Comparing Traffic Classifiers

Luca Salgarelli
DEA, Università degli Studi di Brescia, Italy
luca.salgarelli@ing.unibs.it

Francesco Gringoli
DEA, Università degli Studi di Brescia, Italy
francesco.gringoli@ing.unibs.it

Thomas Karagiannis
Microsoft Research, Cambridge, UK
thomas.karagiannis@microsoft.com

This article is an editorial note submitted to CCR. It has NOT been peer reviewed. Authors take full responsibility for this article's technical content. Comments can be posted through CCR Online.

ABSTRACT

Many reputable research groups have published several interesting papers on traffic classification, proposing mechanisms of different nature. However, it is our opinion that this community should now find an objective and scientific way of comparing results coming out of different groups. We see at least two hurdles before this can happen. A major issue is that we need to find ways to share full-payload data sets, or, if that does not prove to be feasible, at least anonymized traces with complete application layer meta-data. A relatively minor issue refers to finding an agreement on which metric should be used to evaluate the performance of the classifiers. In this note we argue that these are two important issues that the community should address, and sketch a few solutions to foster the discussion on these topics.

Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks**]: Network Operations

General Terms

Classification, algorithms

Keywords

Traffic classification, transport layer, measurement

1. INTRODUCTION

Accurate classification of network traffic has recently become a very popular research topic. The need to deploy effective QoS mechanisms in core networks, increase the effectiveness of security mechanisms and, probably, the growing interest on the debate on Internet neutrality have contributed to foster research in this area.

Traditional methods based on port number analysis and deep payload inspection exhibit a number of shortfalls: for example, they become completely ineffective when encryption is used to protect the payload. Statistical approaches could provide a promising alternative since they are based on the measurement of statistical features, such as packet length and inter-arrival times, whose measure does not require the observation of the application layer data. Despite many recent papers on this subject [1, 2, 3, 4, 5], it is still not clear if the preliminary results that have been reported so far in the literature could ever be extended so that a classifier will identify all the existing network applications.

We believe, however, that a key issue must be addressed to let the scientific community make steady progress in this direction: there is no agreed-upon methodology to compare different approaches in terms of accuracy of results and generality. Such methodology would certainly facilitate cooperative advances in finding optimal solutions for traffic classification. In this contribution we argue that there are at least two obstacles to the definition of a proper comparison procedure. The first one is a little thorny, because it has to do with the availability, or lack of thereof, of *shareable data sets*. The second problem is finding *appropriate metrics* to compare results, since no common approach to this, albeit simple to resolve, problem has been proposed.

We are aware that the (un-)availability of packet traces is an issue that has been discussed for many years in various contexts, including of course the traffic measurement community. With respect specifically to its implications on traffic classification, this short editorial has a twofold objective: to "stir up the crowd" yet one more time on the subject, and to discuss a potential new way to solve this problem.

2. SHAREABLE DATA SETS ANYONE?

An important issue that prevents researchers from effectively comparing the results of the various traffic classifiers is the inconsistency of the analyzed data sets. Some experiments rely on full-payload traces collected at public or private organizations [3, 5]: in such cases, since an accurate payload-based classification of the traffic is possible, the outcome of the statistical method can be compared with the actual protocol carried by each flow. This allows for a very precise assessment of the classifier's accuracy. However, one might argue that such traces can be less representative than the ones available from large backbone operators.

Other researchers have relied upon a mixture of full-payload traces and publicly-available, anonymized traces [4], or even just anonymized traces [6]. While in some cases the anonymized traces can be effective (see Section 2 of [6]), in general the unavailability of payloads renders ascertaining the actual protocol carried by each flow practically impossible. Therefore, the performance of a classifier when applied to such anonymized traces can never be fully trusted. Moreover, the use of anonymized traces makes the comparison of results between different classifiers more difficult, since different techniques can behave differently when classifying "masqueraded" traffic, i.e., flows that misuse well-known ports. In order to be able to compare classifiers fairly, one should be able to contrast their results when applied to the same data sets, or, at least, to similar network environments. This poses the following insurmountable requirement: *public access to common, shareable sets of full-payload traces.*

2.1 Solution A - Move classifiers

One obvious solution to the above issue would be to implement all the competing classifiers at the same location so that they can all access the same data set. The site could be located in the control room of an operator's backbone, where an access link and the traffic flowing through it can be monitored. By implementing the classifiers locally and by making them run in real-time, any privacy concerns related to the data sets could be largely overcome. One obvious but huge obstacle to this approach would be that no commercial or private network operator in their right mind would allow experimental software to be run regularly in real-time on live links for a number of reasons, because for example they could, at the very least, severely impact their performance.

