Service-Based Traffic Classification: Principles and Validation

Service-Based Traffic Classification: Principles and Validation

M. Baldi[■], A. Baldini[•], N. Cascarano[■], F. Risso[■]
[■]*Politecnico di Torino* [•]*Cisco Systems, Inc.*
<first.last>@polito.it *abaldini@cisco.com*

Abstract

This paper presents a novel approach in traffic classification that is based on the identification of the service that generates the traffic. This method is, in some sense, orthogonal to current approaches and it can be used as an efficient complement to existing methods to reduce computation and memory requirements. Experimental results on real traffic confirm that this method is extremely effective and may improve considerably the accuracy of traffic classification, while it is suitable to a large number of applications.

1. Introduction

Traffic classification is one of the hottest topics in computer networks. On the one side, network managers want to know precisely the type of traffic transmitted over their networks to enforce polices of various nature (e.g., quality of service, security, etc.). On the other side, an increasing number of applications tend to hide their behavior (through encryption, tunneling, etc.) trying to avoid limitations imposed by such policies.

This paper presents the novel concept of *service-based classification* that, in some respect, is orthogonal to existing approaches and can be used to complement them. Service-based classification exploits information about previously discovered services to classify traffic flows. Main advantages of this method are robustness, accuracy, a limited use of processing power, reduced memory requirements, and the capability to use any classifier in the early stage of the classification (namely, the *service identification* phase).

The objective of this paper is to present the basic principles underlying service-based classification, describe a first implementation, and report on some general experiments. The purpose of such experiments is to demonstrate the feasibility of the classification approach and provide an initial quantitative assessment of the potential benefits it can bring, while leaving a both wider and more in depth analysis and performance evaluation to a later, complete publication. Section 2 describes the broad categories in which classification approaches fall and discusses three solutions that have some similarities with service-based classification. Section 3 describes the service-based classification idea, while some details about our implementation are given in Section 4. Section 5 presents experimental results and conclusions are drawn in Section 6.

2. Existing Classification Approaches

Traditionally, traffic classification relies on *transport layer information*, i.e., source and destination TCP/UDP ports. However, this method has many well known limitations that make it quite imprecise and inefficient despite its extensive usage.

Payload-based classification, declined in different flavors [1], is applied by most commercial solutions for various purposes ranging from statistics to security, because it provides the best trade-off between classification accuracy and coverage in terms of number of recognizable protocols. It relies on inspection of data transported within packets, i.e., the headers of the higher layer protocols — possibly up to the application layer payload searching for patterns or regular expressions that can uniquely identify each protocol. Known problems of payload-based classification are (*i*) high sensitivity to packet loss and TCP/IP fragmentation and segmentation issues, (*ii*) hard and time-consuming task of creating protocol signatures, that are crucial to the effectiveness of the solution, (*iii*) encryption and/or tunneling that hinders access to data contained into application layer headers and payloads, and (*iv*) significant requirements in terms of computational and memory resources that actually make traffic classification at high line rates difficult.

Due to the high computational requirements of deep packet inspection, payload-based classification algorithms usually limit pattern searching to the initial packets of each flow. According to this method, named Packet Based – Flow State in [1], once the protocol transported by a flow has been recognized, the flow identifier (i.e., the 5-tuple including IP addresses, ports, and transport layer protocol) and the corresponding application-layer protocol are added to a data structure in memory, often called *session table*, that is maintained as long as the flow is active[1]. In case of large networks, the size of such per-flow state grows significantly and this might become an issue.

Another approach in traffic classification relies on *behavioral* techniques, whose main assumption is that each application is characterized by some specific behavior. Applications can then be identified by just gathering information at different levels (e.g., packet inter-arrival time, jitter, packet size, etc.) and analyzing it (e.g., from a statistical point of view), often without inspecting protocol headers and application data transported. Therefore behavioral algorithms are not affected by any of the shortcomings of payload-based algorithms related to information hiding (e.g., by encryption) or camouflage (e.g., by using ports typically associated to services). However, behavioral algorithms have some common limitations mostly stemming from the need for training with a pre-classified traffic trace. Producing training traces of proven accuracy, possibly for each of the network conditions in which the classifier is to be deployed, is costly and sometimes unfeasible.

