22

ISSN 1470-5559

Approaching Real-time Network Traffic Classification

Wei Li, Kaysar Abdin, Robert Dann and Andrew Moore

Queen Mary
University of London
Department of Computer Science

RR-06-12 October 2006

Approaching Real-time Network Traffic Classification

Wei Li, Kaysar Abdin, Robert Dann and Andrew Moore

Department of Computer Science
Queen Mary, University of London
{wei.li, abm, rjd1, andrew.moore}@dcs.qmul.ac.uk

Abstract. Recent research explored the feasibility of using Machine Learning methods to provide accurate network traffic classification. We further believe that these methods can work on real-time Internet traffic with sufficient accuracy for practical applications. In this paper we present ANTc, a framework for quasi-realtime statistical traffic classification. It essentially demultiplexes network flows, collects statistical features of the flows, and then allows classification of the flows into arbitrary traffic classes using a pre-trained Naïve Bayes model. ANTc contains a built-in feature collector for the input of Naïve Bayes classifier and further provides a modular framework to facilitate further investigations into statistical classification methodologies. It also provides a set of flow sampling parameters which can be tuned, thus is capable of demonstrating the impact on classification accuracy from flow sample size restrictions. Results show that ANTc using Naïve Baye model can work in near real-time without obvious decrease in precision.

1 Introduction

Accurate real-time traffic classification is of fundamental importance to network operations, managements and studies. It serves as the input for Intrusion Detection Systems, provides Class-of-Service (CoS) mapping [1] for Quality of Service (QoS) control, and also provides statistics for network monitoring.

Currently on the internet there are different types of network applications which have diverse statistical characteristics and QoS requirements. Based upon this diversity, statistical classification methods such as Naïve Bayes method and kernel estimator were applied to network traffic classification [2]. With as few as 10 features collected from traffic flows, up to 96% of precision was achieved to classify the traffic into 10 different application classes. It is fundamentally different from traditional traffic classification approaches in that statistical classification does not rely on specific port numbers and protocol signatures but on the statistical behaviour of a traffic flow, such as average segment size, variance of payload size and initial window size, thus avoiding from inspecting traffic payload which may cause privacy concerns and can be rendered ineffective due to the encryption of packet payload. Statistical classification has shown accurate classification results and also promising prospect to be further applied in real-time traffic classification systems, either working standalone or in combination with other methodologies.

In this paper we present ANTc, an online traffic classification framework intended to demonstrate the feasibility of statistical traffic classification operating at near real-time, and further an entry toward accurate real-time classification. ANTc contains a prototype implementation of the above Naïve Bayes methodology that uses the same set of 10 flow features as selected by [2], and also provides a framework that makes it easy to further justify alternative classification methodologies using different features and algorithms later on.

Experimental results show that statistical traffic classification methods can provide high accuracy with acceptable computational and memory overheads. Meanwhile, we believe this prototype of ANTc

could be further developed into a faster and more accurate real-time statistical traffic classification framework for higher speed networks.

The structure of this paper is as follows. The next section reviews some related work. Section 3 presents the architecture design of ANTc. Section 4 discusses classification methodology for real-time. Section 5 presents the experimental results and analysis. Section 6 concludes the paper and outlines future work.

2 Related Work

Existing real-time traffic identification systems in application are intrusion detection systems (IDS) such as *Snort* [3] and *Bro* [4]. These applications mainly use IP and port information in TCP/UDP header and signatures in the packet payload to identify the traffic of an application. Matching the signature strings in packet payload can be very complex and laborious, therefore further IDS solutions turned to utilise hardware technology such as FPGA to allow traffic identification on high speed networks [5].

However, an essential limitation for traditional mechanisms is that they relied on looking for the explicit "symbols" (such as protocol information, port number or signatures) of different applications, but in practice applications does not have one-to-one mapping to such symbols or even may not have such symbols. For example, there are currently many applications using port 80 in order to go through firewalls. More interestingly, POP3 and SMTP have been used for remote access and file sharing [6]. Besides, increasing use of packet payload encryption is troublesome for signature matching. Furthermore, prior knowledge of the application (such as port number and signatures it uses) is always required before these systems can effectively identify an application.

A far more sophisticated content-based traffic classification methodology was provided in [7] which composes of 9 functional blocks and can approach 100% accuracy with all 9 blocks in operation. The major constraints of such a system probably include the system throughput and not being possible to be applied in real-time.

