On the Validation of Traffic Classification Algorithms

Géza Szabó, Dániel Orincsay, Szabolcs Malomsoky, and István Szabó

TrafficLab, Ericsson Research, Budapest, Hungary
{geza.szabo,daniel.orincsay,szabolcs.malomsoky,istvan.szabo}@ericsson.com

Abstract. Detailed knowledge of the traffic mixture is essential for network operators and administrators, as it is a key input for numerous network management activities. Traffic classification aims at identifying the traffic mixture in the network. Several different classification approaches can be found in the literature. However, the validation of these methods is weak and ad hoc, because neither a reliable and widely accepted validation technique nor reference packet traces with well-defined content are available. In this paper, a novel validation method is proposed for characterizing the accuracy and completeness of traffic classification algorithms. The main advantages of the new method are that it is based on realistic traffic mixtures, and it enables a highly automated and reliable validation of traffic classification. As a proof-of-concept, it is examined how a state-of-the-art traffic classification method performs for the most common application types.

1 Introduction

The aim of traffic classification is to find out what type of applications are run by the end users, and what is the share of the traffic generated by the different applications in the total traffic mix. Research for better and better traffic classification methods is blooming with the constant increase of network capacity, the emerging application types, and common usage of traffic deceiving techniques. However, the objective comparison of these methods has not been possible yet due to several reasons. Firstly, there are no perfectly classified traffic traces available. Moreover, the validation is typically done with another specific classification method. This situation results in such anarchy that papers can state nearly anything about their introduced method as there is no chance to check it by others or verify with a commonly known and accepted reference test.

In this paper we provide a validation method, which can reliably test the accuracy of traffic classification algorithms. In practice, the objective is typically to identify applications in passively observed traffic. We believe that such a classification method can be convincingly validated only by an active test, for which a number of requirements are fulfilled, such as:

– It should be independent from classification methods, i.e. the validation of a classification method by another one must be avoided,

M. Claypool and S. Uhlig (Eds.): PAM 2008, LNCS 4979, pp. 72–81, 2008.

- About each packet the test should provide reference information that can be compared to the result of the classification method under study,
- The test should be deterministic, meaning that it should not rely on any probabilistic decisions,
- Feasibility: it should be possible to create large tests in a highly automated way, and
- The environment where the active measurements are collected should be realistic.

The paper is organized as follows: in Section 2 an overview of existing traffic classification methods is provided together with a discussion of the techniques and datasets used to validate them. In Section 3 a new method is introduced which makes it possible to validate traffic classification methods. In Section 4, a state-of-the-art traffic classification method is validated as a proof-of-concept, demonstrating how it performs for several application types that are included in the example test.

2 Existing Traffic Classification Methods and Their Evaluation

Currently, there are a couple of fundamentally different approaches for traffic classification. In this section we browse through the state-of-the-art traffic classification methods. We discuss briefly the accuracy of these methods, which is relevant here, because in most cases a classification method is validated by another classification method ([12], [19], [18]).

The most accurate traffic classification would obviously be complete protocol parsing. However, many protocols are ciphered due to security reasons (SSH [5], SSL [4]). Also some are proprietary, thus there is no public description available (Skype [6], MSN Messenger [2], World of Warcraft [9], etc.). In general, it would be difficult to implement every protocol which can occur in the network. In addition, even simple protocol state tracking can make the method so resource consuming that it becomes practically infeasible.

- **Port based classification:** In the simplest and most common method the classification is based on associating a well-known port number with a given traffic type, e.g., web traffic with TCP port 80 [1]. This method needs access only to the header of the packets. The port based method becomes insufficient in many cases, since no specific application can be associated to a dynamically allocated port number, or the traffic classified as web may easily be something else tunneled via HTTP. The port based method is a standard, common method, however due to the above problems, it can not be considered to be reliable.
- **Signature based classification:** To make protocol recognition feasible, only specific byte patterns are searched in the packets in a stateless manner. These byte signatures are predefined to make it possible to identify particular traffic types, e.g., web traffic contains the string 'GET', eDonkey P2P