BLINC, a behavioral algorithm proposed by Faloutsos et al. [3], introduces the idea of looking at the "social" behavior of each host, which is somewhat similar to the service-based classification approach proposed here. However, while BLINC uses specific social behaviors characterizing an application in order to classify the packets of a flow as belonging to that application, service-based classification relies on the fact that most applications display the specific behavior of offering a service at fixed "network coordinates", i.e., at a specific port on a specific host. Hence, although the two approaches have a common inspiring idea, the resulting solutions are very different.

An idea similar to service-based classification was already used in previous works, namely [4] and [5], but with a different purpose. [4] proposes to keep a history of already classified flows to build a knowledge base for particular host/port combinations that can be used to *validate* future classification results by checking their conformance with roles previously observed for the same host. Hence, in [4] historical service data is not used for classification, but rather for validation purposes. [5] proposes a statistical method to classify peer-to-peer traffic; among the three techniques jointly deployed, one consists in keeping a table that contains IP addresses of hosts that are at some point identified as nodes of a peer-to-peer overlay, or that are identified as known (traditional) services (e.g., HTTP server). All flows whose packets contain an IP address included in the P2P table are flagged as "possible P2P" and analyzed in more detail. Hence, this approach uses the host history to single out packets that require further analysis in order to identify the application they belong to. Service-based classification instead relies on other classification approaches to initially identify a service and then uses it in the following classification; i.e., classification is based on the very recent history of a host providing a specific service.

In conclusion, while payload-based methods are usually precise enough and offer excellent coverage (in terms of protocols detected), they are expensive from the memory and computational points of view. Behavioral approaches are promising, but usually limited in terms of coverage, and often suffering from many limitations due to their training requirements. The service-based approach presented in this paper is a breakthrough technology that includes as many advantages as possible from both categories while reducing disadvantages.

3. Service-Based Classification

Service-based classification is a surprisingly simple idea that relies on the observation of how hosts usually interact and on the assumption that certain hosts, typically called *servers*, perform similar interactions, usually offering a *service*, with multiple other hosts over a certain time span. Section 5.2 verifies this assumption, which provides the foundation of our method, through experiments on real network data.

The basic principle in service-based classification is that knowing which service is offered at given "*network coordinates*" (IP address and TCP/UDP port pair) a classifier can infer that all sessions directed toward that coordinates will access such service. For example, while a port based classifier assumes that a session is transporting HTTP because it is connected to TCP port 80, a service-based classifier upon *getting to know* that `www.polito.it` is running a web server

[1] While the *session table* is usually associated to payload-based techniques, in fact it has a broader usage. Particularly, all methods that rely on session identification need to maintain this information.

on TCP port 80 stores the triple identifying it — i.e., IP address (of the server), TCP/UDP port (at the server), and transport protocol — in a *Service Table* and uses it to classify traffic related to such service.

The same principle can be applied to hosts running peer-to-peer applications. In this case the application has a client part and a server part running simultaneously. The port used by the server might not be known in advance, but such port usually does not vary very frequently and is reused many times for the same instance of the peer-to-peer application. Also peer-to-peer applications that use the same port for both the server part and the client part, such as Skype for example, are handled properly. After a peer A has received a connection to its server part, a triple containing its IP address and port is created in the service table as a service. When its client part connects to another peer B, the service-based classifier classifies the corresponding packets according to either A's service entry or B's service entry. Although classification based on A's service entry is in principle mistaken as packets are being exchanged as part of a session whose server side is B, the packets are anyway correctly classified as belonging to the peer-to-peer application at hand.

Finding out which service is running at a certain IP address/port pair, here called *service identification*, is orthogonal to the service-based approach: in principle, any classification method can be used to perform service identification (payload-based, heuristic, or even manual inspection).

Service-based classification features interesting advantages over other classification methods. *Encrypted traffic* at application layer can be properly classified provided that the corresponding service has been previously identified, i.e., it has an entry in the service table. It offers *pattern segmentation transparency*, i.e., a flow can be properly classified even though protocol identifying patterns are split across multiple packets, avoiding the complexity of reassembling application data units. A service-based classifier needs to maintain only information about services (i.e., IP address, port, transport protocol and service offered) independently of the number of traffic flows actually using such services; hence it has *limited memory requirements*. The limited amount of state information kept by a service-based classifier impacts its (*i*) *scalability* and performance in terms of (*ii*) *lookup time* and (*iii*) *hardware implementation* deploying faster on-chip memory. Classification of a packet belonging to a known service requires a single lookup on the three fields (IP address, port and transport protocol) in a relatively small lookup table, therefore with *low computational cost*. Moreover, service identification, which might have higher computational cost, is expected to be performed only on a small fraction of the packets and can be even performed offline; in any case, service identification is orthogonal to the service-based method. Finally, as mentioned above, service-based classification is among the few methods that enable *early classification*, i.e. classifying the first packets (e.g., a TCP SYN) within each session (of a previously identified service), while other methods need to process the first few packets within each session before being able to classify the rest.