BLINC [8] is another approach based on identifying patterns of host behaviour to classifying traffic flows. This is orthogonal to the methodology used in ANTc that is based on behaviour of traffic flow. In future work we would also hope to combine this method to ANTc by interpreting the host behaviour information into features which can be utilised by ANTc.

A preliminary empirical study into different machine learning algorithms was presented in [9]. It focused on comparing the accuracy and performance of the algorithms, and served as a good guide of entrance to apply various machine learning techniques into network traffic classification. However, the authors may have underestimated the complexity of the problem space.

For real-time statistical traffic classification, [10] proposed to classify online game flows with a small sliding window based on a model trained on multiple short sub-flows. A number of experiments were presented on identification of online game which is a very specific kind of application. Apart from the experiment results, further investigation would be required to validate this methodology.

3 Architecture

A real-time network traffic classification framework based on flow behaviour would theoretically comprise of these following procedures:

- Packet Capture: to capture packets from a network interface.
- Flow Demultiplexing: to collect and aggregate packets in each flow into single flow objects.
- Feature Collection: to collect flow features required for classification from single flow objects.

• Classification: to check these flow features with a pre-trained flow model, in order to predict which application class the flow belongs to.

These procedures above form a complete trace of the data flow, and the design of ANTc naturally followed that to become a layered architecture composing of the same four modules (layers).

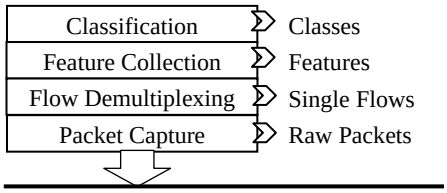

Fig. 1 Layered model of ANTc

Packet Capture and Flow Demultiplexing are bottom layers of the system, which demultiplex the original traffic into single flow objects for classification and are independent of the classification methods. A simple Packet Capture module included in ANTc utilises the libpcap [11] library, and therefore supports both live network traffic and traffic trace (dump) files.

In Flow Demultiplexing layer, currently ANTc only collects packets from the beginning of every TCP flow. Other packets such as UDP and ICMP are ignored in this prototype, as well as those TCP flows where ANTc has not seen the starting of the flow. ANTc concentrates on the beginning of a flow for two reasons: firstly, it is comparably easy to keep track of the state of connection [12] by reading from the beginning of a flow; secondly, the beginning of a flow contains several effective flow features, such as the initial window size. On the other hand, different transport link layer protocols may require totally different Flow Demultiplexing methods, therefore for simplicity we temporarily only support TCP, which composes the majority of Internet traffic [13].

To be operating at near real-time, ANTc should always return the classification result of a flow as soon as a few packets in that flow have arrived. This means it may not collect full TCP flows but an appropriate small number of packets instead. However, with how many packets enough precision can be achieved is still an open problem for research. Hence ANTc allows user to specify the maximum number of packets of a flow sample, and also another maximum flow duration parameter as well, as an alternative to the former.

On top of Flow Demultiplexer layer are Feature Collection and Classification layers which are both related to the choice of classification method. Currently ANTc utilises WEKA [14] framework, which covers a wide range of classification algorithms including the Naïve Bayes method we used, in order to enable handy replacement and justification of classification methodology. In the Feature Collection layer 10 features (as shown in table.1 below) are collected and then stored in WEKA data format as the input of classification. Finally in Classification layer ANTc invokes WEKA to classify the flows based on the pre-trained Naïve Bayes kernel model and provides the output results. We will further discuss the Naïve Bayes methodology we used in the next Section. On the other hand, ANTc also provides a template to collect other flow features, so as to simplify the implementation of other classification methodologies using different features and algorithms later on.

Some of these modules could be combined together, for example, Feature Collection does not necessarily start only after Flow Demultiplexing ends. However, separating them into different modules maintains the modularity with the cost of some computing redundancy. The benefit from modularity is two-fold: it allows network operators to conveniently add this software to their network monitoring suite, and also allows other researchers to easily apply different methodology or packet capture modules. Although the implementation of the Naïve Bayes method with kernel estimator [2] was already built in, it is only a starting point and we recognise there can be many different approaches. We can use ANTc to facilitate further investigations as to justify the classification methodology for real-time network traffic.

The above modules of ANTc are implemented in C while WEKA is Java-based. Therefore JNI [15] is used to invoke WEKA from C.

4 Classification Methodology for Real-time

The real-time implementation of the classification methodology requires modifications and justifications in several aspects. Real-time traffic classification which aims at providing input for IDS or QoS would require early identification of a traffic flow; this indicates that the classification should be finished with limited information from as few packets as possible rather than the whole flow. Reducing number of packets in flow samples may reduce the computing overhead in collecting the features, as shown in Fig.2. Additionally, reducing the number of features collected will also reduce the computing and memory overhead both in collecting features and in classification. However, on the contrary ideally more information (more packets and more features) on a flow would be required in order to provide higher precision.

