hour long TCP trace from a large ADSL platform, we highlight in Section

M. Papadopouli, P. Owezarski, and A. Pras (Eds.): TMA 2009, LNCS 5537, pp. 75–83, 2009.

3 the possible weaknesses of DPI tools and the difficulty of reconciling the results from different tools. Another issue when using any statistical classifier is the choice of the traffic classes, i.e., the desired level of granularity that one wants to achieve, as well as the interpretation of the results of the classifiers. We illustrate those points in Section 5. Last but not least, we focus on the problem of mining the unknown traffic, i.e., the traffic that the DPI tool failed to identify. This is indeed the ultimate goal of a classifier to be applied in the wild and it is important to see what it can extract when DPI techniques fail. We discuss in Section 6 how a classifier can be used for this task and propose several heuristics to confirm the results.

2 Areas of Improvement

In this section, we describe in more details several issues that are often overlooked while experimenting with statistical classification tools. We further illustrate them in Sections 3 to 6 using a trace captured on a large ADSL platform.

Reference point. As already mentioned, in order to assess the accuracy of any method used, a good reference point[1] is required. This point often does not get enough attention. There are many reasons behind this fact. First, it is in general difficult to obtain relevant traces containing application payload. Second, the design of application signatures is a complex task. Let us consider the case of the eMule application. A signature commonly used [3,7] test for the presence of the /(xe3 or xc5)/ bytes in the payload. It has two drawbacks. First, it can lead to a high fraction of false positives. Second, since 2006, eMule is supporting protocol obfuscation, which makes this simple signature missing an important fraction of eMule flows. DPI tools not only rely on signatures but also feature some heuristics to flag application traffic. As an example, authors in [3] deal with the eMule encryption issue by assuming that all unclassified flows of the end users who have at least one flow classified as eMule are due to eMule. This approach might sometimes be misleading, for instance in the simple case where the user runs eMule in parallel with another encrypted service. We further investigate the problem of reference point establishment by comparing 3 DPI tools in Section 3.

Traffic classes definition. Traffic classes can be defined in different ways. Some papers provide a very coarse grained division; others focus on the detection of a single protocol. It makes comparisons difficult. We approach this problem by performing a two level study. First, we apply our methods against a coarse-grained classes definition. Second, we divide the peer-to-peer class into four subclasses, each containing a single application. We discuss the accuracy of the method in the two cases in Section 5.

Unknown class. Most of the studies follow the same schema. They calibrate one or several statistical methods using a fraction of pre-labeled flows and test its accuracy over larger, classified sets. However, no matter how good the DPI tool is, there remains a fraction of traffic not classified. This traffic class, which accounts for as large as 60% of the flows in some cases (e.g., [7]) is put aside. In the best case, the authors obtain

[1] What we term reference point in this work is often called 'ground truth' in the literature.

classifiers that are as good as the refference point provided by the DPI tool. In practice, the classification of the unknown traffic is a key issue. We address this problem in two ways in Section 6. First, we use our (best) classifier over the unknown class of traffic and report on its predictions assuming specific confidence levels. Next, we investigate several heuristics, based on endpoints profiling to back up the statistical tool output.

3 Reference Point Issue

We used three DPI tools and compare their results on an example ADSL trace which high level description is provided in in Section 4:

- A tool based on Bro [9] that implements the set of signatures used by Erman in [3], as extracted from the technical report of the same author;
- An internal tool called Claude that is constantly developed and tested at Orange;
- Tstat v2 [11] that features DPI functions.

The results with Bro turned out to be deceiving, with more than 55% of unknown flows. A more detailed inspection of over thirty of the signatures used, revealed that most of them are outdated.

We thus decided to focus our comparison on Claude and Tstat only. Claude is used for network analysis and dimensioning. It is capable of detecting several terms of applications, including encrypted ones. It combines several methods of traffic classification, from deep packet inspection to methods as sophisticated as parsing the signaling messages of an application in order to extract endpoint IPs and ports. Claude is constantly developed and tested on several sites in France.