traffic contains 'xe3x38'. The common feature of the signature based heuristic methods is that in addition to the packet header, they also need access to the payload of the packets. Especially in the case of well documented open protocols, this method can work well. However, in practice only extensive experiences with real traces provide enough feedback to select the best performing byte signatures. For example, the 'GET' message could be the criterion of both HTTP and Gnutella (a P2P protocol), thus this signature alone, without applying other criteria, is not proper for accurate traffic classification. The main disadvantage of the signature based method is that the signatures have to be kept up to date, otherwise some applications can be missed, or the method can produce false positives. The other disadvantage is that this method cannot deal with encrypted content.

Authors of [16] validated their constructed signature database by manually checking the false positive ratio of their technique. Their approach was to investigate TCP connections which were identified as P2P connections. If in fact the content of the connection did not belong to a P2P protocol they counted the connection as a false positive. By the term 'active measurements' they mean that specific traffic type is generated on purpose, thus what kind of traffic is expected can be exactly known at a certain point in the measurement. This is the most common way of developing signature databases as this method ensures that the traffic is sterile, i.e., only a specific application is measured at a time. The measurements they used are not public, therefore others cannot use them as reference.

- **Connection pattern based classification:** The basic idea is to look at the communication pattern generated by a particular host, and to compare it to the behavior patterns representing different activities/applications [12]. The connection patterns describe network flow characteristics corresponding to different applications by capturing the relationship between the use of source and destination ports, the relative cardinality of the sets of unique destination ports and IPs as well as the magnitude of these sets. The application specific behavior patterns are often difficult to find, especially if multiple application types are used simultaneously. In order to identify a communication pattern reliably, the method needs a lot of flows coming from and going to a host.

  Authors of [12] validated their method by using signature based classification. As there are no commonly accepted and well performing byte-signatures, authors constructed their own signature database.
- **Statistics based classification:** In statistics based classification some statistical feature of the trace is captured and used to classify the network traffic. To automatically discover the features of a specific kind of traffic, the statistical methods are combined with methods coming from the field of artificial intelligence. The most frequently discussed method is the Bayesian analysis technique as in [14], [19], [13], [11], [10]. The basic requirement of these techniques is previously hand-classified network traffic which provides them with training and testing data-sets. In order to reach sufficient accuracy, the ratio of these data-sets should be about 1:1.

In [19] authors used port based classification to validate their method. They assume that for the ports they use in the study the majority of the traffic is from the expected application. In this case, it is most likely that few 'wrong' flows would decrease the homogeneity of the learned classes. Therefore their evaluation results can be treated as lower bound of the effectiveness. They also do not consider traffic of the selected applications on other than the standard server ports. Authors of [11] worked with commonly available traffic traces, but these traces contained only packet headers which excludes such reliable validation methods which are based on packet payload. In [10], the traffic classification method was applied online without capturing the original data due to the lack of capacity to store the massive amount of data which is the consequence of high traffic speeds. This makes impossible to validate the traffic classification by others.

- **Information theory based classification:** A useful aid in traffic classification is introduced in [18] which is an information theoretic approach and can group the hosts into typical behaviors e.g., servers, attackers. The main idea is to look at the variability or randomness of the set of values that appear in the five-tuple of the flow identifiers, which belong to a particular source or destination IP address, source or destination port. The information theoretic approach can not be used for flow level traffic classification in the same way as the other methods. It is just an aid in traffic classification and arises the problem that it can only specify very broad application types but not capable of classifying specific applications. This method intensively uses the five tuple identification of the flows without other additional information.

  Authors of [18] validated the identified clusters by checking the found dedicated port of the hosts with the port-application database used for port based classification.
- **Combined classification method:** A couple of different approaches have been proposed in the literature for traffic classification, but none of them performs well for all different application traffic types present in the Internet. Thus, a combined method that includes the advantages of different approaches is proposed in [17], in order to provide a high level of classification completeness and accuracy. The classification method in [17] is based on a complex decision mechanism, in order to provide an appropriate identification mode for each different application type. As a consequence, the ratio of the unclassified traffic becomes significantly lower. Further, the reliability of the classification improves due to the joint decision of various methods.