To solve this contradiction, for a supervised classification system like ours, the challenge is to find a precise classification model for partial flows with limited number of packets. The problem is three dimensional: 1) the model should be built on a concise feature set resulting in limited computing and memory overheads but containing a maximum amount of information in order to correctly classify data samples (network flows) into application classes; 2) for model correctness, the model should be built on sufficient information, i.e. using a sufficient number of correct data samples in each application class to keep it unbiased, 3) the model itself should be of least-possible computational complexity.

Our current model is based on the previous Naïve Bayes model in [2]. ANTc collectes 10 features as listed in Table.1 below to classify network traffic into 10 arbitrary classes. The features we used are identical to those selected by [2] and have been proven to be effective on classifying full flows. The only difference in our implementation is we collect features from limited number of packets at the beginning of a flow rather than full flow samples. Fig.2 shows the relationship between different packet number limits and the computing overhead of these features, which is approximately linear.

As we are focusing on the beginning of flows, we generate our model also using the beginning of flows limited by the same maximum packet number as we use in classification. This is a natural solution based on the simple rule that the training data and testing data should always match up with each other.

Name	Collecting Time	Memory Overhead	Complexity
Push_packets_server	During capture	O (1)	O (1)
initial_window_bytes_server	During capture	O (1)	O (1)
initial_window_bytes_client	During capture	O (1)	O (1)
average_segment_size_server	During capture and after end	O (1)	O (n)
Ip_data_bytes_median_client	After end	O (n)	O (n^2)
actual_data_packets_client	During capture	O (1)	O (n)
data_bytes_variance_server	After end	O (n)	O (n)
minimum_segment_size_client	During capture	O (1)	O (n)
RTT_samples_client	During capture	O (1)	O (n)
push_packets_client	During capture	O (1)	O (1)

Table. 1. List of features used in the Naive Bayes classifier. Details of these features is available in [16]

5 Experimental Results

Our dataset comprises of two non-consecutive days of internet traffic. Day2 is eight months after Day1. These datasets were collected using a high speed monitoring box [17] installed on the Internet connection of the network of *Genome Campus*. The campus is a research-facility with about 1,000

employees and is connected to the Internet via a full-duplex Gigabit Ethernet link. Every packet on each direction of the link was captured along with its full payload. Then the packets were classified by hand into 10 application classes as the base for experimentations, namely WWW, EMAIL, ATTACK, P2P, DATABASE, MULTIMEDIA, SERVICE, INTERACTION, GAMES and BULK. The compositions of the Day1 and Day2 datasets are shown in Table.2.

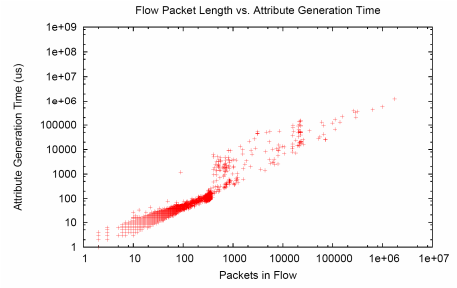

Fig. 2. Feature Collection Time vs. Flow Sample Size

	Total	WWW	MAIL	ATT	P2P	DB	MMED	SERV	INTR	GAME	BULK
Day1	323879	273867	28120	2548	1908	2794	444	1798	86	5	12309
Day2	175651	140868	16483	987	2762	2606	4	1112	36	0	10793

Table. 2. Datasets from two non-consecutive days with an 8 months interval.

In the experiment we tested with five different flow sample length limits, namely 5, 10, 25, 50 and 100 packets. First we collected the Stratified Cross Validation [14] results from Weka for either day's dataset. The total accuracy values of the models are calculated based on these results, as well as the precision and recall values for the three largest application classes, WWW, EMAIL and BULK.

