Heuristics to Classify Internet Backbone Traffic based on Connection Patterns

Wolfgang John and Sven Tafvelin

Department of Computer Science and Engieneering
Chalmers University of Technolgy
Göteborg, Sweden
{johnwolf,tafvelin}@chalmers.se

Abstract— **In this paper Internet backbone traffic is classified on transport layer according to network applications. Classification is done by a set of heuristics inspired by two previous articles and refined in order to better reflect a rough and highly aggregated backbone environment. Obvious misclassified flows by the existing two approaches are revealed and updated heuristics are presented, excluding the revealed false positives, but including missed P2P streams. The proposed set of heuristics is intended to provide researchers and network operators with a relatively simple and fast method to get insight into the type of data carried by their links. A complete application classification can be provided even for short 'snapshot' traces, including identification of attack and malicious traffic. The usefulness of the heuristics is finally shown on a large dataset of backbone traffic, where in the best case only 0.2% of the data is left unclassified.** [1]

I. Introduction

Reliable classification of Internet traffic based on network applications is still an open research issue. However, network operators need to know the type of traffic they are carrying, amongst others in order to improve network design and provisioning and to support QoS and security monitoring. Ongoing measurements will furthermore reveal trends and changes in the usage of network applications. A good example is the shift in the early 2000's, when P2P file sharing replaced HTTP as the Internet's 'killer application', implying not only changes in data volumes, but also in traffic properties.

Different approaches to classify network traffic exist. Traditionally, traffic was classified based on *source and destination port numbers*. While this approach is very simple and does not require any packet payload, it is highly unreliable in modern networks. This is especially true for most P2P applications, which are trying to disguise their traffic in order to evade traffic filters and legal implications. It was shown that pure port number analysis underestimates actual P2P traffic volumes by factors of 2 to 3 [1].

A more reliable technique involves analysis of *packet payloads*. This approach can potentially provide highly accurate results given a complete set of payload signatures [2]. Beside the high effort of keeping the set of signatures updated, this method relies on network traces including packet data, which is uncommon due to privacy and legal concerns. Furthermore matching payload signatures on high-speed links is far from trivial and poses high processing requirements.

A more recent classification technique is based on *statistical properties* of flows. A promising feature of these methods is that they are neither relying on port numbers nor on packet payload. However, the success of such 'statistical fingerprints' highly depends on the accuracy of the training data used. Ensuring accuracy and authenticity of the training sets is still an open issue [3], especially for disguised P2P flows.

Finally, network data can be classified according to *connection patterns*. Instead of looking at individual packets or flows, sequences of flows to or from a specific endpoint are matched with a set of predefined heuristics [4], [5]. These heuristics typically don't require packet payload and could potentially even disregard port numbers.

We initially intended to classify Internet backbone data in order to investigate the influence of P2P applications on traffic properties. Consequently it was planned to apply an existing and verified classification technique. Since our available datasets did not include packet payload and accurate training data, payload signatures or statistical fingerprinting could not be applied. Thus applying straight-forward connection pattern heuristics was the obvious approach. In [4], Karagiannis presents a set of two heuristics for transport layer identification of P2P traffic, including seven rules for removing false positives. The paper verifies that their method can identify 95% of P2P flows, with around 10% false positives compared to a carefully carried out payload analysis on OC-48 backbone data. Additionally, Perenyi [5] recently proposed an updated set of six heuristics to identify and analyze P2P traffic, based on very similar ideas like Karagiannis. These heuristics were verified against traffic generated in a lab environment, yielding a hit ratio for P2P traffic of over 99%, with less than 1% false positives or unclassified P2P flows.

After applying the approaches of both Karagiannis and Perenyi to our data, it turned out that their results differ substantially. Furthermore, obvious false positives were detected in our data with both classification methods. As a result, we propose a refined combination of the heuristics by Karagiannis and Perenyi including some additions. The modifications were necessary to make the classification suitable for relatively short traces of a harsh Internet backbone environment, including highly aggregated and diverse traffic with a substantial amount of attacking and malicious traffic. Besides being based on the verified heuristics of Karagiannis and Perenyi, the results

[1] This work was supported by SUNET, the Swedish University Network

where further verified by manual inspection. Flows, which are not classified as P2P traffic by all three applied sets of heuristics are separately discussed regarding their most probable traffic class, thereby identifying obvious misclassification.

