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

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

C. Dovrolis (Ed.): PAM 2005, LNCS 3431, pp. 41–54, 2005.

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 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, 2, 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[1]. 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

[1] 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.

Table 1. Summary of traffic analysed

	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

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.

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

Table 2. Methods of flow identification

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

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.

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

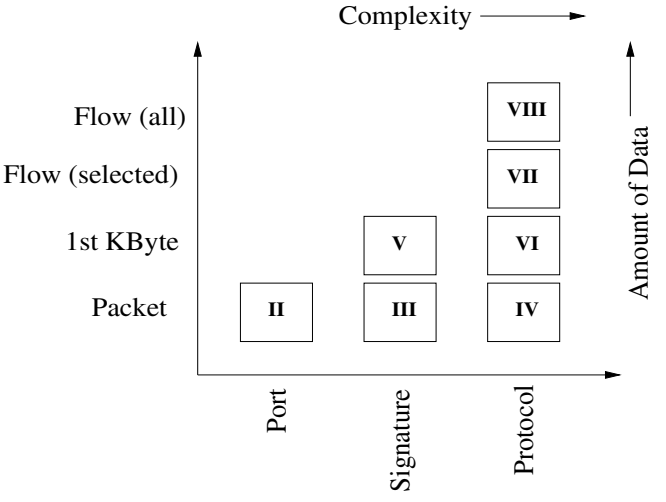

Fig. 1. Requirements of identification methods

once up to 1 KByte of their data has been observed[2]. Flows that have not been 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**.

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 successful identification of specific flows caused by a particular network application reveals important information about the hosts active in our trace.

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

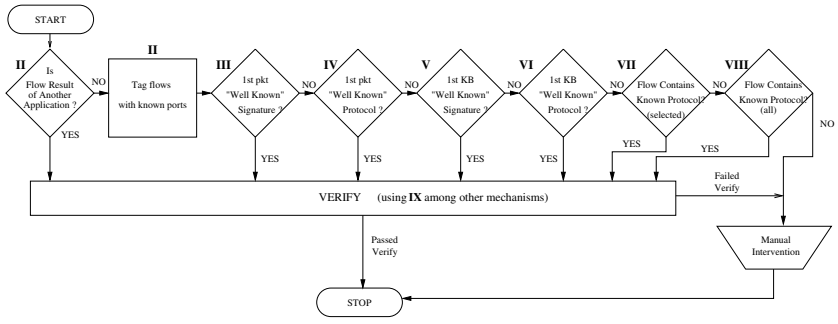

Fig. 2. Classification procedure

Our technique utilises this information to build a knowledge base for particular host/port combinations that can be used to validate future classification by testing conformance with already-observed host roles (Method **IX**). One outcome of this operation is the identification of hosts performing port scanning where a particular destination host is contacted from the same source host on many sequential port numbers. These flows evidently do not belong to a particular application (unless port scanning is part of the applications looked into). For a different set of flows, this process validated the streaming audio from a pool of machines serving a local broadcaster.

Method **IX** can be further enhanced to use information from the host name as recorded in the DNS. While we used this as a process-of-last-resort (DNS names can be notoriously un-representative), DNS names in our trace did reveal the presence of an HTTP proxy, a Mail exchange server and a VPN endpoint operating over a TCP/IP connection.

3.3 Classification Approach

An illustration of the flow through the different identification sub-methods, as employed by our approach, is shown in Figure 2. In the first step we attempt to reduce the number of flows to be further processed by using context obtained through previous iterations. Specific flows in our data can be seen as "child" connections arising from "parent" connections that precede them. One such example is a web browser that initiates multiple connections in order to retrieve parts of a single web page. Having parsed the "parent" connection allows us to immediately identify the "child" connections and classify them to the causal web application.

A second example, that has a predominant effect in our data, is passive FTP. Parsing the "parent" FTP session (Method **VIII**) allows the identification of the subsequent "child" connection that may be established toward a different host at a non-standard port. Testing whether a flow is the result of an already-classified flow at the beginning of the classification process allows for the fast characterisation of a network flow without the need to go through the remainder of the process.

If the flow is not positively identified in the first stage then it goes through several additional classification criteria. The first mechanism examines whether a flow uses a well-known port number. While port-based classification is prone to error, the port number is still a useful input into the classification process because it may convey useful information about the identity of the flow. If no well-known port is used, the classification proceeds through the next stages. However, even in the case when a flow is found to operate on a well-known port, it is tagged as well-known but still forwarded through the remainder of the classification process.