To compare Tstat to Claude we need to devise a set of application classes that both tools detect. We consider the following set: Web, eDonkey, BitTorrent, Ares and Mail. We will use more classes in Section 4.

Results of the comparison between Tstat and Claude are depicted in Table 1. We report the breakdown of flows obtained using each tool and also the overlap between the two tools taking the union of both sets as a reference for each class. For p2p applications, the agreement is very good, in each case higher than 90%. For Mail and Web, we have more significant differences. A closer look at Web traffic revealed that the difference between the two tools is mostly due to Claude identifying more Web transfers than Tstat. This additional set of flows consists of a large fraction of connections to port 443 - https service - or flows with less than three packets. This most probably explains why Tstat did not classify them as Web. As for the Web flows identified by Tstat only, they appeared to be mostly due to streaming applications over http, e.g., YouTube video streaming. Tstat labels those flows as Web while Claude labels them as Http Streaming. While there is a limited number of such flows, they carry a significant amount of bytes, which leads to a more pronounced disagreement between Tstat and Claude when focusing on bytes rather than flows. More generally, looking at bytes provides a different picture. For instance, for the case of eDonkey, Tstat and Claude agree for only 50% of the bytes. This is because Tstat does not recognized obfuscated eDonkey traffic.

Table 1. Tstat vs. Claude comparison

Tstat 2.0 vs Claude [%]			
Class	Tstat Breakdown	Claude Breakdown	Overlap
UNKNOWN	32,67	12	27,92
WEB	58,35	77	81,31
EDONKEY	3,81	4,85	90,72
BITTORENT	0,91	1,06	99,52
ARES	0,09	0,06	99,53
MAIL	3,65	5,06	83,41

We fed Tstat with hand-made obfuscated eDonkey traces to confirm that it does not detect encrypted traffic.

The main lesson we learn from the above study is that even two state-of-the-art DPI tools can lead to sometimes significantly different reference points. We leave the detailed investigation of the root cause of those differences as further work. In the remaining of this paper, we rely on Claude only, due to the lowest fraction of the Unknown traffic it offers and the largest variety of the applications that the tool can handle, as compared to Tstat.

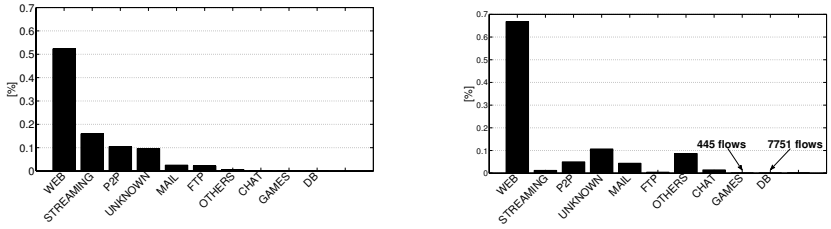
4 Trace

For our tests we use an ADSL trace captured in 2008. The trace was collected using passive probes located just behind the so-called Broadband Access Server. The capture was performed without any sampling or loss. The trace contains one hour full bidirectional traffic of 3237 end users of the ADSL platform. The whole payload is available along with packet headers. A short description of the trace is provided in Table 2. In this work we consider TCP traffic only as it is the dominating transport layer. All the IPs except for the Unknown class flows were fully anonymized. Traffic was classified with Claude.

We consider two levels of traffic class division. First, a less detailed division containing: Web, Streaming, P2P, Mail, Ftp, Others, Chat, Games, Database. Second, a richer set where the p2p class is further divided into several popular applications, namely eMule , Bittorrent , Gnutella , Ares, TribalWeb and other applications. In the second set, we also divide the Streaming class into p2p Streaming and http Streaming. Not all of the applications that our classification tool is capable of detecting are present in the set. Breakdown of flows and bytes for the trace is given in Figures 1(a), 1(b). Figure 2(d) depicts a more detailed breakdown of peer-to-peer applications. Our data set contains a large fraction of Web traffic, which accounts for more than half of the bytes and over 70% of flows. Concerning the bytes transferred, the second class is Streaming which transfers even more data than the p2p class. As for the p2p class, most bytes and flows are generated due to eDonkey followed by Bittorent. Among the less popular applications, we observed only Gnutella and Ares. The fraction of Unknown flows, which our refference point tool was not able to classify, is 11%.