II. Data Description

Our dataset was collected during 20 days in April 2006 on the OC192 backbone of the Swedish University Network (SUNET). During this period, four traces of 20 minutes were collected each day at identical times (2AM, 10AM, 2PM, 8PM), as described in [6] and [7]. After recording the packet level traces on the 2x10 Gbit/s links, payload beyond transport layer was removed and IP addresses were anonymized due to privacy concerns. A per-flow analysis was conducted on the resulting bidirectional traces, where flows are defined by the 5-tuple of source and destination IP and port numbers as well as transport protocol. TCP flows represent connections, and are therefore further separated by SYN, FIN and RST packets. UDP flows are separated by a timeout of 64 seconds. The 73 traces in the dataset sum up to 10.7 billion packets, containing 7.5 TB of data. We identified 81 Million TCP connections and 91 Million UDP flows, with the TCP connections carrying 97% of all data. The further analysis is dealing with TCP connections only, even though the classification heuristics have been successfully applied to UDP flows as well.

III. Proposed Heuristics

The set of heuristics proposed in this paper is strongly inspired by the heuristics by Karagiannis [4] and Perenyi [5], and will therefore be presented briefly only. The classification is based on connection patterns, but in some cases also port numbers are taken into account. Besides the rules for filtering out P2P traffic (H1-H5), a number of heuristics are used to remove false positives from flows suspected to be P2P traffic (F1-F10). These 'false positive' rules in turn can be used to classify other types of traffic, as shown in section V. In contrast to Perenyi's approach, most of our proposed heuristics (with exception of H5 and F10) are first applied independently to all flows and are then prioritized. We apply these heuristics to our dataset in 10 minute intervals, which means that every interval is analyzed self-contained, without memory of previous intervals. Even though such memory could improve the accuracy of the results, our approach has the advantage to allow operators to classify snapshots of their traffic fast and in an ad hoc fashion. We will show that even 10 minute intervals can provide satisfying results. The proposed heuristics include a number of thresholds which might be adjusted. For our data the thresholds used were derived empirically through experiments on a number of traces. In the following list of heuristics, *(K)* (Karagiannis) or *(P)* (Perenyi) indicate by which previous method the heuristic was inspired, while *(J)* (John) marks newly introduced rules.

H1: TCP/UDP IP Pairs:(K),(P). This rule exploits the fact that many P2P applications use TCP for data transfer and UDP for signaling traffic. Source and destination IP pairs, which concurrently use TCP and UDP are therefore marked as P2P hosts. All flows to and from these hosts are marked as potential P2P flows. Concurrent here means usage of TCP and UDP within the 10 minutes interval. Karagiannis identified some non-P2P applications which show a similar behavior, such as netbios, dns, ntp and irc (Table 3 in [4]). UDP flows from these applications are excluded from this heuristic based on their port-numbers.

H2: P2P Ports:(P). Even though many P2P applications choose arbitrary ports for their communication, approx. one third of all P2P traffic can still be identified by known P2P destination port numbers [1]. Furthermore, it seems disadvantageous for non-P2P applications to deliberately use well known P2P ports for their services, since traffic on these ports is often blocked by traffic filters in some networks. Flows to and from port numbers listed in Table 3 of [5], enriched with additional P2P ports, are marked as potential P2P traffic.

H3: Port Usage:(P). In normal application, the operating system assigns ephemeral port numbers to source ports when initiating connections. These numbers are often iterating through a configured ephemeral port space. It is very unusual, that the same port numbers are used within short time periods. This however can be the case for P2P applications with fixed ports assigned for signaling traffic or data transfer. If a source port on a host is repeatedly used within 60 seconds, the host is marked as P2P host, and all flows to and from this host are marked as potential P2P flows.

H4: P2P IP/Port Pairs:(K). If listening ports on peers in P2P networks are not well known in advance, they are typically propagated to other peers by some kind of signaling traffic (e.g. an overlay network). This means that each host connecting to such a peer will connect to this agreed port number, using a random, ephemeral source port. As noted by Karagiannis, P2P peers usually maintain only one connection to other peers, which means that each endpoint (IP,port) has at least the same number of distinct IP addresses (#sIP) and number of distinct ports (#sPort) connected to it. If #sPort-#sIP<2 and #sIP>5, the host is considered as P2P host, and all flows to and from this host are marked as potential P2P.