In the next stage we test whether the flow contains a known signature in its first packet. At this point we will be able to identify flows that may be directed to well-known port numbers but carry non-legitimate traffic as in the case of virus or attack traffic. Signature-scanning is a process that sees common use within Intrusion Detection Systems such as *snort* [6]. It has the advantage that a suitable scanner is often optimised for string-matching while still allowing the expression of flexible matching criteria. By scanning for signatures, applications such as web-servers operating on non-standard ports may be identified.

If no known signature has been found in the first packet we check whether the first packet of the flow conveys semantics of a well-known protocol. An example to that effect is IDENT which is a single packet IP protocol. If this test fails we look for well-known signatures in the first KByte of the flow, which may require assembly of multiple individual packets. At this stage we will be able to identify peer-to-peer traffic if it uses well known signatures. Traffic due to SMTP will have been detected from the port-based classification but only the examination of the protocol semantics within the first KByte of the flow will allow for the confident characterisation of the flow. Network protocol analysis tools, such as *ethereal* [7], employ a number of such protocol decoders and may be used to make or validate a protocol identification.

Specific flows will still remain unclassified even at this stage and will require inspection of their entire payload. This operation may be manual or automated for particular protocols. From our experience, focusing on the protocol semantics of FTP led to the identification of a very significant fraction of the overall traffic limiting the unknown traffic to less than 2%. At this point the classification procedure can end. However, if 100% accuracy is to be approached we envision that the last stage of the classification process may involve the manual inspection of all unidentified flows. This stage is rather important since it is likely to reveal new applications. While labour-intensive, the individual examination of the remaining, unidentified, flows caused the creation of a number of new signatures and protocol-templates that were then able to be used for identifying protocols such as PCAnywhere, the sdserver and CVS. This process also served to identify more task-specific systems. An example of this was a host offering protocol-specific database services.

On occasion flows may remain unclassified despite this process; this takes the form of small samples (e.g., 1–2 packets) of data that do not provide enough information to allow any classification process to proceed. These packets used

unrecognised ports and rarely carried any payload. While such *background noise* was not zero in the context of classification for accounting, Quality-of-Service, or resource planning, these amounts could be considered insignificant. The actual amount of data in terms of either packets or bytes that remained unclassified represented less than 0.001% of the total.

3.4 Validation Process

Accurate classification is complicated by the unusual use to which some protocols are put. As noted earlier, the use of one protocol to carry another, such as the use of HTTP to carry peer-to-peer application traffic, will confuse a simple signature-based classification system. Additionally, the use of FTP to carry an HTTP transaction log will similarly confuse signature matching.

Due to these unusual cases the certainty of any classification appears to be a difficult task. Throughout the work presented in this paper validation was performed manually in order to approach 100% accuracy in our results. Our validation approach features several distinct methods.

Each flow is tested against multiple classification criteria. If this procedure leads to several criteria being satisfied simultaneously, manual intervention can allow for the identification of the true causal application. An example is the peer-to-peer situation. Identifying a flow as HTTP does not suggest anything more than that the flow contains HTTP signatures. After applying all classification methods we may conclude that the flow is HTTP alone, or additional signature-matching (e.g. identifying a peer-to-peer application) may indicate that the flow is the result of a peer-to-peer transfer.

If the flow classification results from a well-known protocol, then the validation approach tests the conformance of the flow to the actual protocol. An example of this procedure is the identification of FTP PASV flows. A PASV flow can be valid only if the FTP control-stream overlaps the duration of the PASV flow — such cursory, protocol-based, examination allows an invalid classification to be identified. Alongside this process, flows can be further validated against the perceived function of a host, e.g., an identified router would be valid to relay BGP whereas for a machine identified as (probably) a desktop Windows box behind a NAT, concluding it was transferring BGP is unlikely and this potentially invalid classification requires manual-intervention.

4 Results

Given the large number of identified applications, and for ease of presentation, we group applications into types according to their potential requirements from the network infrastructure. Table 3 indicates ten such classes of traffic. Importantly, the characteristics of the traffic within each category is not necessarily unique. For example, the BULK category which is made up of ftp traffic consists of both ftp control channel: data on both directions, and the ftp data channel which consists of a simplex flow of data for each object transferred.

