Classification of P2P and HTTP Using Specific Protocol Characteristics

John Hurley, Emi Garcia-Palacios, and Sakir Sezer

The Institute of Electronics, Communication and Information Technology (ECIT), Queens University of Belfast
jhurley03@qub.ac.uk, e.garcia@ee.qub.ac.uk, s.sezer@ecit.qub.ac.uk

Abstract. A key aspect of traffic classification is the early identification of individual flows which may utilise strategies such as ephemeral ports and transport later encryption to 'hide' on the network. This paper focuses on P2P and HTTP – the two main producers of network traffic – to determine the characteristics of their individual flows. We propose a heuristic based classification system to distinguish HTTP and P2P flows using only the structure of how packets are passed and the lengths of the individual packets. The classification system is then tested on real network traffic and results presented to show it can accurately detect P2P and HTTP within the early part of a TCP flow.

Keywords: Traffic Classification, P2P, HTTP.

1 Introduction

ISPs (Internet Service Providers) and network administrators have seen a huge increase in high bandwidth consuming traffic in recent years [1]. In order for them to deploy effective bandwidth management and quality of service (QoS) policies a traffic classification strategy must be in place.

The original concept of traffic classification was applied on a per packet basis and took into account the port numbers stipulated at the transport layer [2]. Well known port numbers were connected to specific applications (e.g. port 80 for HTTP). However, many applications now use random (or ephemeral) ports to complete data transfer and some even try to avoid detection by masquerading as another application. For example, HTTP traffic mostly travels to port 80, but some P2P protocols can masquerade as HTTP by also using this port [3], thereby confusing a classifier.

To overcome the ineffectiveness of port based classifiers, packets can be considered on a 'per flow' basis rather than individually [4]. A flow is defined as all the packets involved in a single process or task and identified by a 5 tuple value (source and destination IP address, source and destination port number, and transport layer protocol) [4, 5]. On a 'per flow' classification only one packet within the flow requires classifying to enable the identification of the protocol that has generated all packets from the flow.

To identify one packet within a flow, application signatures can be used [6, 7]. These are ASCII stings or a byte series that appear in the transport layer payload of certain signalling packets from specific protocols. However, protocols wishing to avoid detection can apply encryption to their payloads negating the use of signatures. This strategy is deployed on all packets within the mainly Asian used P2P protocol Winny [8], and has also been introduced into the bitTorrent protocol [9]. It is envisaged that, as legal pressure continues to mount on ISPs to combat the transfer of copyrighted material, and as they continue to restrict the available bandwidth for P2P file swapping, more P2P protocols will utilise such classification avoidance techniques.

If port classification cannot be relied on and applications signatures are not available then flow statistics like packet lengths and packet inter-arrival times can be used to predict the underlying protocol of a flow [10, 11]. However, the use of statistics for classification has two main restrictions that must be overcome. Firstly, the statistical strategy must be accurate in its classification of traffic and, secondly, the strategy must reach a classification decision without having to examine too many packets. For example, while the total amount of data passed in a flow can be an indicator of the protocol that generated the traffic, it is of little value to an ISP as all the data would have to pass across the router unclassified before any decision can be made. This introduces the need for fast or early packet classification.

Our research aims to generate fast classifications of the two most common forms on internet traffic - HTTP web traffic and P2P file swapping [17]. We do so by examining the lengths and properties of the initial packets of a flow. We concentrate on TCP flows as the main protocol used by HTTP and P2P to exchange large data quantities (high bandwidth consumers). A series of heuristics are produced to extract 'fingerprints' that are common in either P2P or HTTP flows but not in both. This allows the differentiation between the two even in situations where transport layer encryption is in place, or where P2P flows are trying to masquerade as HTTP by travelling on port 80. Our strategy is applied to real core internet traffic and the results presented.

The rest of this paper is as follows; section 2 gives an overview of related work; section 3 gives our methodology; section 4 introduces the network data used; section 5 outlines our heuristics; section 6 introduces our classification strategy and section 7 provides the classification results.