F1: Web IP/Port Pairs:(K). Web traffic on the other hand typically uses multiple connections to one server. For this reason hosts are marked as web-hosts, if the difference between #sPort and #sIP connected to an endpoint (IP,port) is larger than 10, the ratio between #sPort and #sIP is larger than two and at least 10 different IPs are connected to this endpoint (#sPort-#sIP>10 and #sPort/#sIP>2 and #sIP>10). All flows with http port numbers (80, 443, 8080) to and from these webhosts are then marked as web traffic.

F2: Web:(P). To further identify web traffic, we follow Perenyi's heuristic number 2, taking advantage of the fact that web clients typically not only use multiple, but even parallel connections to webservers. Hosts with parallel connections to a http port are considered as webservers. All flows to and from web servers on http ports are marked as web traffic.

F3: DNS:(K). Traditional services like dns sometimes use equal source port and destination port numbers. As suggested by Kargiannis, we mark endpoints (IP,port) as non-P2P, if it includes flows with equal source- and destination port and port numbers smaller than 501. All flows to and from this endpoint are then marked as non-P2P traffic.

F4: Mail:(K). Hosts receiving traffic on mail ports (smtp, pop, imap) and in the same analysis interval also initiate connections to port 25 on other hosts are considered to be mailservers. All flows to and from mailservers are marked as mail traffic.

F5: Messenger:(K). Popular messenger and chat servers (icq, yahoo, msn, jabber, irc) tend to have long uptimes and rarely change IP addresses, especially when maintained by commercial providers such as Microsoft and Yahoo. To improve the accuracy of the results, in this heuristic we therefore take advantage of the whole 20 day long dataset. Hosts, connected to by at least 10 different IPs on well known messenger ports within a period of at least 10 days, are marked as messenger servers. All traffic to and from these hosts on known messenger ports is classified as messenger traffic.

F6: Gaming:(J). Popular game servers (currently only the most common online games Half-Life and World of Warcraft) are identified in the same fashion as messenger servers. All traffic to and from the game servers on well known gaming ports is classified as gaming traffic.

F7: Ftp: (J). Ftp was not taken into account by Karagiannis, while Perenyi implicitly included it as part of its 'well known port' rule. Identifying data transfer in passive ftp remains a problem. Active ftp data transfer on the other hand can easily be marked as ftp traffic identified by an initiating sourceport number of 20, as used by ftp servers to actively serve their requesting clients.

F8: non P2P Ports:(P). As noted by Perenyi, destination ports are still suitable to identify traffic of some common applications. Our set of well known non-P2P ports includes netbios, dns, telnet, ssh, ftp, mail, rtp and bgp. All flows to the listed destination ports are marked as non-P2P flows.

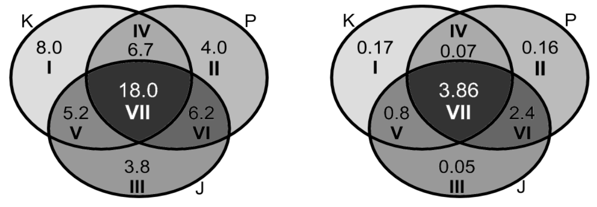
F9: Attacks:(J). This rule is probably the most significant improvement to the original heuristics. While Perenyi does not take malicious traffic into account at all, Karagiannis rules out simple network scans as false positives. We first identify suspicious pairs of source IPs and destination Ports (*AttackPairs*). All flows with source IP and destination port inside the list of AttackPairs are then marked as attacks. AttackPairs are identified by three different cases:
a) *Sweep*: The ratio between number of destination IPs (#dIP) and number of destination ports (#dPort) from a certain host is greater than 30. This means that one host is connecting to a lot of hosts with only a few different port numbers, as typically the case when scanning IP ranges for vulnerabilities on specific ports.
b) *Scan*: The ratio between #dIP and #dPort is less than 0.33 and #dIP is less than 5. This would be the case if one host is scanning a small number of specific, dedicated targets on a large number of different ports.
c) *DoS*: #dIP is less than 5, #dPort is less than 5 and the average number of conn. per sec (conn/s) is greater than 6. This behavior represents 'hammering' attacks, where one host is trying to overload a few targets (typically one) by opening connections to a few services very frequently.

F10: unclassified, known non-P2P Port:(J). Up to this point all heuristics mark flows independent of each other. All flows left unmarked until now are neither suspected to be P2P traffic nor obvious cases of non-P2P traffic. We believe it is safe now to apply a port number classification on the previously unclassified flows. All flows, whose source- or destination port number matches a set of well-known non-P2P port numbers including (http, messenger, game) are classified non-P2P, if not classified by any heuristics (H1-H4, F1-F9).