This problem could be overcome by recording live traffic on large network-attached storage devices. Different classification techniques could later access all the stored traces, and run their algorithms on the same traces of recorded traffic, therefore simplifying the problem of comparing their results objectively. The main issues in this case would be protecting the privacy of the users who generated the traffic, and the security of the networks involved in the experiments. In addition, being research efforts, most of the classifiers that are being reported in literature are usually experimental. Optimizing such experimental software for a variety of environments in terms of operating system, hardware, etc., except for the one that were initially developed in, is a cumbersome task for such research groups. Software adaptation however would be a prerequisite if such a comparison in the aforementioned setting were to occur.

2.2 Solution B - Anon-pcap: ACLs for traffic traces

Besides moving classifiers, or releasing anonymized data sets (covered in the next section), we think that the research community should also take into consideration the development of a new model of traffic capture library, which would also serve as an interface to access recorded traces. Even though this library could be similar to existing ones such as *pcap* [7], we argue that it should add two capabilities. We will call this new packet capture library "*anon-pcap*".

First, it should be oriented to treating flows rather than packets. Second, the library should offer the capability to control which features of the captured traffic are exposed to its users. At the very least, any user should be able to retrieve statistical information for each flow, such as inter-packet delay and packet size, plus critical transport level information such as port numbers. A higher level of security would be required to access meta-data such as whether a given flow matches a particular pattern-based filter derived from a payload-based classifier, or whether each TCP packet contained any and which options, and so on. This approach could be extended to define different levels of clearance, and the associated different operations that each user could perform on the traces through this modified pcap library.

Note that, although *anon-pcap* would have to be substantially more complex than existing pcap anonymizers such as *tcpdpriv* [8], its flexibility could balance privacy concerns and requirements of fair comparisons among different classifiers. For example, the library could be setup to show, for a given set of classifiers, only the number of HTTP flows from a given trace that were correctly classified by a specific approach, while not giving direct access to any particular flow's payloads.

Anon-pcap would be ideally used as follows: organizations with access to backbone networks and experience in data collection efforts such as the *Cooperative Association for Internet Data Analysis* [9] would save encrypted, full-payload traces in secure storage spaces. Such traces would then be readable only through the *anon-pcap* library, adopting different levels of anonymization depending on the security clearance of the user. Running classifiers on the recorded traces would then be a matter of getting the necessary clearance (this would probably involve signing a ton of legal papers...), and obtaining an account with an interface to the traces through *anon-pcap*.

In this scenario, a regular user with standard privileges would only obtain, through the interface exposed by the library, fully anonymized packets, perhaps with pre-classification meta-data. A "superuser" would instead be able to access even the payload of specific types of traffic (say to port 443). There could be many security-clearance levels in between the two extremes above: in short, the *anon-pcap* interface would act as a sort of *"Access Control List (ACL)" system for traffic traces.*

The fact that recording and storing full-payload traces might be seen as unrealistic (which probably it is...) will be discussed in the final section of this editorial: we invite the reader to bear with us on this subject for the time being.

2.3 Solution C - Move [anonymized] data sets

As the saying goes, "if you can't bring Mohammad to the mountain..." While we think that making data sets recorded by large ISPs available through some form of APIs such as *anon-pcap* could be ideal, the ability to release anonymized traces that can be saved and used at a later stage would be very convenient for researchers.

A first form of anonymization that should not introduce new security concerns compared to current traffic anonymizers might be to release payload-stripped traces with *the addition of classification meta-data.* This would require each organization that releases traffic traces to perform pattern-matching classification on the payloads before anonymizing them, and release such classification meta-data together with the anonymized traces. This would allow researchers to use the traces knowing the "ground truth" as to which protocol generated each anonymized trace, and therefore to be able to realistically judge the accuracy of a classifier, at least compared to the accuracy of the pattern-matching mechanism used to produce the meta-data.