2 Related Work

BLINC [13] has taken a unique approach to classifying traffic by examining the behaviour of the hosts on the network and connecting it to an application. This behaviour is examined in increasing levels of detail. These levels are defined as the social level (connections at a host), the functional level (hosts role on the network), and the application level (packet transport layer information). This technique can classify around 80 to 90% of all flows with a high level of accuracy. However, this approach utilises application signatures within its application level. Its effectiveness for encrypted traffic when signatures are not available is unknown.

Other classification proposals have used machine learning to test a set of discriminators for their validity in the classification process. [14] and [15] make use of 249 statistical variables in an attempt to find those most suited to traffic classification, while [16] tests machine learning techniques against training set sizes to find the most accurate classifier of P2P traffic. However, these machine learning techniques utilise discriminators based on entire flows (e.g. flow duration) and are only really applicable in data mining processes, not for real time ISP requirements.

[10] introduces a new approach to classification by considering only the first few packets in each bidirectional TCP connection. This focuses on the negotiation phase of the flows meaning that few packets will pass before a classification is determined. It takes the packet lengths of the first 5 data packet of a flow (with server to client packet marked negative) and clusters them in N-dimensional space. These clusters are then used outside their training data to classify new flows.

Our work has a similar concept but expands on the strategy [10] to overcome some of the problems inherent in it. Firstly we consider incoming and outgoing packet lengths in groups rather than individually to lower any errors that will occur through out of order packets (e.g. in retransmissions). Secondly we examine slightly more data within a flow to consider not only the initial handshake phase but to view how data transfer is progressing and to define the structure the flow is taking in an attempt to generate more accurate classifications. However, we are still classifying within the early part of a high bandwidth consuming P2P or HTTP flow. Finally, [10] only examines eDonkey and Kazaa P2P flows while we also aim to investigate and classify a wider range of P2P protocols including the extremely popular bitTorrent and Gnutella networks [17].

3 Methodology

HTTP and P2P flows tend to make use of the TCP transport layer protocol for the transfer of large amounts of data (UDP tends only to be used for signalling [12]). The connection orientated nature of TCP means that the beginning of a flow and the initiating host can easily be determined through observation of the SYN control flag within the TCP header. This is opposite to the connectionless properties of UDP where the initiator or beginning of a flow is much harder to detect. In this work we concentrate on detection of large, high-bandwidth consuming TCP flows.

We examine flow properties in a bidirectional format. This means that the packets sent from a client (flow initiator) to a server and those returned from the server to the client are considered the same flow. A bidirectional flow is recognised in a network trace by its 5 tuple value and by the reverse 5 tuple value. That is *clientIP, clientPort, serverIP, serverPort, TCP* is considered the same flow as *serverIP, serverPort, clientIP, clientPort, TCP*. Properties of these flows are examined in both directions for the maximum packet size, the number of maximum packet sizes, the minimum packet size, the number of minimum packet sizes, throughput (total data passed) , and flow structure (layout of the flow).

Each flow's bidirectional statistics are measured until one of the directions has received a limit of X data packets (those that contain a TCP payload). For experimentation in this paper we set X to 20. Through our flow analysis we determined that this value would allow a depiction of how a flow transfers data after any initial handshakes (first 5 packets in [10]) while still allowing a classification to be made 'early' within the flow. The number of acknowledgement (ACK) packets in the opposite direction is also considered to confirm all packets in both directions are received and have not been lost or chosen a different route across the network. If the number of data packets plus the number of non payload ACK's are equal in both directions then it is reasonable to say that the recorded statistics give an accurate interpretation of the early activity of the TCP flow. This validates the recording of each flow and removes any anomalies that may be present in the data.

4 Data

The research carried out in this paper has utilised real network traffic. This traffic has been taken from a core router situated in the home users ADSL network of a major European ISP in March 2007. There are approximately 20,000 ADSL modems on the network with the data recording covering half of the flows passing over the router at that time period. Different samples of traces from different time periods have been used for heuristic development and for testing.