H5: unclassified, long flow:(P) After removing well known applications from the unclassified flows, we mark remaining unclassified flows which carry more than 1 MB of data in one direction or have connection durations of over 10 minutes as P2P flows. This rule is based on Perenyis heuristic 6, even though we believe it is a very weak rule. However, there is a large probability, that such long flows in fact are P2P flows.

After running an analysis on our dataset based on the presented heuristics, we classify all flows as P2P traffic which have been classified by one or more of the heuristics H1-H5, and at the same time NOT being classified by any of the false positive heuristics F1-F10. In Section IV, flows marked by H5 are included to P2P traffic. However, in Section V we chose to treat traffic classified by this heuristic separately.

Weaknesses: The above suggested mixture of connection pattern and port number classification has some weaknesses. First of all, the analysis interval can greatly influence the success of the heuristics, especially for those analyzing connection patterns. Longer intervals yield better results given that the various empirical thresholds are adjusted. A natural border for the analysis interval is obviously given by memory and computational constraints. Additionally, there is a risk with too long intervals since activities on the Internet are often short lived, and e.g. a host doing a scanning campaign on port 80 might simply surf the Internet an hour later. Another problem in this context are networks behind NATs or with dynamically assigned IP addresses. A second weakness is the length of the traces used. For connections established before the measurement interval the initiator is unknown, and it is unclear which host is source and which is destination. Additionally there is typically some asymmetrically routed traffic in backbone networks, which needs to be considered as special case when implementing the heuristics. Furthermore, heuristics based on connection patterns are depending on a certain amount of connections per host during the analysis interval. Finally, heuristics relying on empirical thresholds are not fail-proof, and it is possible to come up with examples for false positives for any of them. However, both Karagiannis and Perenyi proved that these heuristics can be effective when carefully prioritizing the different rules.



(a) #connections (10^6) (b) Amount of data (TB)

Fig. 1. P2P traffic by Karagiannis (K), Perenyi (P) and new proposal (J)

IV. Verification of the proposed Heuristics

To verify the proposed adjustments, we classified our backbone data by each of the three sets of heuristics (Karagiannis, Perenyi and our own proposal in section III). For each flow, a bitmask was set in a database according to matching rules. This method allowed us to analyze intersections between the three approaches separately - meaning flows marked as P2P traffic by either one, two or all three of the approaches. The results are illustrated by the Venn diagrams in fig.1, presenting connection counts (a) and amount of data (b) in absolute numbers. The three circles represent P2P flows classified by the different rule-sets (Karagiannis left, Perenyi right, new proposal beneath). The following paragraphs will discuss the different intersections (IS I-VII), thereby motivating the proposed modifications and additions to the original approaches.

IS I: This intersection represents flows classified as P2P by Karagiannis only. A number of updated rules identified these flows as false positives. Rule F9 (attacks) marked 53% of them, often classified as known non-P2P ports by Perenyi. This is plausible, considering that these connections are mainly 1-packet flows, directed to popular scanning ports (135, 139, 445). Rule F2 (web) classified another 25% of these connections, carrying 40% of the data in this intersection. Since parallel connections to http ports are a strong indication for web traffic, F2 is regarded as a reliable rule. F8 (non P2P-ports) accounts for 15% of these connections, carrying 43% of the data, mainly on ports for rtp, ssh and mail. This is plausible, since it is common that these applications carry large amounts of data, so there is no reason considering them as P2P flows. The remaining flows are either marked by F7 (active ftp) or F10 (unclassified, but known non-P2P port).

IS II: In this intersection, 99% of the data was classified as P2P by Perenyi's 'long flow' rule only. This is obviously Perenyi's weakest heuristic, since it simply considers any flow carrying more than 1 MB of data or lasting longer than 10 minutes as P2P. 75% of this data is considered as false positive according to F10. Unclassified by any other heuristic, a pure port number classification marks these flows as web flows according to their destination http ports. Another 10% are marked as web traffic by F2. The remaining data was classified by F4 (mail), F5 (messenger) and F6 (gaming), all three considered to be accurate rules, taking connection patterns and port number into account. In terms of connection numbers, 95% of the connections in IS II are again identified as false positives by F9 (attacks) with similar properties as in IS I.