Table 2. Trace summary

Data set	Date	Start time	Duration [h]	Size [GB]	Packets	TCP Flows
MS-I	2008-02-04	14:45	1	26	47‘616‘695	626‘727



(a) Application breakdown. Bytes. (b) Application breakdown. Flows.

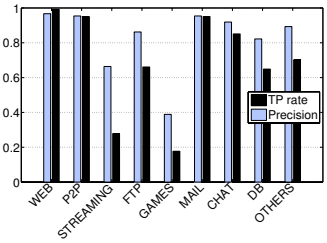
Fig. 1.

5 Machine Learning Classification

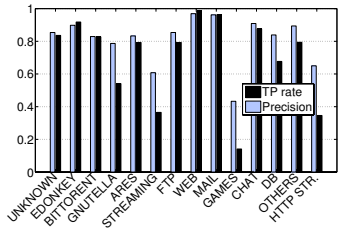
In this section, we report on our experience with machine learning techniques and different sets of application classes. We used the same flow feature to perform the classification for both levels of precision. Those were extracted using an internal tool providing over eighty per flow features and ad-hoc scripts to obtain other discriminators. We tested several supervised algorithms (Naive Bayes with Kernel estimation, Bayesian Network, Support Vector Machine and C4.5) and features sets in order to find the best performing one. For all the algorithms, we used the WEKA suite [10] that is a machine learning toolkit implementing a variety of state of the art methods for data mining. We selected the following per flow features: inter packet time (up and down), mean packet size (up and down), number of pushed packets (up/down), number of data packets. Using this set, we tested several supervised classification algorithms in order to find the one offering the best overall accuracy for our case. The best performance in terms of accuracy and speed were provided by the C4.5 decision tree algorithm. Those results are in line with the ones in [7], where C4.5 appeared also to be the fastest algorithm with a good (but not the best) accuracy.

We applied the widely used method called N-fold cross validation to train the tool [8]. The dataset is randomly split into N parts. Each part is used for training, while the rest is used for testing the accuracy. The process is repeated N times and the resulting performance measures are averaged across all the experiments. We used N=10, as it was claimed in [8] that this number provides a good approximation of operational performance.

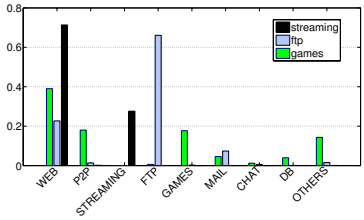
The overall accuracy, defined as the fraction of correctly classified flows over all class is 96.62% for the general application breakdown using C.4.5. However, the overall accuracy can be misleading, as classes of traffic are not equally represented in the set. In Figure 2(a) we depict per class accuracy and precision. For most classes we get reasonable accuracy varying between 65% for the DB class to 98% for Web. Streaming



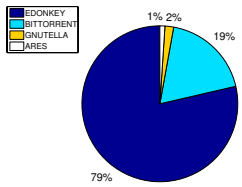
(a) Per application TP rate and precision



(b) Per application TP rate and precision. Detailed grained classes definition.



(c) Miss classification results



(d) Application breakdown inside peer-to-peer class. Flows.

Fig. 2.

and Games achieved very poor TP rate, but reasonable precision. This means that although a large fraction of Streaming and Games flows are misclassified, flows classified as beeing in those classes are actually belonging to them. In order to better understand the misclassification problem, we present in Figure 2(c) the relative confusion matrix for the applications classes that performed poorly. Streaming is misclassified almost each time as Web traffic. A significant fraction of Ftp flows falls also into this class. For Games, flows are spread mainly over Web, P2P and Others classes. This result suggests that additional features should be used to better discriminate the poorly performing traffic types.