Service-based classification also has some potentially critical issues. Its effectiveness, in terms of minimizing both misses and wrong matches, as well as its performance, heavily depend on identification of network services that must be as accurate as possible. A wrong entry in the service table leads to wrongly classifying a potentially large number of flows, while a missing entry possibly leads to both a failing classification of a large number of flows and deploying significant amount of computational resources in an effort to identify the service being used, e.g., by deeply inspecting the corresponding packets.

In addition, since service-based classification does not keep information about individual sessions, it is not suitable for applications requiring that granularity level, such as, for example, per-session enforcement of quality of service policies. A service-based classifier can be customized for such applications by keeping an additional session table for those services requiring so.

Other potential issues include *dynamic sessions* and *proxies*. With respect to the first problem, some applications (e.g. FTP, SIP) use a control session to dynamically negotiate the port of the data transfer session that cannot be associated to a stable service based on its ports. Consequently, the benefits of service-based classification are limited to the identification of packets belonging to the control session on which a deep inspection is required to find out the port used for the data transfer.

The second problem is related to proxies (and SOCKS servers), which handle the access to various types of services (HTTP, FTP, etc.) on behalf of different clients using the same transport-layer port. The service-based classifier has no problem in classifying traffic from these servers to the target service, but it is unable to distinguish the service contacted when analyzing the traffic between clients and the proxy/SOCKS server. Like for services that use

dynamic sessions, the service-based method can be used to identify the packets on which a deep inspection is to be performed.

Finally, the service-based classifier is not effective with traffic encrypted at IP level, e.g. with IPsec.

4. Implementing a service-based classifier

Notwithstanding the conceptual simplicity of service-based classification, careful consideration of some issues is required to ensure proper operation. This section presents a possible approach; other strategies are sensible.

4.1 Service identification

Given our expertise and previous work, a payload-based implementation of a service identification module has been an obvious choice. In particular, an existing packet processing engine based on the Network Packet Description Language (*NetPDL*) [2] has been deployed[2]. NetPDL is an application-independent packet format description language. It has proved extremely effective and robust with respect to traffic classification [1], thanks to an extension that enables management of lookup tables, originally conceived to maintain transport-level sessions [2].

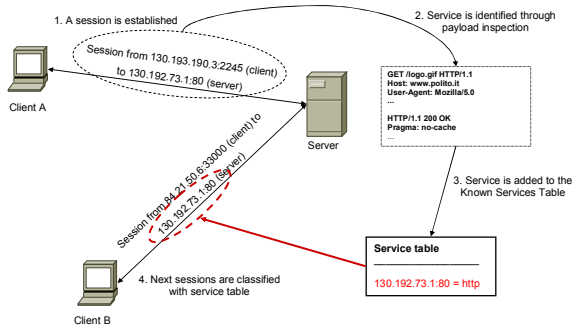

Figure 1. Service identification.

The high flexibility of NetPDL makes the engine suitable for the implementation of the service-based classifier as well. The main modification required to the NetPDL engine for the implementation of a service-based classifier is the addition of some new tables, such as the service table.

Since the server side of the communication cannot be inferred on a packet-basis, the service table is looked up twice: once with the source network coordinates and once with the destination ones. If one of these lookups is successful, the packet is classified through the service-based method. Otherwise, as depicted in Figure 1, the service identification module performs a payload-based classification to possibly introduce a new entry in the service table containing the IP address and the transport layer port used by the server side of the session and the application protocol associated. Any new packet toward this "known service" can subsequently be classified directly through the information kept in the service table as described above without any further processing (e.g., payload inspection).

4.2 Distinguishing clients and servers

The identification of the server side of a session is not straightforward. Observation of the SYN and ACK flags during in the three-way handshake can be leveraged of for a TCP session. Our implementation uses an additional lookup table, called *Candidate Service Table*, in which a new entry is added with the IP address and port of a host that accepted an unclassified TCP session by generating a TCP packet with both the SYN and ACK flag enabled. The Candidate Service Table is required to keep track of the server side of a session while waiting that the service is possibly identified, e.g., through payload inspection, which is possible only after the session has been successfully opened. Upon service identification, the server information is moved from the candidate service table to the service table.