IS III: All of the flows only classified as P2P by the proposed heuristics are unclassified by Perenyi. Even Karagiannis left 45% unclassified, with the remaining 45% classified by the non-P2P IP/Port Pair rule. In [4] this rule was identified as unreliable if less than 5 IPs are connected to an IP/Port Pair. Since in H4 this restriction was taken into account, it is plausible to include the flows marked as P2P in IS III based on combinations of H4 and/or H3 (port usage).

IS IV: The flows classified as P2P by both Karagiannis and Perenyi are in 98% of the cases again marked as false positives by F9 (attacks), carrying very little data. In terms of data, Perenyi's 'long flow' rule and Karagiannis' IP/Port Pair rule are responsible for 90% of the data in this intersection. As discussed above, both rules are considered rather weak. Since additionally none of the refined P2P heuristics (H1-H4) matched, rule F10 (unclassified, but well known port) is reason enough to exclude 80% of this flows as false positives (mainly targeting http ports). The remaining flows have been marked by F1 (web pairs), F5 (messenger) and F6 (gaming).

IS V: In this intersection, flows are entirely unclassified by Perenyi. Since these flows are classified as P2P by both Karagiannis and the proposed heuristics, there is no reason not to consider them as P2P traffic.

IS VI: Perenyi's 'long flow' rule identified 77% of the data in this large intersection as P2P, with the remaining connections classified according to known P2P port numbers. The proposed heuristics on the other hand classify 88% of these flows as P2P by H2-H4, accounting for 72% of data. Most of the data is even classified by 2 or 3 of the heuristics. The remainder (685 GB) is classified by H5 (long flows) only, and will therefore be treated as a special category in our results section. Karagiannis leaves a large part (60%) of this intersection unclassified, with the rest classified by the non-P2P IP/Port Pair rule, which is an inaccurate rule for endpoints with few connected hosts as noted above. Since there is no strong indication to rule out flows as false positives, they are classified as P2P except for the 685 GB by H5 (long flows).

IS VII: Data in this intersection is classified as P2P by both Karagiannis and Perenyi, and no false positives were identified by the proposed heuristics. Consequently, there is no reason not to consider this intersection as P2P.

V. Classification Results

We finally applied the proposed heuristics to our data traces (Section II). Fig.2 represents time series of classified network protocols. The x-axis of the graphs represents time, with one bar for each trace time (2AM, 10AM, 2PM and 8PM). Four traces on three days (07/04, 09/04, 23/04) had to be discarded due to measurement errors. The remaining whitespaces between bars represent the 8 hour measurement break between 2AM and 10AM, which means that each continuous block represents 4 traces collected in the order of [10AM, 2PM, 8PM, 2AM]. The first graph shows total amount

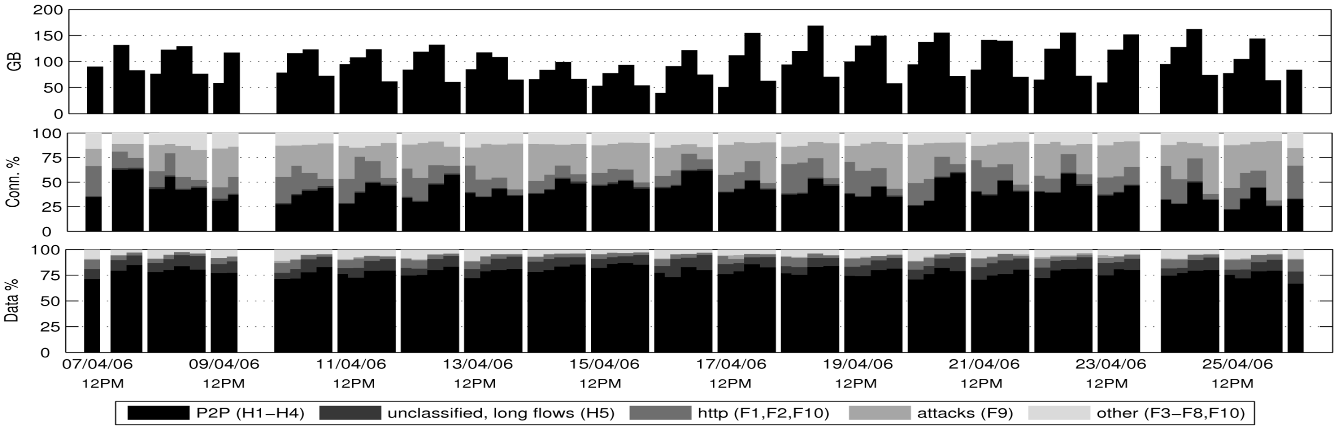


Fig. 2. TCP data vs trace times (first row); Appl. breakdown by #conn. (second row); Appl. breakdown by data carried (third row)

of TCP data in GByte versus trace times. The second and third row illustrate application breakdown for the particular trace in terms of connection numbers and data volumes.