When carrying out experiments on the core network traffic a background classification (pre-processing) of application signatures was used to determine the protocol of each flow. The application signatures used are developed from [6] and [7] along with further modern signatures not presented in these research papers (e.g. 'ajprot' signalling the use of the appleJuice P2P protocol). Although it is the purpose of this paper to provide an alternative to application signatures, the high accuracy they achieve [6] justifies their use in pre-processing. It is likely that P2P flows existing in our trace data have avoided detection in our pre-processing as they already use encryption. However, we can be confident that those flows that are selected through pre-processing have been accurately identified.

Analysis of flows longer than 20 data packets (our packet limit selection) within our sample traffic show that over 90% of flows are created by P2P and HTTP with the percentage possibly higher as some P2P flows may be unclassified by pre-processing. This figure highlights the importance of differentiation between these two protocol sets. The most popular P2P flows seen are bitTorrent, Gnutella, and eDonkey. We use a first sample trace for protocol analysis and heuristic identification (training trace). A second sample has been recorded from a different time period to ensure fairness [12], and is used to test the heuristics (test trace).

5 Heuristics

To determine the protocol of a flow as P2P or HTTP with a packet limit (20) we use the lengths, throughputs, and structure of the early packets in both directions of a bidirectional TCP flow. The structure of the flow is defined as the shape it has taken once the data packet limit has been received in either the client to server or server to client direction. In other words the number of data packets that have been passed in the opposite direction when one side reaches 20. Figure 1 gives an example of structures by showing how many packets appear in the opposite direction when 20 packets are received in either the incoming or outgoing direction from the flow initiator (client). The results are generated from our training network trace.

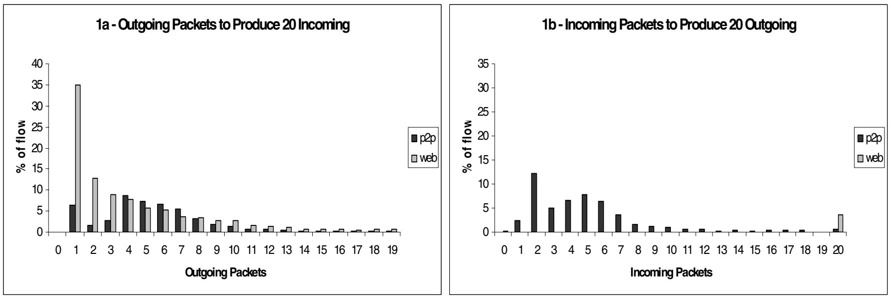

Fig. 1. The structure of P2P and HTTP flows

The most common structure of HTTP traffic shown in figure 1 sees 20 incoming packets signalled by 1 outgoing data packet (figure 1a). However, it can be seen in figure 1b that many P2P flows have structures uncommon with HTTP traffic. These structures are created when the initiator of the flow sends 20 outgoing data packets while only receiving a few incoming packets. This suggests the uploading of data from source to destination – a key aspect in the sharing nature of P2P protocols.

A further example of how packet lengths can be used to differentiate P2P and HTTP is shown in figure 2. This figure describes flows with a structure of 20 incoming packets and at least 2 outgoing. The total of these 20 incoming packets that are of the maximum size witnessed in that direction of the flow are presented.

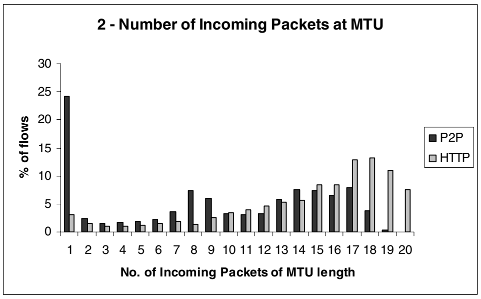

Fig. 2. Number of incoming packets at maximum size

Figure 2 indicates that most of the P2P flows with the associated structure have only 1 incoming packet of maximum length. HTTP flows are more likely to have a high number of Maximum Transmission Unit (MTU) packets with a significant portion of flows having all of their incoming packets at the maximum. HTTP over TCP can stream its data together and pass it out in all, or mostly, MTU packets enabling more efficient transfer of the data. P2P, on the other hand, normally chooses to pass its data in predefined blocks [7]. This means breaking each block into MTU slices and passing the remainder of the block in a packet of its own. This means fewer MTU packets will be viewed in P2P flows.