Table 3. Network traffic allocated to each category

Classification	Example Application
BULK	ftp
DATABASE	postgres, sqlnet, oracle, ingres
INTERACTIVE	ssh, klogin, rlogin, telnet
MAIL	imap, pop2/3, smtp
SERVICES	X11, dns, ident, ldap, ntp
WWW	www
P2P	KaZaA, BitTorrent, GnuTella
MALICIOUS	Internet work and virus attacks
GAMES	Half-Life
MULTIMEDIA	Windows Media Player, Real

In Table 4 we compare the results of simple port-based classification with content-based classification. The technique of port-analysis, against which we compare our approach, is common industry practise (e.g., Cisco *NetFlow* or [1, 2]). UNKNOWN refers to applications which for port-based analysis are not readily identifiable. Notice that under the content-based classification approach we had nearly no UNKNOWN traffic; instead we have 5 new traffic-classes detected. The traffic we were not able to classify corresponds to a small number of flows. A limited number of flows provides a minimal sample of the application behavior and thus cannot allow for the confident identification of the causal application.

Table 4 shows that under the simple port-based classification scheme based upon the IANA port assignments 30% of the carried bytes cannot be attributed

Table 4. Contrasting port-based and Content-based classification

Classification Type	Port-Based		Content-Based	
	Packets	Bytes	Packets	Bytes
	As a percentage of total traffic			
BULK	46.97	45.00	65.06	64.54
DATABASE	0.03	0.03	0.84	0.76
GRID	0.03	0.07	0.00	0.00
INTERACTIVE	1.19	0.43	0.75	0.39
MAIL	3.37	3.62	3.37	3.62
SERVICES	0.07	0.02	0.29	0.28
WWW	19.98	20.40	26.49	27.30
UNKNOWN	28.36	30.43	<0.01	<0.01
MALICIOUS	—	—	1.10	1.17
IRC/CHAT	—	—	0.44	0.05
P2P	—	—	1.27	1.50
GAMES	—	—	0.17	0.18
MULTIMEDIA	—	—	0.22	0.21

to a particular application. Further observation reveals that the BULK traffic is underestimated by approximately 20% while we see a difference of 6% in the WWW traffic. However, the port-based approach does not only underestimate traffic but for some classes, e.g., INTERACTIVE applications, it may over-estimate it. This means that traffic flows can also be misidentified under the port-based technique. Lastly, applications such as peer-to-peer and mal-ware appear to contribute zero traffic in the port-based case. This is due to the port through which such protocols travel not providing a standard identification. Such port-based estimation errors are believed to be significant.

4.1 Examining Under and Over-Estimation

Of the results in Table 4 we will concentrate on only a few example situations. The first and most dominant difference is for BULK — traffic created as a result of FTP. The reason is that port-based classification will not be able to correctly identify a large class of (FTP) traffic transported using the PASV mechanism. Content-based classification is able to identify the causal relationship between the FTP control flow and any resulting data-transport. This means that traffic that was formerly either of unknown origin or incorrectly classified may be ascribed to FTP which is a traffic source that will be consistently underestimated by port-based classification.

A comparison of values for MAIL, a category consisting of the SMTP, IMAP, MAPI and POP protocols, reveals that it is estimated with surprising accuracy in both cases. Both the number of packets and bytes transferred is unchanged between the two classification techniques. We also did not find any other non-MAIL traffic present on MAIL ports. We would assert that the reason MAIL is found exclusively on the commonly defined ports, while no other MAIL transactions are found on other ports, is that MAIL must be exchanged with other sites and other hosts. MAIL relies on common, Internet-wide standards for port and protocol assignment. No single site could arbitrarily change the ports on which MAIL is exchanged without effectively cutting itself off from exchanges with other Internet sites. Therefore, MAIL is a traffic source that, for quantifying traffic exchanged with other sites at least, may be accurately estimated by port-based classification.

Despite the fact that such an effect was not pronounced in the analysed data set, port-based classification can also lead to over-estimation of the amount of traffic carried by a particular application. One reason is that mal-ware or attack traffic may use the well-known ports of a particular service, thus inflating the amount of traffic attributed to that application. In addition, if a particular application uses another application as a relay, then the traffic attributed to the latter will be inflated by the amount of traffic of the former. An example of such a case is peer-to-peer traffic using HTTP to avoid blocking by firewalls, an effect that was not present in our data. In fact, we notice that under the content-based approach we can attribute more traffic to WWW since our data included web servers operating on non-standard ports that could not be detected under the port-based approach.