Another approach could be *"soft anonymization"*. This would require modifying trace anonymizers so that, instead of simply stripping-off the majority of each packet's payload, they would mangle it in such a way that the application layer protocol would remain recognizable, but any privacy and security related information would be overwritten. For example, they would anonymize an HTTP trace by overwriting any URL, DNS names and privacy-sensitive information in HTTP requests, and any content or privacy-sensitive information in HTTP responses. This would still make each trace semantically valid (at the application layer), while possibly removing privacy concerns. Clearly, while this approach would generate more useful traces for the purpose of traffic classification, it would be difficult to put into practice, because it would require implementing any known protocol's state machine in the anonymizer. In addition, one might argue that if current anonymizers, by stripping off any transport-layer payload, leak enough information for attackers to use against a given network [10], it could be almost impossible to guarantee the security of such "soft anonymizers". It is further questionable if it is even possible to distinguish between nested application layer protocols within the layer-4 payload, especially taking into consideration the numerous applications tunneled over HTTP (e.g., streaming, mail, chat, etc.)

3. FINDING THE RIGHT METRIC

The use of a common metric, or at least the definition of an agreed-upon set of parameters that can be used to assess the performance of traffic classification mechanisms constitutes the basis of any comparison of traffic classifiers, and indeed of any mechanism. Although this is seemingly a rather simple problem to solve, many papers that have dealt with traffic classification in the past few years have all used different metrics to present their results. In some cases, some of the metric parameters used in a paper can be simply derived from similar parameters used in other works, possibly with a different name. However, sometimes parameters defined in a paper simply cannot be found in others, or are related by more complex expressions that make comparisons tricky.

Let us compare as an example three relatively recent papers on traffic classification: T. Karagiannis et al. [3], L. Bernaille et al. [4] and Crotti et al. [5].

Karagiannis et al. rely on two performance parameters, *completeness* (we will refer to it as C) and *accuracy* (A). With completeness they indicate the ratio between the number of flows assigned by their classifier to a given traffic class and the total number of flows of that class present in a given data set. Accuracy is defined as the percentage of flows that the classifiers labels correctly.

Bernaille et al. use the term *True Positive* (TP) to indicate what the previous paper indicates as *accuracy.* In addition, they define the term *False Positive* (FP, referred to a given protocol p) as the ratio between the number of flows incorrectly labeled by their classifier and the total number of flows in their data set *not* including protocol p. They also introduce other parameters, such as *False Negative* to help them evaluate how their mechanisms deals with unknowns.

Crotti et al. quantify the performance of the classifier by using two main parameters. *Hit Ratio* (H_r) is defined as the ratio between the flows of a given protocol that are classified correctly versus the total number of flows of the same protocol present in the data set. The parameter *False positive* for a given protocol p (F_+), refers to the ratio between the number of flows that were incorrectly classified as protocol p and the number of flows labeled as the same protocol[1].

Although some parameters can be easily derived from others with simple algebraic operations[2], others simply cannot. Furthermore, even though some of the parameters refer to the same broad concept, such as "False Positive", they are used differently in different papers[3], confusing things even further.

Summarizing, we see a problem not so much in the *terminology* that is used, but in the fact that different papers rely on sometimes very different parameters to validate their approach. Furthermore, parameters that are used in a paper to verify the classifier's performance in one area, say, for example, its ability to deal with "unknowns", are not always present in other works, at least not in the same form. This makes the effective comparison of different approaches much more difficult than it should be.

3.1 A proposal for a common metric

We think that it could be useful, for the purpose of making comparisons among different traffic classification mechanisms, to have each paper base their evaluation on a common set of parameters. Of course each paper could define its own metric, but we argue that it would be useful if each metric would include at least the following parameters.

True Positive (TP) : An indication of true positives. This could be defined similarly as the "Accuracy" parameter defined in [3].

False Positive (FP) : An indication of false positives, **not** defined over the total number of flows, but over the number of flows assigned by the classifier to a given protocol. In practice, parameter FP answers the question "out of x number of flows classified as protocol p, what's the fraction that was *not* really produced by p?" We think this parameter is very useful for assessing the performance of a classifier in real-world scenarios, where it would express the trustworthiness of the classifier in dealing with a given traffic class.

Unknown (U) : An indication of how the classifier deals with unknown traffic, i.e., traffic that the classifier has not been trained for. This could be easily derived by linear combination of the terms TP and FP, but it would be easier if it would be declared explicitly.

4. ADDITIONAL OPEN ISSUES

While we have argued over the previous sections that data-sharing and metric definition are the major issues that inhibit progress, comparing classifiers involves a number of other complications that could possibly be addressed more easily, yet are still important. Specifically:

- *The definition of application classes is different across the classifiers.* For example, a class of "Network Management" protocols may comprise different applications across the various techniques. Thus, comparing the performance of the classifiers must account for such discrepancies.