Entries of the candidate service table are subject to fast ageing (about ten seconds) in order to avoid their number to explode over time due to sessions opened by unidentified services, unsuccessful handshakes, or unused opened sessions, as in cases of malicious activity such as SYN flooding and port scanning.

Identifying UDP servers requires a different approach since explicit information, like the SYN flag, is not available. Although, especially with the growing adoption of broadband multimedia applications, UDP is expected to significantly increase its traffic share, possibly becoming predominant, this paper focuses on TCP traffic, which as of today accounts for the vast majority of data. UDP traffic classification, that requires a non-straightforward extension of what is proposed here, is left to a companion future paper.

4.3 Service table maintenance

Prompt elimination of service table entries once the corresponding service is no longer active is important in order to avoid both the explosion of the table and classification errors due to services offered only

[2] Available at http://www.nbee.org/.

temporarily. A possible approach is to purge an entry that does not make a hit for a certain amount of time, hereafter referred to as *service inactivity timeout*. As a further refinement, the service inactivity timeout can be differentiated on the basis of service classes. For example, an SMTP server — usually contacted only few times in a day, but providing its service over a very long time period — can be assigned to a class with a long service inactivity timeout. Vice versa, a peer-to-peer application can be assigned to a class with short inactivity timeout. The choice of inactivity timeout is non trivial and its impact requires further study that is outside the scope of this work. Service table maintenance can be done by an independent process, thus not impacting performance and scalability of the classifier.

5. Experimental Evaluation

Network traffic is analyzed with the purpose of first quantitatively estimating the potential benefits of service-based classification and then substantiating its underlying service stability assumption.

5.1 Benefit estimation

Traffic traces have been analyzed with the goal of determining the number of service entries required by a service-based classifier, compared to the number of session entries required by a classifier based on session identification. In the analysis a TCP session is considered closed when a FIN or RST packet is observed; a 10-minute session inactivity timeout is used in case of abnormal termination. Analogously, services are considered closed if no traffic is observed during an idle period of the same duration. The obtained results can be considered a lower bound of the service table size since they account for the sessions/services present and actually active at any given time.

Figure 2 shows, for each minute, the number of active traffic sessions and the corresponding number of services on the (100 Mbps) link connecting our university network (about 6,000 hosts) to the Internet measured using *Tstat*[3] over a 7-day period. Figure 3 shows the same figures for a traffic trace captured on a 150Mbps trans-pacific backbone link and available in the MAWI archive[4]. The average on the whole observation period of the *session to service* ratio is about 20 for both traces, which means that a service table requires roughly 20 times fewer entries than a session table. Furthermore, a service entry is smaller than a session entry as less information is to be stored. This is beneficial in terms of memory requirements as well as of both processing requirements and performance for service information look-up.

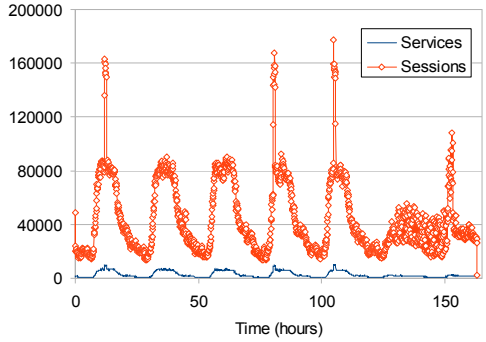

Figure 2. Services vs. sessions on a campus network.

Although these numbers show clearly the advantage of the service-based classification (at least in the tested environments[5]), they are derived under the assumption that services are stable over time. This assumption will be empirically demonstrated through the experiments reported in Section 5.2.

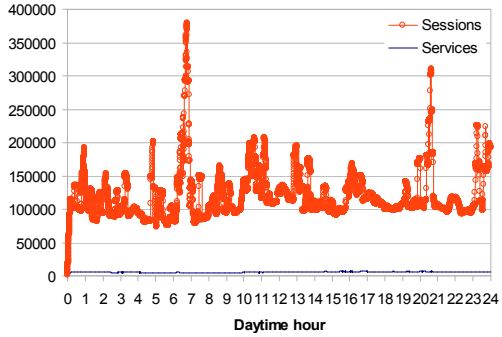

Figure 3. Services vs. sessions on a trans-pacific link.

