Toward the Accurate Identification of Network Applications

Andrew W. Moore[1] and Konstantina Papagiannaki[2]

[1] University of Cambridge, andrew.moore@cl.cam.ac.uk*
[2] Intel Research, Cambridge, dina.papagiannaki@intel.com

Abstract. Well-known port numbers can no longer be used to reliably identify network applications. There is a variety of new Internet applications that either do not use well-known port numbers or use other protocols, such as HTTP, as wrappers in order to go through firewalls without being blocked. One consequence of this is that a simple inspection of the port numbers used by flows may lead to the inaccurate classification of network traffic. In this work, we look at these inaccuracies in detail. Using a full payload packet trace collected from an Internet site we attempt to identify the types of errors that may result from port-based classification and quantify them for the specific trace under study. To address this question we devise a classification methodology that relies on the full packet payload. We describe the building blocks of this methodology and elaborate on the complications that arise in that context. A classification technique approaching 100% accuracy proves to be a labor-intensive process that needs to test flow-characteristics against multiple classification criteria in order to gain sufficient confidence in the nature of the causal application. Nevertheless, the benefits gained from a content-based classification approach are evident. We are capable of accurately classifying what would be otherwise classified as unknown as well as identifying traffic flows that could otherwise be classified incorrectly. Our work opens up multiple research issues that we intend to address in future work.

1 Introduction

Network traffic monitoring has attracted a lot of interest in the recent past. One of the main operations performed within such a context has to do with the identification of the different applications utilising a network's resources. Such information proves invaluable for network administrators and network designers. Only knowledge about the traffic mix carried by an IP network can allow efficient design and provisioning. Network operators can identify the requirements of different users from the underlying infrastructure and provision appropriately. In addition, they can track the growth of different user populations and design the network to accommodate the diverse needs. Lastly, accurate identification

* Andrew Moore thanks the Intel Corporation for its generous support of his research fellowship

of network applications can shed light on the emerging applications as well as possible mis-use of network resources.

The state of the art in the identification of network applications through traffic monitoring relies on the use of well known ports: an analysis of the headers of packets is used to identify traffic associated with a particular port and thus of a particular application [1–3]. It is well known that such a process is likely to lead to inaccurate estimates of the amount of traffic carried by different applications given that specific protocols, such as HTTP, are frequently used to relay other types of traffic, e.g., the NeoTeris VLAN over HTTP product. In addition, emerging services typically avoid the use of well known ports, e.g., some peer-to-peer applications. This paper describes a method to address the accurate identification of network applications in the presence of packet payload information[3]. We illustrate the benefits of our method by comparing a characterisation of the same period of network traffic using ports-alone and our content-based method.

This comparison allows us to highlight how differences between port and content-based classification may arise. Having established the benefits of the proposed methodology, we proceed to evaluate the requirements of our scheme in terms of complexity and amount of data that needs to be accessed. We demonstrate the trade-offs that need to be addressed between the complexity of the different classification mechanisms employed by our technique and the resulting classification accuracy. The presented methodology is not automated and may require human intervention. Consequently, in future work we intend to study its requirements in terms of a real-time implementation.

The remainder of the paper is structured as follows. In Section 2 we present the data used throughout this work. In Section 3 we describe our content-based classification technique. Its application is shown in Section 4. The obtained results are contrasted against the outcome of a port-based classification scheme. In Section 5 we describe our future work.

2 Collected Data

This work presents an application-level approach to characterising network traffic. We illustrate the benefits of our technique using data collected by the high-performance network monitor described in [5].

The site we examined hosts several Biology-related facilities, collectively referred to as a *Genome Campus*. There are three institutions on-site that employ about 1,000 researchers, administrators and technical staff. This campus is connected to the Internet via a full-duplex Gigabit Ethernet link. It was on this connection to the Internet that our monitor was placed. Traffic was monitored for a full 24 hour, week-day period and for both link directions.

[3] Packet payload for the identification of network applications is also used in [4]. Nonetheless, no specific details are provided by [4] on the implementation of the system thus making comparison infeasible. No further literature was found by the authors regarding that work.

	Total Packets	Total MBytes
Total	573,429,697	268,543
	As percentage of Total	
TCP	94.819	98.596
ICMP	3.588	0.710
UDP	1.516	0.617
OTHER	0.077	0.077

Table 1. Summary of traffic analysed

Brief statistics on the traffic data collected are given in Table 1. Other protocols were observed in the trace, namely IPv6-crypt, PIM, GRE, IGMP, NARP and private encryption, but the largest of them accounted for fewer than one million packets (less than 0.06%) over the 24 hour period and the total of all OTHER protocols was fewer than one and a half million packets. All percentage values given henceforth are from the total of UDP and TCP packets only.

3 Methodology

3.1 Overview of *Content-based* classification

Our content-based classification scheme can be viewed as an iterative procedure whose target is to gain sufficient confidence that a particular traffic stream is caused by a specific application. To achieve such a goal our classification method operates on traffic flows and not packets. Grouping packets into flows allows for more-efficient processing of the collected information as well the acquisition of the necessary context for an appropriate identification of the network application responsible for a flow. Obviously, the first step we need to take is that of aggregating packets into flows according to their 5-tuple. In the case of TCP, additional semantics can also allow for the identification of the start and end time of the flow. The fact that we observe traffic in both directions allows classification of all nearly flows on the link. A traffic monitor on a unidirectional link can identify only those applications that use the monitored link for their datapath.