Table 5. Analysis method compared against percentage of UNKNOWN and correctly identified data

Method									UNKNOWN Data		% Correctly Identified	
I	II	III	IV	V	VI	VII	VIII	IX	Packets	Bytes	Packets	Bytes
•									28.36	30.44	71.03	69.27
•	•							•	27.35	30.33	72.05	69.38
•	•	•						•	27.35	30.32	72.05	69.39
•	•	•	•					•	27.12	30.09	72.29	69.62
•	•	•	•	•				•	25.72	28.43	74.23	71.48
•	•	•	•	•	•			•	19.11	21.07	80.84	78.84
•	•	•	•	•	•	•		•	1.07	1.22	98.94	98.78
•	•	•	•	•	•	•	•	•	<0.01	<0.01	>99.99	>99.99

Clearly this work leads to an obvious question of how we *know* that our content-based method is correct. We would emphasise that it was only through the labour-intensive examining of all data-flows along with numerous exchanges with system administrators and users of the examined site that we were able to arrive at a system of sufficient accuracy. We do not consider that such a laborious process would need to be repeated for the analysis of similar traffic profiles. However, the identification of new types of applications will require a more limited examination of a future, unclassifiable anomaly.

4.2 Overheads of *Content-Based* Analysis

Alongside a presentation of the effectiveness of the content-based method we present the overheads this method incurs. For our study we were able to iterate through traffic multiple times, studying data for many months after its collection. Clearly, such a labour-intensive approach would not be suitable if it were to be used as part of real-time operator feedback.

We emphasise that while performing this work, we built a considerable body of knowledge applicable to future studies. The data collected for one monitor can be reapplied for future collections made at that location. Additionally, while specific host information may quickly become out-of-date, the techniques for identifying applications through signatures and protocol-fitting continue to be applicable. In this way historical data becomes an a-priori that can assist in the decision-making process of the characterisation for each analysis of the future.

Table 5 indicates the relationship between the complexity of analysis and the quantity of data we could positively identify — items are ordered in the table as increasing levels of complexity. The Method column refers to methods listed in Table 2 in Section 3.

Currently our method employs packet-header analysis and host-profile construction for all levels of complexity. Signature matching is easier to implement and perform than protocol matching due to its application of static string matching. Analysis that is based upon a single packet (the first packet) is inherently

less complex than analysis based upon (up to) the first KByte. The first KByte may require reassembly from the payload of multiple packets. Finally, any form of flow-analysis is complicated although this will clearly reduce the overheads of analysis if the number of flows that require parsing is limited.

Table 5 clearly illustrates the accuracy achieved by applying successively-more-complicated characterisation techniques. The correctness of classification reported in Table 5 is computed by comparing the results using that method and the results using the content-based methodology. Importantly, the quantity of UNKNOWN traffic is not simply the difference between total and identified traffic. Traffic quantified as UNKNOWN has no category and does not account for traffic that is mis-classified. It may be considered the residual following each classification attempt.

Table 5 shows that port-based classification is actually capable of correctly classifying 69% of the bytes. Contrasting this value with the known traffic in Table 4 further demonstrates that the mis-identified amount of traffic is rather limited. Nonetheless, 31% of the traffic is unknown. Applying host-specific knowledge is capable of limiting the unknown traffic by less than 1% and signature and application semantics analysis based on the first packet of the flow provides an additional benefit of less than 1%. It's only after we observe up to 1 KByte of the flow that we can increase the correctly-identified traffic from approximately 70% to almost 79%. Application of mechanism **VII** can further increase this percentage to 98%. In Table 2 we have listed example applications that are correctly identified when the particular mechanism is applied.

In summary, we notice that port-based classification can lead to the positive identification of a significant amount of the carried traffic. Nonetheless, it contains errors that can be detected only through the application of a content-based technique. Our analysis shows that typically the greatest benefit of applying such a technique, unfortunately, comes from the most complicated mechanisms. If a site contains a traffic mix biased toward the harder-to-detect applications, then these inaccuracies may have even more adverse consequences.

5 Summary and Future Work

Motivated by the need for more accurate identification techniques for network applications, we presented a framework for traffic characterisation in the presence of packet payload. We laid out the principles for the correct classification of network traffic. Such principles are captured by several individual building blocks that, if applied iteratively, can provide sufficient confidence in the identity of the causal application. Our technique is not automated due to the fact that a particular Internet flow could satisfy more than one classification criterion or it could belong to an emerging application having behaviour that is not yet common knowledge.