  Authors of [17] validated their method by comparing the results of the introduced method to the results gained from applying all the independent traffic classification mechanisms and their trivial combination on the same traffic traces. The used datasets are full packet length traces measured in several operational mobile broadband networks, but none of them publicly available.

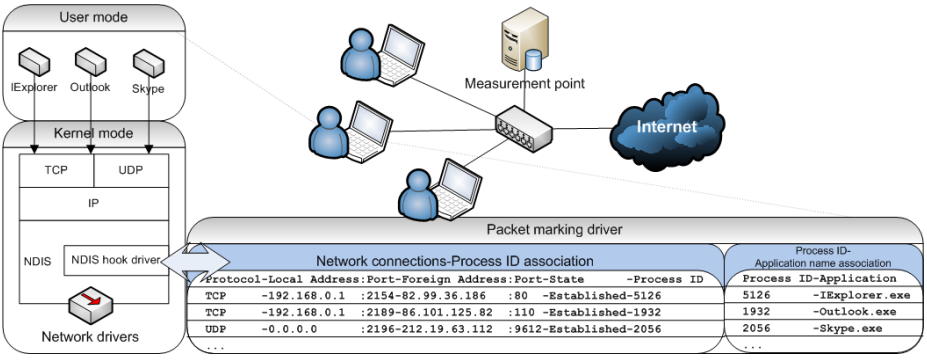

Fig. 1. The position of the proposed driver within the terminal

3 The Proposed Method for Validation

In this section we describe our proposed method for the validation of traffic classification algorithms. As we already mentioned before, instead of validating passive methods by each other we design an active validation method, because we look for a deterministic and reliable solution.

The principle of the method is the following: at the traffic generating terminal, packets are collected into flows and flows are marked with the identifier of the application that generated the packets of the flow. The two main requirements on the realization of the method are that it should not deteriorate the performance of the terminal, and the byte overhead of marking should also be negligible.

The preferred realization is a driver that can be easily installed on terminals. The position of the introduced driver can be seen in Figure 1. It takes place right before the network interface thus each packet exchanged between the terminal and the network has to pass through it. We have implemented a prototype, which is a Windows XP driver based on the Network Driver Interface Specification (NDIS) library. The kernel NDIS library abstracts the network hardware from network drivers and provides an API through which intelligent network drivers can be efficiently programmed. If the sending and receiving functions of the NDIS IP protocol driver are hooked, all TCP and UDP packets can be intercepted and filtered. This method lets developers create for example, firewalls, sniffers, traffic meters or network analyzers based on this technology.

To meet our requirements, the driver is designed to work in the following way. In the case of a passing through packet the following process takes place (see Figure 2):

1. The packet is examined whether it is an incoming or outgoing packet. In case of an incoming packet, the process ends without marking the packet as it is not beneficial to mark incoming packets.
2. In case of an outgoing packet, the size of the packet is examined. If the current packet size is already the size of Maximum Transmission Unit (MTU), the

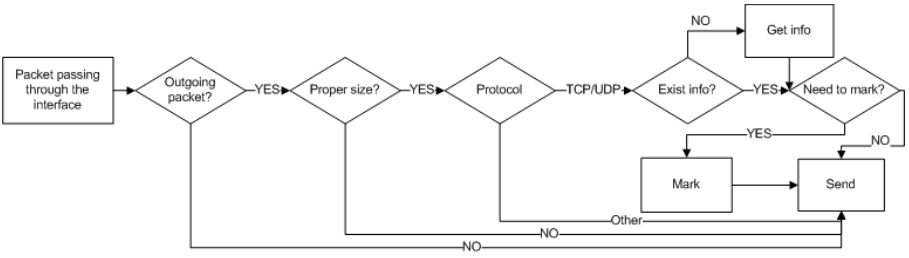

Fig. 2. The working mechanism of the introduced driver

extension of the packet with marking would lead to IP fragmentation. To avoid this, the process continues with only those packets which are smaller than the MTU decreased with the size of marking. Initiating messages in protocols are typically small e.g., the SYN packet of a TCP packet is only a flag, thus there is practically no loss (unmarked flow) with the introduction of this condition.