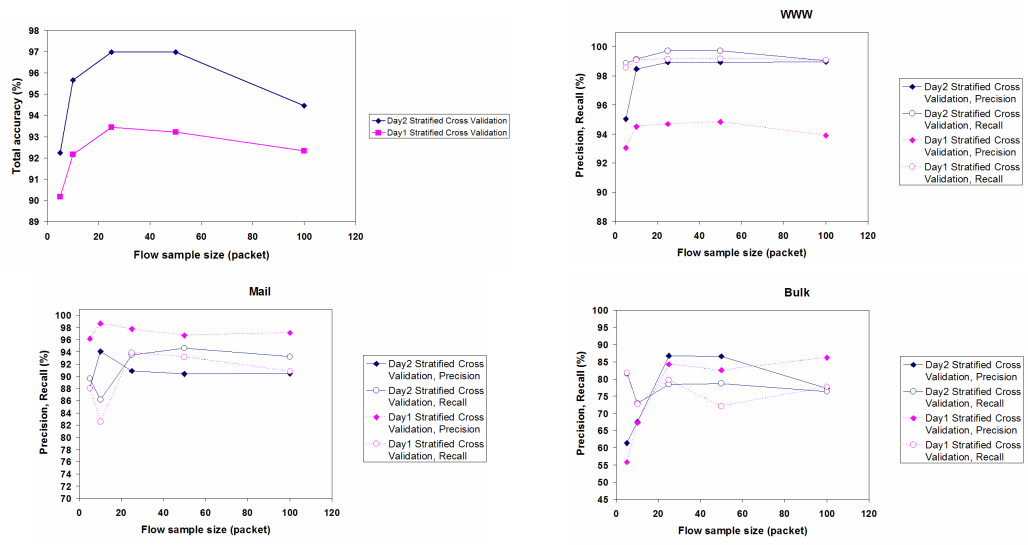

Fig. 3. Stratified Cross Validation of Day1 and Day2 Model

From the results we can find the overall accuracy for Day1 and Day2 using different packet limits. With as few as 5 packets the overall accuracy either dataset can reach 90% and with 25 packets day1 can reach up to 93% while day2 achieves surprisingly 97% accuracy. For comparison study we also tested full flow sampling using the same Naïve Bayes methodology (same features) built on the same datasets. These result in 96.5% and 95.8% accuracy for day1 and day2 respectively.

Therefore, even the feature set we used for real-time beginning-of-flow samples are the same as those had been selected for full flow samples in [2], the results from these features can be regarded as of the same level of accuracy.

Furthermore, in order to validate the temporal stasis of the model, we test using cross validation between the two datasets. Accuracy may decay with time due to emergence of new applications the transformation of Internet traffic. Although some level of decrease in the total accuracy can be seen in our results, the model is still capable of providing useful information on a flow. For example, the recall value of WWW traffic is still very high which means the prediction of a flow to be of WWW can be to some extent reliable. In comparison, sampling full flows result in 94% and 92% accuracy respectively.

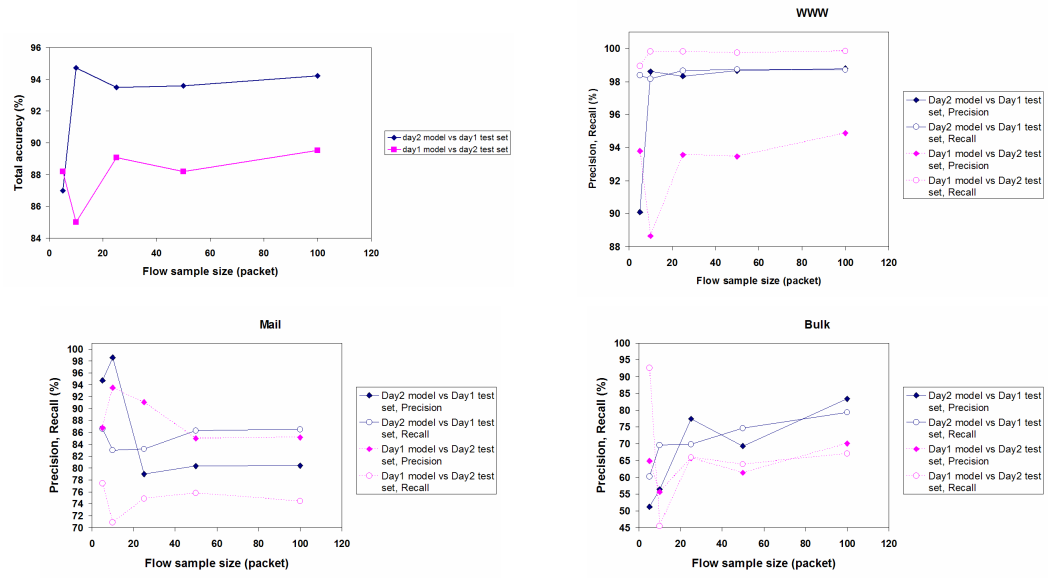

Fig. 4. Cross validation between Day1 and Day2 datasets.