- *Scope of the classifier.* The scope and goal of the different methodologies render them not directly comparable in some cases. For example, BLINC [3] attempts to identify peer-to-peer traffic irrespective of the specific underlying application (i.e., Kazaa, BitTorrent, etc), in contrast to the Bernaille et al.'s methodology [4]

[1] The rationale behind this definition will be clear in Section 3.1.

[2] For example, Karagiannis et al.'s A can be defined as $(1 - F_+)$, where F_+ is the False Positive parameter defined in [5].

[3] Compare, for example, the definition of False Positive in [4] and [5].

which distinguishes between the various peer-to-peer applications.

- *Evaluation of unknown/encrypted traffic.* Even assuming that a payload trace is possible to be properly anonymized and shared publicly, still a portion of that trace will potentially contain unknown and/or encrypted traffic. As one of the motivating forces behind most classification techniques is the treatment of such unknown or encrypted traffic, or even further the identification of new applications in the existing traffic classes (e.g., a new peer-to-peer application), how can we effectively compare methodologies regarding their results for such traffic?

5. FINAL REMARKS

In this short contribution we have argued that research in traffic classification could proceed more speedily towards its goals if the community is able to effectively and precisely compare different approaches. At least two main issues must be resolved for this to happen: we must find a way to share data sets, and a common metric that captures the capabilities of each classification approach.

Although the first problem has been debated for a long time within several communities, in this editorial we try to summarize and discuss three approaches. What we called "Solution A" is simple, but to the best of our knowledge has not been applied so far. Nothing makes us believe that this will change in the future. "Solution C" requires an enhancement to anonymization libraries, and could be appealing to some types of research.

"Solution B" is a new type of approach, and relies on a controlled access to traces, which are saved with their full-payloads, and are therefore amenable to any kind of research in this area. We realize that having large organizations record and store full-payload traces, even if in encrypted format and with access strictly controlled through the *anon-pcap* library, could be a Utopian idea. However, we feel that it is important that even an extreme scenario like this one should be discussed within the community. After all, the design and development of a system such as *anon-pcap* could be an interesting research project by itself. Furthermore, we believe that the basic idea of realizing an ACL-like system for accessing traffic traces has its own merits, and could bring actual benefits to the measurement community even if not applied to the scenario of full-payload traces that we described in this editorial.

We then have explained why we think that defining a common evaluation metric is important, and, starting from the main evaluation parameters used in recent papers, we propose the high-level elements that should be part of a common metric.

Finally, it should be clear that there are many other issues that hamper cooperative advances in traffic classification, such as the evaluation of encrypted traffic. While we mentioned a few of those in this editorial, an exhaustive list would probably be much longer.

6. REFERENCES

[1] A. W. Moore and K. Papagiannaki. Toward the Accurate Identification of Network Applications. In *Proceedings of the 6th Passive and Active Measurement Workshop (PAM 2005)*, pages 41–54, October 2005.

[2] A. W. Moore and D. Zuev. Internet traffic classification using bayesian analysis techniques. In *SIGMETRICS '05: Proceedings of the 2005 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pages 50–60, Banff, Alberta, Canada, June 2005.

[3] T. Karagiannis, K. Papagiannaki, and M. Faloutsos. BLINC: multilevel traffic classification in the dark. In *SIGCOMM'05: Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 229–240, Philadelphia, PA, USA, August 2005.

[4] L. Bernaille, R. Teixeira, and K. Salamatian. Early Application Identification. In *The 2nd ADETTI/ISCTE CoNEXT Conference*, Lisboa, Portugal, December 2006.

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

[6] C. V. Wright, F. Monrose, and G. M. Masson. On Inferring Application Protocol Behaviors in Encrypted Network Traffic. *Journal of Machine Learning Research*, 7:2745–2769, December 2006.

[7] Tcpdump/Libpcap. http://www.tcpdump.org.

[8] tcpdpriv: eliminating confidential information from packets. http://ita.ee.lbl.gov/html/software.html.

[9] The Cooperative Association for Internet Data Analysis (CAIDA). http://www.caida.org.

[10] R. Pang, M. Allman, V. Paxson, and J. Lee. The Devil and Packet Trace Anonymization. *ACM SIGCOMM Computer Communication Review*, 36(1):27–38, January 2006.