Furthers heuristics were proposed based on protocol properties and flow observations. All heuristics were tested in an iterative manner with variations made to ‘cut off’ values and structure types to optimise their results. For each heuristic a calculation was made on our training trace showing the percentage of P2P and HTTP flows that the heuristic selects. From these calculations we chose a total of 10 heuristics that we determined to select a high enough percentage of one protocol while selecting few flows from the other protocol.

6 Classification Strategy

From the 10 proposed heuristics a classification strategy was formed. Six of the heuristics detect P2P protocols with four used to define HTTP. The protocol heuristics used to define HTTP are:

Initial incoming packets of same length
Suggests incoming data is being streamed or that the initial handshake information is large. The short handshake lengths that are common in P2P are not taking place.

All incoming packets of MTU
Shows that all data is being streamed at a constant length (figure 2).

Large first 5 packet averages in both directions
Assumes a structure of at least 5 packets in each direction and divides the throughput of the first 5 packets by the MTU for the flow. If both directions produce a result of above 3 then the heuristic is matched. Determines that HTTP request messages are longer than in P2P.

19 out of 20 packets at MTU with the non MTU packet occurring in the first 5
Suggests data is being streamed in an incoming direction but that not all packets are at MTU, possibly due an initial signalling/handshake packet (figure 2).

The characteristics used to define P2P are:

Throughput ratio between outgoing and incoming (> 4 packets in lesser direction)
At least 10 times more bytes in one direction of the flow. Based on the observation that P2P flows tend to receive more data with smaller requests than in HTTP.

Average packet length of non MTU packets < 100 bytes in both directions
P2P tends to have small signalling messages and transfers data in blocks which are divided into MTU packets with any remainder in a small single packet. Many P2P flows consist of all small packets and MTU packets - uncommon in HTTP.

Average of first 5 packet lengths in both directions <100 bytes
P2P flows can require many short signalling messages before passing file data greatly reducing the average of the early packets in both directions - uncommon in HTTP.

20 outgoing packets with between 0 and 5 incoming
P2P flows are commonly used for uploading as well as downloading (figure 1).

Average packet size in the second group of 5 packets <100 bytes in one direction
If one direction of a flow has passed more than 5 data packets and the average size of the next packets is small then these are likely to be short signalling packets (P2P).

Average incoming packet length more than 5 times greater than average outgoing
Similar to the throughput ratio but does not require a structure of 5 packets or more in both directions. Assumes that P2P requests are small compared to responses.

7 Results

To test the validity of this classification scheme the heuristics were applied in order to our test network trace. The results of HTTP and P2P classification are analysed through the use of our pre-processing scheme of application signatures. We define our results with 2 calculations (X represents either P2P or HTTP):

Selected: percentage of the total number of protocol X in the trace data (based on our pre-processing) that are selected by a heuristic defining protocol X

Misclassified: the percentage of total flow selections by heuristics defining protocol X that are shown by pre-processing not to be generated by protocol X – i.e. the number selected as P2P that are actually HTTP and vice versa

Table 1 presents the classification results on our test trace (recorded over a separate time period from training trace).

Table 1. HTTP versus P2P classification results

Protocol	Selected	Misclassified
HTTP	82.39	0.84
P2P	77.43	2.85

The results in table 1 show that over 82% of HTTP and over 77% of P2P traffic flows can be identified with a low misclassification rate (at most 2.85%) using our proposed heuristic set. However, to show how the accuracy of our classifications compares with a similar classification technique we must apply the technique to the same data.