3. As there is no information in the operating system about those 'network connections' which use other protocols than TCP or UDP, the process continues with only TCP or UDP packets.
4. According to the five tuple identifier of the packet, it is checked whether there is already available information about which application the flow belongs to. The driver has to cache this information because querying the operation system about the existing network connections is very resource consuming and can not be done at high network speeds. We used a hash as the data structure for the cache as it can be directly addressed by the searched data. If there is no information on the flow in the cache yet, the operating system is queried to supply the established network connections and the process IDs of the responsible applications. The process IDs are state specific information in the operating system. To get a universal name about the application, the process IDs are connected to the application's executable name as can be seen in Figure 1.
5. When all information is prepared for the marking of the packet, there is a final chance to decide whether the driver should mark the packet or not. The packet marking can be done for all of the packets in the flow, randomly selected packets of the flow, only the first packet of the flow or it is also possible to switch off the marking for specific applications. There is an option for the random selection of packets to be marked to enforce the first packet of the flow to be marked or avoid the first packet to be marked. The sense of these options is to make an optimal trade-off between performance, network transparency and to ensure high chance of recordable marked packets in the case of network loss.

The marking is done by extending the original IP packet with one option field. We selected the Router Alert option field, because the existence of this

field is transparent for both the routers on the path and also for the receiver host (according to RFC 2113 [3]). If one uses another option field, it should be carefully checked whether the marking is conform to the security policy of the given network, otherwise the marking can be easily removed by an edge router in the border of the access network. In the option field, the first two characters of the corresponding executable file name are added, thus increasing the size of the packet with 4 bytes. The packet size field in the IP header is also increased with 4 bytes and the header checksum is recalculated. As already discussed above, the driver does not mark packets larger than (MTU-4 bytes).

4 The Validation of a State-of-the-Art Traffic Classification Method

A reference measurement [7] has been created as a proof-of-concept of the introduced validation method. For the sake of simplicity, the measurement took place in a separated access network. Our driver has been installed onto all computers on this network. The duration of the measurement was 43 hours. The captured data volume in the network is 6 Gbytes, containing 12 million packets. The measurement contains the traffic of the most popular P2P protocols: BitTorrent, eDonkey, Gnutella, DirectConnect; VoIP and chat applications: Skype, MSN Live; FTP sessions, filetransfer with download manager; e-mail sending, receiving sessions; web based e-mail (e.g., Gmail); SSH sessions; SCP sessions; FPS, MMORPG gaming sessions; streaming radio; streaming video and web based streaming. In Figure 4 the traffic mix of the measurement can be seen. Both the volume and the flow number ratio of different applications is presented.

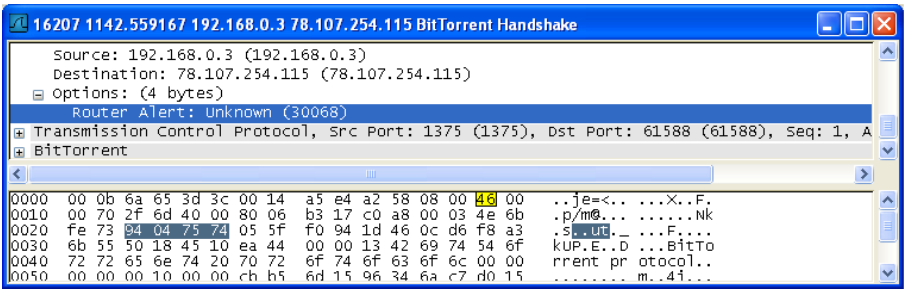

Fig. 3. A marked packet of the BitTorrent protocol

In Figure 3 an example of the marked packets can be seen. The IP header shows the increased size of the packet (without the option field, the value where currently is 46 would be 45) and the option field is highlighted, where the last two fields could be used to place the marking. The marking shows that the generating application was the uTorrent [8] BitTorrent client (by the first two characters in its name).