One outcome of this operation is the identification of unusual or peculiar flows — specifically *simplex* flows. These flows consist of packets exchanged between a particular port/protocol combination in only one direction between two hosts. A common cause of a simplex flow is that packets have been sent to an invalid or non-responsive destination host. The data of the simplex flows were not discarded, they were classified — commonly identified as carrying worm and virus attacks. The identification and removal of simplex flows (each flow consisting of between three and ten packets sent over a 24-hour period) allowed the number of unidentified flows that needed further processing to be significantly reduced.

The second step of our method iteratively tests flow characteristics against different criteria until sufficient certainty has been gained as to the identity of the application. Such a process consists of nine different identification sub-methods. We describe these mechanisms in the next section. Each identification sub-method is followed by the evaluation of the acquired certainty in the candidate application. Currently this is a (labour-intensive) manual process.

3.2 Identification Methods

The nine distinct identification methods applied by our scheme are listed in Table 2. Alongside each method is an example application that we could identify using this method. Each one tests a particular property of the flow attempting to obtain evidence of the identity of the causal application.

Identification Method	Example
I Port-based classification (only)	—
II Packet Header (including I)	*simplex* flows
III Single packet signature	Many worm/virus
IV Single packet protocol	IDENT
V Signature on the first KByte	P2P
VI first KByte Protocol	SMTP
VII Selected flow(s) Protocol	FTP
VIII (All) Flow Protocol	VNC, CVS
IX Host history	Port-scanning

Table 2. Methods of flow identification.

Method **I** classifies flows according to their port numbers. This method represents the state of the art and requires access only to the part in the packet header that contains the port numbers. Method **II** relies on access to the entire packet header for both traffic directions. It is this method that is able to identify simplex flows and significantly limit the number of flows that need to go through the remainder of the classification process. Methods **III** to **VIII** examine whether a flow carries a well-known signature or follows well-known protocol semantics. Such operations are accompanied by higher complexity and may require access to more than a single packet's payload. We have listed the different identification mechanisms in terms of their complexity and the amount of data they require in Figure 1. According to our experience, specific flows may be classified positively from their first packet alone. Nonetheless, other flows may need to be examined in more detail and a positive identification may be feasible once up to 1 KByte of their data has been observed[4]. Flows that have not been

[4] The value of 1 KByte has been experimentally found to be an upper bound for the amount of packet information that needs to be processed for the identification of several applications making use of signatures. In future work, we intend to address

classified at this stage will require inspection of the entire flow payload and we separate such a process into two distinct steps. In the first step (Method **VII**) we perform full-flow analysis for a subset of the flows that perform a control-function. In our case FTP appeared to carry a significant amount of the overall traffic and Method **VII** was applied only to those flows that used the standard FTP control port. The control messages were parsed and further context was obtained that allowed us to classify more flows in the trace. Lastly, if there are still flows to be classified, we analyse them using specific protocol information attributing them to their causal application using Method **VIII**.

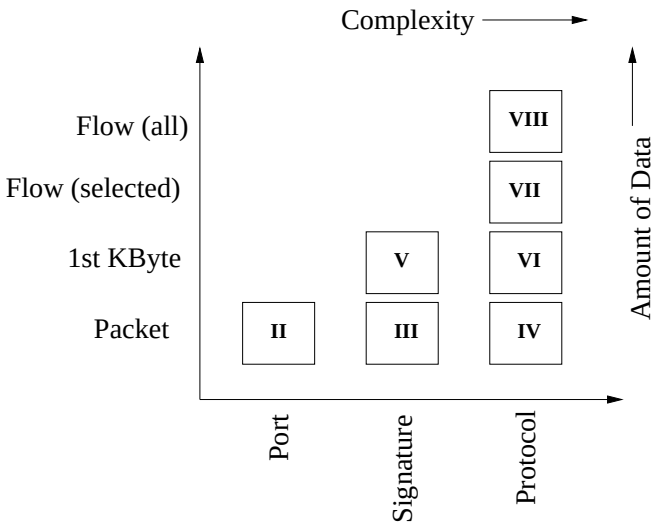

Fig. 1. Requirements of identification methods.

In our classification technique we will apply each identification method in turn and in such a way that the more-complex or more-data-demanding methods (as shown in Figure 1) are used only if no previous signature or protocol method has generated a match. The outcome of this process may be that (i) we have positively identified a flow to belong to a specific application, (ii) a flow appears to agree with more than one application profile, or (iii) no candidate application has been identified. In our current methodology all three cases will trigger manual intervention in order to validate the accuracy of the classification, resolve cases where multiple criteria have generated a match or inspect flows that have not matched any identification criteria. We describe our validation approach in more detail in Section 3.4.

---

the exact question of what is the necessary amount of payload one needs to capture in order to identify different types of applications.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.