Table 2 shows the results of the k-means clustering strategy proposed in [10] when applied to our network traces. We run the technique by generating clusters from an equal sample of P2P and HTTP traffic in our training trace using the optimum values of the first 5 packet lengths and 50 clusters [10]. The clusters are then used to classify the HTTP and P2P traffic in our test trace. Because a nearest cluster strategy is used [10] to assign flows to a protocol, all flows will be classified as either P2P or HTTP with none remaining ‘unselected’. Therefore table 2 only presents the *misclassification* rating for the strategy proposed in [10].

Table 2. HTTP versus P2P classification from strategy proposed in [10]

Protocol	Misclassified
HTTP	16.79
P2P	16.59

Comparison of our results with those achieved through the clustering strategy of [10] show that our approach is much more reliable by wrongly classifying far fewer flows. In a real-time, online classifier it is important that an ISP of network administrator falsely classifies as few flows as possible.

7.1 Variation of Results with Packet Limits

The results featured for heuristic classification in table 1 require information from the first 20 packets of a TCP flow to make a decision. Figure 3 shows how these results vary if the same (or slightly adapted) heuristics are applied with a smaller packet limit on the same flows.

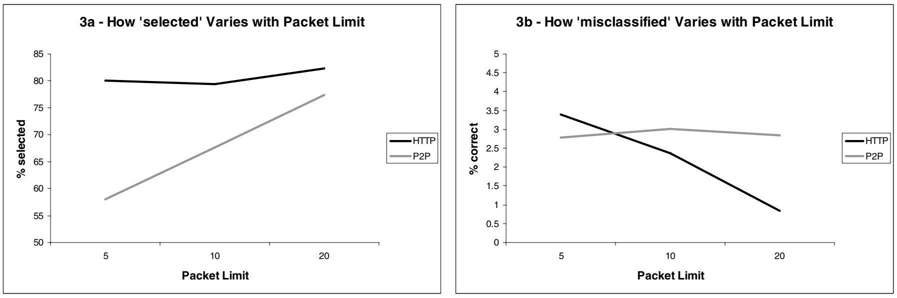

Fig. 3. How results vary when the packet limit for analysis is less than 20

Figure 3a shows how the *selected* calculation varies when the packet limit is changed from 20 data packets in one direction to 5 or 10. The number of total HTTP flows selected is almost constant (varies by approximately 3%) although there is a slight decrease when 10 packets are used as the limit. There is a large change in the number of P2P flows classified with the number selected growing almost 20% as we move from 5 packets to 20 packets. This suggests that the more packets examined within a flow, the more likely that an increasing number of P2P flows will become distinguishable.

Figure 3b signifies how the *misclassified* values vary as the number of packets used in the analysis is increased. The accuracy of the P2P classifications is approximately constant while the accuracy of the HTTP selections is increasing (misclassifications decreasing). Combining the observations from both graphs in figure 3 concludes that the more packets analysed the better the classification of HTTP and P2P. As the number of packets examined is increased the accuracy of P2P will remain constant but the number of total P2P flows selected will increase. In HTTP the number of flows selected will remain constant but the accuracy will increase. Therefore, when a higher number of packets are examined, overall, more flows are selected with a higher accuracy. However, even with this it should be noted that the misclassification rate for both P2P and HTTP never surpasses 3.5% of flows no matter how many packets are measured. This is still a big improvement on the misclassification rate generated by a similar method (table 2).

8 Conclusions and Future Research

In this paper we introduced a new strategy for classifying high bandwidth consuming TCP flows within their early stages meaning ISPs or network administrators can use the classification results for real-time bandwidth management, QoS purposes, and security policies. It was shown that by applying heuristics to analyse the structure and packet lengths of the early transactions of a flow that the two main generators of network traffic, P2P and HTTP, could be classified separately. Our approach produced misclassification rates far lower than a similar strategy [10] when applied to the same test and training data. We also show how these results varied depending on the number of packets examined and suggested that a correlation exists between the number of packets analysed and overall accuracy and detection success of the classification process.

The classification technique presented in this paper has shown accuracy in the distinction of P2P and HTTP. Although these are the main contributors of network traffic other protocols can generate high bandwidth consuming TCP flows. Future work will consider these other flows from applications including FTP, RTSP, email and instant messaging protocols to move the system towards a complete traffic classifier. Future work will also concentrate on the differentiation of the P2P network flows from HTTP through other classification strategies involving social aspects and how hosts behave on the network for situations when flow activity alone is not enough to determine the residing protocol.