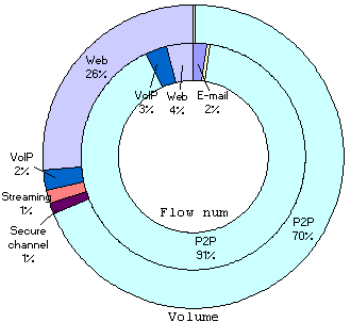

Fig. 4. The traffic mix of the measurement

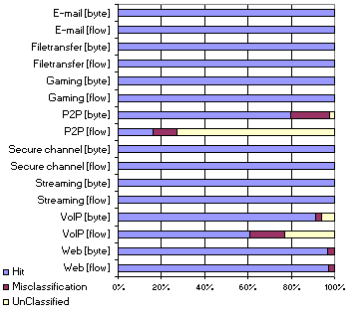

Fig. 5. The results of the classification compared [17] to the reference measurement

The traffic classification method that we wish to validate is described in reference [17], with the addition that the classification of VoIP applications has been extended with ideas from [15] (see the discussion later below). In Figure 5, it can be seen that e-mail, filetransfer, streaming, secure channel, and gaming traffic has been identified very accurately. This is due to the fact that these applications use well-documented protocols, open standards, and do not constantly change. In the case of those protocols which use encryption, the session initiation phase is critical as this phase can be identified the most accurately. In such common protocols as SSH or SCP it can be done with full success, however in such proprietary protocols like Skype the identification fails for several flows.

In the case of classification of P2P applications there are several problems: one thing to note is that P2P applications created plethora of TCP flows containing 1-2 SYN packets probably to disconnected peers. This is the primary reason of the large number of unclassified P2P flows, while the unclassified P2P volume is low. As there is no payload in these packets, the signature based methods can not work. The flows are initiated from dynamically allocated source ports towards not well-known destination ports, thus the port based methods also fail. The server search and P2P communication heuristic [17] methods also fail because there are no other successful flows to such IPs.

Also some small non-P2P flows were misclassified into the P2P class. Fortunately, the number of such flows is small both in flow number and byte volume. We realized that the reason behind is the not fully proper content of the port-application database. Creating too many port-application associations easily results in the rise of the misclassification ratio.

The constant change of P2P protocols also causes some inaccuracy in the classification: there are new features added to P2P clients day-by-day, and their working mechanism can be typical for a selected client not the whole protocol itself.

Another problem of traffic classification is a matter of philosophy. There is traffic which is the derivation of other traffic: the simplest case is the DNS traffic

which is the result of any traffic which uses domain names instead of specific IP addresses. For example, web creates DNS traffic though users do not want to create DNS traffic on purpose. There are more complicated cases: e.g., MSN uses HTTP protocol for transmitting chat messages, which do not need to be considered as web. Furthermore, the MSN client transmits advertisements over HTTP, but this cannot be recognized as deliberate web browsing. This raises the question whether such HTTP flows from the MSN application which are classified as web would have to be considered as misclassification, or it is acceptable that they are classified as web. In this comparison, to be fully objective, only that kind of traffic was considered as hit where the classification outcome and the generating application type (the validation outcome) agreed. For example, the chat on the DirectConnect hubs which has been classified as chat could have been considered as actually correct but in this comparison it was considered as misclassification.

The high VoIP hit ratio is due to the successful identification of both MSN Messenger and Skype. Skype is difficult to identify: for some of the Skype flows the problem is the same as in the case of P2P applications, further Skype is a proprietary protocol designed to ensure secure communication thus it is difficult to obtain a good protocol description. However, authors of [15] found a characteristic feature of Skype: the application sends packets even when there is no ongoing call with an exact 20 sec interval. In [17], there is a P2P identification heuristic which was designed to track any message which has a periodicity in packet sending thus the extension of the original method in [17] for the specific 20 sec periodicity of Skype was straightforward. The validation showed us the deficiency of the classification of Skype, thus with a simple extension of the algorithm it became proper for accurate Skype traffic identification as well. In this way the idea of [17] has been validated as it proved to be robust for the extension with new application recognition, and also the validation mechanism proved to be useful.