We collected a full payload packet traces from an Internet site and compared the results of our content-based scheme against the current state of the art —

the port-based classification technique. We showed that classifying traffic based on the usage of well-known ports leads to a high amount of the overall traffic being unknown and a small amount of traffic being misclassified. We quantified these inaccuracies for the analysed packet trace.

We then presented an analysis of the accuracy-gain as a function of the complexity introduced by the different classification sub-methods. Our results show that simple port-based classification can correctly identify approximately 70% of the overall traffic. Application of increasingly complex mechanisms can approach 100% accuracy with great benefits gained even through the analysis of up to 1 KByte of a traffic flow.

Our work should be viewed as being at an early stage and the avenues for future research are multiple. One of the fundamental questions that need investigation is how such a system could be implemented for real-time operation. We would argue that an adapted version of the architecture described in [5], which currently performs on-line flow analysis as part of its protocol-parsing and feature-compression, would be a suitable system. Such an architecture overcomes the (potential) over-load of a single monitor by employing a method work-load sharing among multiple nodes. This technique incorporates dynamic load-distribution and assumes that a single flow will not overwhelm a single monitoring node. In our experience such a limitation is sufficiently flexible as to not be concerning.

We clearly need to apply our technique to other Internet locations. We need to identify how applicable our techniques are for other mixes of user traffic and when our monitoring is subject to other limitations. Examples of such limitations include having access to only unidirectional traffic or to a sample of the data. Both these situations are common for ISP core networks and for multi-homed sites. We already identify that the first phase of identification and *culling* of simplex flows would not be possible if the only data available corresponded to a single link direction.

We emphasise that application identification from traffic data is not an easy task. Simple signature matching may not prove adequate in cases where multiple classification criteria seem to be satisfied simultaneously. Validation of the candidate application for a traffic flow in an automated fashion is an open issue. Further research needs to be carried out in this direction. Moreover, we envision that as new applications appear in the Internet there will always be cases when manual intervention will be required in order to gain understanding of its nature.

Lastly, in future work we intend to address the issue of how much information needs to be accessible by a traffic classifier for the identification of different network applications. Our study has shown that in certain cases one may need access to the entire flow payload in order to arrive to the correct causal application. Nonetheless, if system limitations dictate an upper bound on the captured information, then the knowledge of the application(s) that will evade identification is essential.

A technical report describing the (manual) process we used is provided in [8].

Acknowledgments

We gratefully acknowledge the assistance of Geoff Gibbs, Tim Granger, and Ian Pratt during the course of this work. We also thank Michael Dales, Jon Crowcroft, Tim Griffin and Ralphe Neill for their feedback.

References

1. Moore, D., Keys, K., Koga, R., Lagache, E., kc Claffy: CoralReef software suite as a tool for system and network administrators. In: Proceedings of the LISA 2001 15th Systems Administration Conference. (2001)
2. Connie Logg and Les Cottrell: Characterization of the Traffic between SLAC and the Internet (2003) http://www.slac.stanford.edu/comp/net/slac-netflow/html/SLAC-netflow.html.
3. Fraleigh, C., Moon, S., Lyles, B., Cotton, C., Khan, M., Moll, D., Rockell, R., Seely, T., Diot, C.: Packet-level traffic measurements from the sprint IP backbone. IEEE Network (2003) 6–16
4. Choi, T., Kim, C., Yoon, S., Park, J., Lee, B., Kim, H., Chung, H., Jeong, T.: Content-aware Internet Application Traffic Measurement and Analysis. In: IEEE/IFIP Network Operations & Management Symposium (NOMS) 2004. (2004)
5. Moore, A., Hall, J., Kreibich, C., Harris, E., Pratt, I.: Architecture of a Network Monitor. In: Passive & Active Measurement Workshop 2003 (PAM2003). (2003)
6. Roesch, M.: Snort - Lightweight Intrusion Detection for Networks. In: USENIX 13th Systems Administration Conference — LISA '99, Seattle, WA (1999)
7. Orebaugh, A., Morris, G., Warnicke, E., Ramirez, G.: Ethereal Packet Sniffing. Syngress Publishing, Rockland, MA (2004)
8. Moore, A.: Discrete content-based classification — a data set. Technical Report, Intel Research, Cambridge (2005)