6 Conclusions and Future Work

There are multiple important factors in online traffic classification system, such as: accuracy, completeness, latency and throughput. The challenges in improving the overall performance as well as in balancing between these factors shape a sophisticated problem space. We acknowledge that our experiments in real-time classification research presented in this paper are early attempts which may be coarse-grained and unsophisticated. We also note that there are downsides of current statistical approaches as well. For example, statistical techniques would practically require the information of at least a few packets before it can make a reliable prediction on an unknown flow object. This means the latency in statistical traffic classification is probably higher than traditional header-based and signature-based mechanisms. The way to solve this problem leaves an open area for future work, and is largely depended on the purpose of the real-time traffic classification system. However, a more sophisticated traffic classification combining the strength of different approaches (flow-based, host-behaviour based and header/signature-based) is very promising to counter this problem. This can be a challenge but not a limitation for statistical traffic classification systems to be operating at real-time.

In this paper we presented the architecture of ANTc: a statistical traffic classification framework operating at quasi-realtime. Based on ANTc, we also discussed the real-time implementation of a Naïve Bayesian classification methodology by collecting features at the beginning of flows. Our experimental results show that this classification method can achieve the same level of accuracy as it is used in offline traffic classification.

ANTc is a simple framework but is starting to be a powerful tool in exploring the whole problem space. Allowing collecting different feature sets, using different classifiers and tuning flow sampling parameters, ANTc can easily facilitate further investigations into identifying a best-suitable feature set and a highly effective classification algorithm for real-time classification. Furthermore, ANTc would be equipped with much more built-in contents in the near future, so as to form a complete research platform for traffic classification on high speed networks.

References

1. M Roughan, S Sen, O Spatscheck, N Duffield. Class-of-service mapping for QoS: a statistical signature-based approach to IP traffic classification. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement.* Taormina, Sicily, Italy, 2004.
2. A. W. Moore and D. Zuev. Internet Traffic Classification Using Bayesian Analysis Techniques. In *Proceedings of ACM SIGMETRICS 2005*. Banff, Alberta, Canada.
3. M. Roesch. Snort - Lightweight Intrusion Detection for Networks. In *Proceedings of 13th LISA Conference*, Seattle, Washington, USA. November 7-12, 1999
4. V. Paxson. Bro: A System for Detecting Network Intruders in Real-time. In *Computer Networks (Amsterdam, Netherlands, 1999)*, 31(23-24), 2435-2463.
5. H. Song, J. W. Lockwood. Efficient Packet Classification for Network Intrusion Detection using FPGA. In *Proceedings of International Symposium on Field-Programmable Gate Arrays (FPGA'05)*, Monterey, CA, Feb 20-22, 2005
6. GetByMail, http://www.getbymail.com/en/home/overview.php.
7. A. Moore, K. Papagiannaki. Toward the Accurate Identification of Network Applications. In *Proceedings of Sixth Passive and Active Measurement Workshop (PAM 2005)*, March/April 2005, Boston, MA
8. T. Karagiannis, K. Papagiannaki, M. Faloutsos. BLINC: multilevel traffic classification in the dark. In *Proceedings of ACM SIGCOMM 2005*. Philadelphia, Pennsylvania, USA.
9. N. Williams, S. Zander, G. Armitage "A Preliminary Performance Comparison of Five Machine Learning Algorithms for Practical IP Traffic Flow Classification", (accepted 21 August 2006, to appear) in SIGCOMM Computer Communication Review, October 2006.
10. T.T.T. Nguyen, G. Armitage. Training on multiple sub-flows to optimise the use of Machine Learning classifiers in real-world IP networks. To appear in *IEEE 31st Conference on Local Computer Networks*, Tampa, Florida, USA, November 2006
11. TCPDUMP, http://www.tcpdump.org.
12. J. Postel. Transmission Control Protocol. RFC-793.
13. C. Fraleigh et al. Packet-Level Traffic Measurements from the Sprint IP Backbone. In *IEEE Network*, 2003
14. Waikato Environment for Knowledge Analysis (WEKA), http://www.cs.waikato.ac.nz/ml/weka/.
15. JNI, http://java.sun.com.
16. A. Moore, D. Zuev, M. Crogan. Discriminators for use in flow-based classification. Technical Report, Queen Mary University of London, 2005.
17. A. Moore et al. Architecture of a Network Monitor, In *Proceedings of the Fourth Passive and Active Measurement Workshop (PAM 2003)*, April 2003.