In the connection breakdown, only four categories are visible, since flows classified by H5 are too small in number to show up in this graph. Anyhow, these 31,000 long flows are responsible for almost 10% of the TCP data. Typically, these flows begin and end outside the measurement period and transfer data between hosts, which do not generate additional traffic on our links. Since our classification method is based on connection patterns, insufficient connection numbers for a particular host reveal a weakness of this method. In the data breakdown on the other hand, flows classified by F9 (attacks) are not visible. Even though attacks represent between 8 and 60% of the flows, they carry less than 1% of the data on average. This also proves the power of F9, since it effectively detects DoS attacks and network scanning, which typically show up as short 1-packet flows only, carrying no payload data. P2P flows (flows matching H1-H4, while not matching any of the false positive rules F1-F10) account for an average of 42% of the connections. On the other hand, they carry between 66 and 87% of the traffic, with an average of 79%. This indicates once more the success of the heuristics, since P2P flows are expected to carry more data on average than non-P2P flows. On this dataset, the proposed heuristics left as little as 1% of the connections and 0.2% of the data unclassified (except the flows classified by H5).

While a careful analysis of these results need to be done as future work, the short result section should indicate the power and usefulness of the proposed heuristics.

VI. Summary and Conclusions

This article proposes a set of heuristics for classifying backbone-type data according to applications. The proposed heuristics are intended to provide researchers and network operators with a comparably simple[2] method to get insight into the type of data carried by their links. Furthermore these heuristics work on traces as short as 10 minutes, which allows operators to classify snapshots of their traffic relatively fast, by only adjusting applied thresholds and parameters empirically. The heuristics can be used to classify backbone traffic according to a number of applications, including P2P traffic, web traffic and other common applications. Furthermore, we introduce a new rule that successfully identifies network attacks, which is an additional feature for network operators and researchers interested in network security or intrusion detection issues. Some of the proposed heuristics are based on two existing methods. Besides relying on the verification methods of these original heuristics, a careful analysis of the resulting classifications was carried out, pinpointing obvious cases of false positives. Both previous sets of heuristics overestimate the number of P2P flows, mainly because attacking traffic is not taken into account accordingly. On the other hand, both methods underestimate the amount of P2P data on the links. By combining the successful rules of the two methods and adding new, necessary rules, a set of refined and updated heuristics is presented. The heuristics are successfully applied to a large collection of backbone data, yielding a valuable breakdown of applied application protocols. When considering the few large flows classified by the H5 rule as P2P traffic, the proposed heuristics leave only 0.2% of the data unclassified.

[2] Simple, because it does not require packet payloads, updated payload signatures or training data for statistical fingerprinting methods.

References

[1] A. W. Moore and K. Papagiannaki, *Toward the Accurate Identification of Network Applications*, ser. Lecture Notes in Computer Science, 2005.

[2] S. Sen, O. Spatscheck, and D. Wang, "Accurate, scalable in-network identification of p2p traffic using application signatures," ser. 13th International World Wide Web Conference, New York, USA, 2004.

[3] M. Crotti, M. Dusi, F. Gringoli, and L. Salgarelli, "Traffic classification through simple statistical fingerprinting," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 1, pp. 5–16, 2007.

[4] T. Karagiannis, A. Broido, M. Faloutsos, and K. Claffy, "Transport layer identification of p2p traffic," in *Proceedings of the 4th ACM Conference on Internet Measurement*, Taormina, Sicily, Italy, 2004.

[5] M. Perenyi, D. Trang Dinh, A. Gefferth, and S. Molnar, "Identification and analysis of peer-to-peer traffic," *Journal of Communications*, vol. 1, no. 7, pp. 36–46, 2006.

[6] W. John and S. Tafvelin, "Analysis of internet backbone traffic and header anomalies observed," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, San Diego, California, USA, 2007.

[7] W. John and S. Tafvelin, "SUNET OC 192 Traces, April 2006 (collection)," http://imdc.datcat.org/collection/1-04HN-W=SUNET+OC+192+Traces%2C+April+2006 (accessed 071207).