5 Summary and Future Work

In this paper we introduced a new active measurement method which can help in the validation of traffic classification methods. The introduced method is a network driver which can mark the outgoing packets from the clients with an application specific marking. With the introduced method we created a measurement and used this to validate the method presented in [17]. The method has been proved to be working accurately but also some deficiencies in the classification of P2P applications and Skype has been identified.

The introduced method can be used in several ways besides the main target of validating traffic classification. One straightforward continuation of this work is to use the marking method at the measurement side for online traffic classification (which we actually did during the debugging of the prototype). This assumes that the terminals accessing an operator's network are all installed with the proposed driver, and also that the driver is made tamper-proof to avoid users

forging the marking. Such an online classification could be used for online clustering of the traffic into QoS classes based on the resource requirements of the generating application. It could also be used by operators to charge on the basis of the used application by the user. The marking could be extended by other information about the traffic generating application, e.g., version number, thus the operator could track the security risks of an old application.

Acknowledgements

We would like to thank the help of Péter Brezina in the development of the introduced driver and the support of his supervisor Sándor Molnár.

References

1. IANA.TCP and UDP port numbers, http://www.iana.org/assignments/port-numbers
2. MSN Messenger, http://join.msn.com/messenger/overview2000
3. RFC 2113, http://www.networksorcery.com/enp/rfc/rfc2113.txt
4. RFC 2246, http://www.ietf.org/rfc/rfc2246.txt
5. RFC 4251, http://www.ietf.org/rfc/rfc4251.txt
6. Skype, http://www.skype.com
7. The measurement created for this article, http://pics.etl.hu/~szabog/measurement.tar
8. uTorrent, http://www.utorrent.com
9. World of Warcraft, http://www.worldofwarcraft.com/index.xml
10. Bernaille, L., Teixeira, R., Akodkenou, I., Soule, A., Salamatian, K.: Traffic classification on the fly, vol. 36, pp. 23–26. ACM Press, New York, USA (2006)
11. Erman, J., Arlitt, M., Mahanti, A.: Traffic classification using clustering algorithms. In: Proc. MineNet 2006, New York, USA (2006)
12. Karagiannis, T., Papagiannaki, K., Faloutsos, M.: BLINC: Multilevel Traffic Classification in the Dark. In: Proc. ACM SIGCOMM, Philadelphia, Pennsylvania, USA (August 2005)
13. McGregor, A., Hall, M., Lorier, P., Brunskill, A.: Flow Clustering Using Machine Learning Techniques. In: Proc. PAM, Antibes Juan-les-Pins, France (April 2004)
14. Moore, A.W., Zuev, D.: Internet Traffic Classification Using Bayesian Analysis Techniques. In: Proc. SIGMETRICS, Banff, Alberta, Canada (June 2005)
15. Perenyi, M., Molnar, S.: Enhanced skype traffic identification. In: Proc. Valuetools 2007 (2007)
16. Sen, S., Wang, J.: Analyzing peer-to-peer traffic across large networks. In: Proc. Second Annual ACM Internet Measurement Workshop (November 2002)
17. Szabó, G., Szabó, I., Orincsay, D.: Accurate traffic classification. In: Proc. IEEE WOWMoM, Helsinki, Finnland (June 2007)
18. Xu, K., Zhang, Z., Bhattacharyya, S.: Profiling Internet Backbone Traffic: Behavior Models and Applications. In: Proc. ACM SIGCOMM, Philadelphia, Pennsylvania, USA (August 2005)
19. Zander, S., Nguyen, T., Armitage, G.: Automated Traffic Classification and Application Identification Using Machine Learning. In: Proc. IEEE LCN, Sydney, Australia (November 2005)