References

1. Karagiannis, T., Broido, A., Brownlee, N., Claffy, K.C., Faloutsos, M.: Is P2P dying or just hiding. In: Proceedings of the 47th IEEE Globecom Conference, Dallas, TX (2004)
2. Roughan, M., Sen, S., Spatscheck, O., Duffield, N.: Class-of-Service Mapping for QoS: A Statistical Signature-based. In: 4th ACM SIGCOMM conference on Internet measurement, Taormina, Sicily, Italy, October 25-27 (2004)
3. Kazaa, `http://www.kazaa.com`
4. Gouda, M.G., Liu, A.X.: A model of Stateful Firewalls and its properties. In: Proceedings of the IEEE International Conference on Dependable Systems and Networks (DSN 2005), pp. 320–327 (2005)
5. Ocampe, R., Galis, A., Todd, C., Meer, H.D.: Towards Context-Based Flow Classification. Autonomic and Autonomous Systems. In: ICAS 2006, p. 44 (2006)
6. Sen, S., Spatscheck, O., Wang, D.: Accurate, Scalable In-Network Identification of P2P Traffic Using Application Signatures. In: WWW 2004 (2004)

7. Karagiannis, T., Broido, A., Faloutsos, M., Claffy, K.C.: File-sharing in the Internet: A characterization of P2P traffic in the backbone. Technical report (2004), `http://www.cs.usr.edu/~tkarag`
8. Ohzahata, S., Hagiwara, Y., Terada, M., Kawashima, K.: A traffic identification method and evaluations for a pure P2P application. In: Dovrolis, C. (ed.) PAM 2005. LNCS, vol. 3431, pp. 55–68. Springer, Heidelberg (2005)
9. BitTorrent – A technical description of the bitTorrent protocol, `http://www.cs.chalmers.se/~tsigas/Courses/DCDSeminar/Files/BitTorrent.pdf`
10. Bernaille, L., Teixeira, R., Akodjenou, I., Soule, A., Salamatian, K.: Traffic Classification On The Fly. ACM SIGCOMM Computer Communication Review 36(2) (2006)
11. Bernaille, L., Teixeira, R.: Early recognition of encrypted applications. In: Uhlig, S., Papagiannaki, K., Bonaventure, O. (eds.) PAM 2007. LNCS, vol. 4427, pp. 165–175. Springer, Heidelberg (2007)
12. Karagiannis, T., Broido, A., Faloutsos, M., Claffy, K.C.: Transport Layer Identification of P2P Traffic. In: Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement (IMC 2004), Italy, October 2004, pp. 121–134 (2004)
13. Karagiannis, T., Papagiannaki, K., Faloutsos, M.: BLINC: Multilevel Traffic Classification in the Dark. In: Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications, Philadelphia, Pennsylvania, USA, August 22-26 (2005)
14. Moore, A., Zuev, D.: Internet Traffic Classification Using Bayesian Analysis Techniques. In: ACM SIGMETRICS, Banff, Canada (June 2005)
15. Junior, G.P.S., Maia, J.E.B., Holanda, R., De Sousa, J.N.: P2P Traffic Identification using Cluster Analysis. In: Global Information Infrastructure Symposium. GIIS 2007, pp. 128–133 (July 2007)
16. Raahemi, B., Kouznetsov, A., Hayajneh, A., Rabinovitch, P.: Classification of Peer-to-Peer traffic using incremental neural networks (Fuzzy ARTMAP). In: Electrical and Computer Engineering, CCECE 2008, Canada, pp. 719–724 (2008)
17. Basher, N., Mahanti, A., Mahanti, A., Williamson, C., Arlitt, M.: A Comparative Analysis of Web and Peer-to-Peer Traffic. In: Proceeding of the 17th international conference on World Wide Web, China, pp. 287–296 (2008)