5.2 Service stability

As mentioned in Section 3, the fundamental assumption of the service-based classifier is the stability of services: packets are misclassified when a new service is offered at the same "network coordinates" where another service was previously offered. Service stability has been assessed using a probe jointly developed at University of Brescia and Politecnico di Torino that logs on a centralized server the name of each application creating a network socket on its host. By running the probe on all the hosts of a network used for the evaluation of a classifier, the

[3] Available at http://tstat.tlc.polito.it/
[4] Available at http://tracer.csl.sony.co.jp/mawi/

[5] The gain of service-based classification may be more limited in different network conditions; e.g., the *session vs services* ratio may be smaller in case of a large fraction of P2P traffic.

application generating each session can be known precisely. The tool has been installed on 11 hosts (with Linux, Windows and MacOS-X operating systems, running several applications; among others Skype, eMule, Joost, uTorrent) and the traffic produced has been captured for 4 days. In order to study the relationship between services (i.e., their network coordinates) and applications the traffic traces have been analyzed by means of a payload-based classifier.

Table 1. Services and applications.

Observed sessions	40503
Observed services	21675
Observed applications	81
Services with univocally classified sessions	21042
Services with least one unknown session	633

The experiment, whose results are summarized in Table 1, found no cases in which sessions belonging to different applications were using the same network coordinates, i.e., could be associated to the same service, which provides a strong (albeit experimental) foundation to our method: errors due to service instability are, to say the least, really rare. The experiments also demonstrated the limitation of the payload-based classifier:

- A number of sessions (last row in Table 1) classified as unknown, mostly due to signature-related problems (e.g., a signature split across two packets).
- (Very few) sessions to the same service classified as different applications (not shown in Table 1).

6. Conclusions

This paper presents a new idea for traffic classification, named *service-based classification*, that proposes traffic to be classified according to the *service* it belongs to. Consequently, analysis of all sessions is not required: once a service has been previously recognized, sessions are classified as they access it — even if encrypted at application-layer. Service-based classification is somewhat orthogonal to the other classification techniques that are used to identify services. As such, it introduces the concept of *fast path* in traffic classification: the vast majority of the traffic is processed with a limited use of processing and memory resources —ultimately in a short time — while a *slow path* is used in a limited number of cases, i.e., for packets belonging to sessions not accessing an identified service. This makes the deployment of sophisticated methods for service identification, such as behavioral algorithms, or even a combination of them with payload-based classification, practical notwithstanding their high complexity and processing requirements as they can be executed only on a small fraction of the packets.

Experimental data confirm that services are very stable, i.e., offered consistently at the same network coordinates (IP address and transport port) even over long periods, making this extremely simple, efficient and robust method viable. Presented results in terms of efficiency show a 20 fold reduction in the number of entries in data structures compared to session based classifiers; furthermore each entry is half the size. Real-time measurements (not reported in detail for the sake of brevity) on the traffic transmitted on the link connecting our University campus network to the Internet show that the service-based classifier successfully classifies roughly 81% of the packets and 93% of the traffic (in terms of bytes). Furthermore, service based classification is among the few methods that guarantee early classification, including the initial TCP handshake of a session. Among the few drawbacks of this method is the impossibility to classify encrypted IPsec traffic.

7. References

[1] F. Risso, A. Baldini, M. Baldi, P. Monclus, O. Morandi. Lightweight, Session-Based Traffic Classification. IEEE International Conference on Communications (ICC 2008) - Advances in Networks & Internet Symposium, Beijing, China, May 2008.

[2] F. Risso, A. Baldini, F. Bonomi. Extending the NetPDL Language to Support Traffic Classification. In IEEE Globecom 2007, Washington, D.C, USA, Nov. 2007.

[3] T. Karagiannis, K. Papagiannaki, and M. Faloutsos. BLINC: Multilevel Traffic Classification in the Dark. In ACM SIGCOMM, Philadelphia, PA, Aug. 2005.

[4] W. Moore, K. Papagiannaki. Toward the Accurate Identification of Network Applications. International Workshop on Passive and Active Network Measurement (PAM 2005), Boston MA , USA, vol. 3431, Mar. 2005

[5] T. Karagiannis, A. Broido, M. Faloutsos, Kc claffy. Transport layer identification of P2P traffic. 4th ACM SIGCOMM conference on Internet measurement table of contents, pp. 121 - 134, Taormina, Italy, Oct. 2004.