For the case of the more detailed application breakdown, the overall accuracy remains as high as 94.77%. As a result, we get very promising per class performance for most cases. The four p2p applications are well separated. The best accuracy is obtained for eDonkey: 95% and the worst one for Gnutella (50%). Bittorent and Ares have reasonable accuracy, around 80%. As we used the same features set and algorithm as in the previous step, the problem of Streaming and Games still remains. As for the confusion matrix for p2p applications, Gnutella flows fall mainly into the Bittorent class, whereas Bittorrent itself, is well classified. For both applications, the misclassified flows fall into non p2p classes.

This section aimed at pinpointing the need to tune the classification technique (one could change the algorithm or the flow features) to be used depending on the level of granularity and also on the application one wants to focus on. The confusion matrix appears to be a valuable tool in the tuning phase.

6 Mining the Unknown Class

We now focus on mining the unknown class, which was not classified by our reference tool. We first calibrated our statistical classifier using the flows classified by our DPI tool and then used it to the flows labeled as Unknown. Our algorithm outputs for every flow a prediction of class along with a probability of correct classification. We use the following heuristic to interpret results. If the prediction probability is higher or equal to a given threshold we assume the predicted class is correct. Otherwise we assume that the confidence is to low, so the flow remains unknown. Figure 3 depicts the cumulative distribution function of per flow confidence levels of classification for the unknown class. From the curve we can read what fraction of the flows can be classified assuming a specific confidence threshold. We test 2 threshold levels: 0.95 and 0.99. In this way, we are able to give insights about probable applications types generating unknown traffic. Results are provided in Table 3. Depending on the threshold, we are able to reveal between 50% and 63% of the unknown flows. Assuming a strict confidence level of 0.99, we are able to reduce the number of unknown flows by a factor of 2. Classified flows in our unknown class are mainly eDonkey and Web flows.

For the unknown flows case, we lack the reference point, so we are not able to directly assess the actual accuracy of the method proposed. However, we performed several side analyzes aiming to challenge the statistical predictions. We leverage the fact that we have the IP addresses and used port numbers of the endpoints of our flows. We perform the following tests:

- Reverse DNS lookup for each remote endpoint: We parse the answer, searching for meaningful keywords.
- For each flow classified as Web, we tested if there was indeed a Web server. We used wget service.
- Look for unknown remote endpoints in the known set: if a flow concerning this endpoint was once classified, e.g. eDonkey, other connections concerning same endpoint are very likely to be of similar type.

For the DNS lookup test, we obtained 79% of answers. In many cases, the host name can be meaningful. For example, many providers indicate in the host name that it is an ADSL host. Also searching for known ADSL providers names in the domains helps identifying home users[2]. Almost all end hosts, for flows predicted as p2p are ADSL hosts which seems to confirm the results of the classifier. What is more, we observe a large fraction of legacy ports of eDonkey or its simple variations in this set. Trying to connect using **wget** to endpoints for flows predicted to be Web, we obtained answers for only 7 % of the hosts, usually with https services, e.g., login pages of webmails. This explains why the ground truth tool failed in these cases. Looking at DNS resolutions of these endpoints, we observe a large fraction of home users rather than typical Web servers. It is hard to believe that home users were running so many Web services at the time of the capture. Given the results in Figure 2(c), it is very probable that a large

[2] We use keywords: ”DSL”, ”wanadoo”, ”free.fr”, ”club-internet” (popular french ADSL providers). We also check for specific operators hostname syntax to aviod confusion with providers website adress.

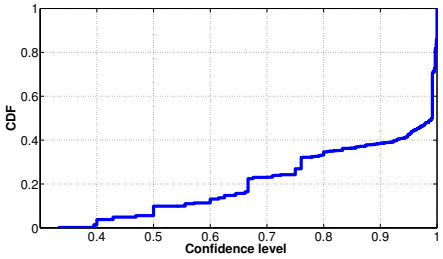


Fig. 3. CDF of fraction of flows classified depending on confidence level

Table 3. Unknown class predictions from C4.5

Unknown class predictions			
Confidence	0.95	Confidence	0.99
Class	[%]	Class	[%]
EDONKEY	23.07	EDONKEY	21.75
WEB	21.54	WEB	18.09
OTHERS	5.06	OTHERS	4.39
CHAT	1.75	CHAT	1.63
MAIL	1.69	GAMES	1.50
BITTORRENT	1.67	MAIL	1.07
GAMES	1.50	BITTORRENT	0.96
DB	0.56	DB	0.55
FTP	0.41	FTP	0.29
ARES	0.01	ARES	0.01
SUM	57.26	SUM	50.24

fraction of predictions for Web traffic are misclassified Streaming applications. What is more, the large fraction of Streaming in the known set consists of http Streaming. This explains why this traffic is statistically close to Web. Finally, for each endpoint in the unknown set, we look if it was present among the known flows. Only 18% of the unknown endpoints were present in the classified set, so it is not enough to draw overall conclusions. Endpoints identified contains mainly eDonkey users hosts. The method could work well if we had a larger users set, resulting in possibly larger fraction of the remote endpoints identified.

As a conclusion, except for the Web class our predictions are backed up by endpoints profiling. We might need a more precise classification method for Streaming application in order to provide more reliable predictions for the unknown class.

7 Discussion

In this paper, we have highlighted some key issues that arise when using statistical traffic classification tools. We have compared the outcomes provided by three different DPI tools. This comparison underscores the difficulty of assessing the results of any

statistical tool as the accuracy it achieves is partly correlated to the quality of the DPI tool used to establish the reference point. Reference point establishment is in fact a complex task and a good understanding of the design choices (e.g., Is http streaming classified as Streaming or Web?) is necessary to interpret the result.

We also shown that the exact level granularity that is requested might require changes in the method in terms of classification algorithm or flow features. From a methodological point of view, the confusion matrix turns out to provide a simple way of pinpointing the defaults of a classification method.

Last but not least, we have focused on the problem of mining flows classified as unknown by the DPI tool. We have shown how to take advantage of the confidence level provided by the classification algorithm to control the accuracy of the classification. We further demonstrated that simple heuristics could further back the results of the classifier and overcome the lack of reference point in this case.

References

1. Trestian, I., Ranjan, S., Kuzmanovic, A., Nucci, A.: Unconstrained Endpoint Profiling (Googling the Internet). In: Proceedings of ACM SIGCOMM 2008, Seattle, WA (August 2008)
2. Bernaille, L., Teixeira, R., Salamatian, K.: Early Application Identification. In: The 2nd ADETTI/ISCTE CoNEXT Conference, Lisboa, Portugal (December 2006)
3. Erman, M.A., Mahanti, A.: Traffic Classification Using Clustering Algorithms. In: Proceedings of the 2006 SIGCOMM workshop on Mining network data, Pisa (Italy), September 2006, pp. 281–286 (2006)
4. Dreder, H., Feldmann, A., Paxson, V., Sommer, R.: Operational Experiences with High-Volume Network Intrusion Detection. In: Proceedings of the 11th ACM conference on Computer and communications security, Washington DC, USA (2004)
5. Szabo, G., Orincsay, D., Malomsoky, S., Szabó, I.: On the Validation of Traffic Classification Algorithms. In: Claypool, M., Uhlig, S. (eds.) PAM 2008. LNCS, vol. 4979, pp. 72–81. Springer, Heidelberg (2008)
6. Paxson, V.: Empirically derived analytic models of wide-area TCP connections. IEEE/ACM Transactions on Networking 2(4), 316–336 (1994)
7. Kim, H., Claffy, K.C., Fomenkova, M., Barman, D., Faloutsos, M., Lee, K.Y.: Internet Traffic Classificatoin Demystified: Myths, Caveats, and the Best Practices. In: ACM CoNEXT, Madrid, Spain (December 2008)
8. Nguyen, T.T.T., Armitage, G.: A Survey of Techniques for Internet Traffic Classification using Machine Learning. In: IEEE Communications Surveys Tutorials, 4th edn. (2008)
9. Bro, `http://www.bro-ids.org/`
10. WEKA data mining, `http://www.cs.waikato.ac.nz/ml/weka/`
11. Tstat, `http://tstat.tlc.polito.it/`