Characterizing the network behavior of P2P traffic

Raffaele Bolla, Marco Canini, Riccardo Rapuzzi, Michele Sciuto

DIST - Department of Communication, Computer and System Sciences, University of Genoa
Via Opera Pia 13, 16145 Genova, ITALY
{raffaele.bolla, marco.canini, riccardo.rapuzzi, michele.sciuto}@unige.it

***Abstract*— Nowadays the majority of Internet traffic is generated by peer-to-peer (P2P) file sharing applications. As the popularity of these applications has been increasing dramatically over the past few years, it becomes increasingly important to analyze their behavior and to understand their effects on the network. The ability to quantify their impact on the network is fundamental to a number of network operations, including traffic engineering, capacity planning, quality of service, forecasting for long-term provisioning, etc.**

We present here a measurement study on the characteristics of the traffic associated with two different P2P applications. Our aim is to provide useful insight into the nature of P2P traffic from the point of view of the network. To achieve this, we introduce a novel meauserement, Content Transfer Index (CTI), to distinguish two classes of behavior associated with P2P traffic: the download and the signaling traffic profile. Next we apply the CTI to our data sets and show that it effectively offers a general characterization of P2P traffic. Finally, we present a number of statistical measurements that are significantly unbiased due to having considered the distinction between the two classes. To the best of our knowledge, this is the first study to follow this approach.

We believe such a study will help researchers better understand the impact of P2P applications on the network and how to improve their performance.

I. Introduction

Nowadays peer-to-peer (P2P) file sharing applications constitute a major share of the total traffic in the Internet [1]. P2P traffic is believed to be hazardous for networks, not only because of its high traffic volume, but also the transfer of large files. As the popularity of these applications has been increasing dramatically over the past few years, it becomes important to analyze their behavior and understand their effects on the network. In particular, quantifying their impact on the network is important to a broad range of network operations, including traffic engineering, capacity planning, quality of service, forecasting for long-term provisioning, etc.

Recent works ([2], [3] and [4]) have shown that accurate identification of P2P traffic is challenging because P2P applications, particularling the newer generations are incorporating various strategies to avoid detection. The ability to identify P2P traffic is fundamental to quantify the impact of P2P applications. However, it only represents a first step towards fully understanding their behavior and effects on the network.

We present here a measurement study on the characteristics of the traffic associated with two different P2P applications. Our aim is to offer useful insight into the nature of P2P traffic as it is seen from the point of view of the network. To achieve this, we introduce a novel measurement, Content Transfer Index (CTI), that distinguishes two classes of behavior for the P2P traffic: the download and the signaling traffic profile. We applied the CTI to our data sets and we show that it effectively offers a general characterization of P2P traffic by presenting a number of statistical measurements.

Our results show that the download traffic is, as expected, the majority of the total traffic volume. However, because of the large number of signaling communications, simple statistical measurements applied to the entire traffic aggregate are biased as they fail to capture the real behavior of P2P traffic.

Because of the large differences between these two types of traffic, we argue that a comprehensive P2P traffic characterization should include this distinction.

As a preliminary validation, we compared the CTI's outcome, specifically the eDonkey downloads, against the ground truth built using the methodology presented in [5], resulting in an accuracy of above 95%.

The remainder of this paper is organized as follows. Section II describes how we identified P2P traffic in our data sets. Section III gives a brief overview of the eDonkey and BitTorrent P2P networks. Section IV presents our characterization of P2P traffic, and the results of its application to a number of measurements are shown in Section V. Finally, Section VI concludes the paper.

II. Data collection

We analyzed traces that were collected using an optimized[1] Linux-based open router [6] turned into a monitoring box. The monitor was located at two different links of the University's campus network. For both traces, we captured every packet seen on each direction of the links along with its full payload and we removed the link layer header (ethernet).

To capture the first trace (DEPT), the monitor was located on the link connecting our department to the campus network. The second trace (GENUA) was captured by monitoring the main connection to the Internet. Being a data set that spans over two weeks, the DEPT trace is our reference trace, whereas GENUA is used to confirm our findings. We post-processed the trace in order to exclude the TCP connections for which we do not observe the canonical set up (triple handshake).

Table I lists general workload dimensions of our data sets: counts of distinct source and destination IP addresses, and the number of flows, packets, and bytes observed.

[1]The optimizations include using the Linux NAPI's polling mode and tuning the network card's RX ring buffer and the OS's socket buffers.

TABLE I

GENERAL WORKLOAD DIMENSIONS OF OUR TRACES.

Set	Dur.	Src. IP	Dst. IP	Flows	Packets	Bytes
DEPT	449h	2.8M	5.9M	46M	1241M	738GB
GENUA	1h	214K	253K	976K	20M	10.5GB

In this study we define flows as unidirectional, while we use the term *conversation* to denote bidirectional traffic, i.e., a conversation is composed by two flows: traffic from A to B, and traffic from B to A. Each flow is always identified by two end points consisting of {IP, port} pairs and the transport level protocol. For a protocol like UDP, which is not connection oriented, we use a timeout of 60s to determine the end of a conversation.

We used two open source tools, namely *l7-filter* [7] and *ipp2p* [8], to classify the P2P traffic in our traces. Both these tools identify P2P flows via pattern matching, i.e., searching the payload content of the packets for known protocol signatures. These classifiers act every time a packet is received, and mark a conversation as classified as soon as they find a known pattern in one direction. For scalability reasons, only up to the first N packets of each conversation are tested, where N is a user configurable parameter. These tools differ in the way the pattern matching is realized: *l7-filter* reassembles the packet payloads into a buffer (there is one buffer for each direction), stripping the null bytes, and uses regular expressions to search the buffer for strings containing a match to a known protocol signature; *ipp2p* searches each individual packet for known patterns of the most common P2P protocols.

Many of the signatures used by *l7-filter* and *ipp2p* are obtained from protocol specifications. However, because this is not generally possible for proprietary protocols, they are in some cases derived from reverse engineering the protocols, like what has been done by the authors of [2].

Because these tools are not available as off-line trace processing tools (they are originally meant to be deployed as filters in the Linux's iptables firewall for traffic shaping purposes), we ported their source code to the Click modular router [9], which turned out to be a viable analysis framework.

We validated our versions of the tools against the original tools by comparing the results obtained from the classification of the GENUA data set. The outputs were indeed the same.

We exploited both these tools to accurately identify P2P traffic in our data sets. As done in [3], we limited to 10 the number of packets per flow searched for signatures.

By running the tools on our traces, we found that the differences in their classification results were negligible, therefore we only used *ipp2p*.

Finally, we compared the classification results obtained with our *ipp2p* classifier with the output of the payload classifier used to validate BLINC [4], obtaining very close results on our data sets.

Table II presents the volumes of P2P traffic in our traces, divided by P2P application.

In the remainder of the paper we only focus on the traffic generated by eDonkey and BitTorrent, since the majority of P2P traffic in our campus network is associated with these two applications[2].

TABLE II

BREAKDOWN BY PROTOCOL OF P2P TRAFFIC VOLUME IN OUR TRACES.

P2P Protocol	GENUA	DEPT
BitTorrent	9.74%	26.49%
eDonkey	72.33%	73.55%
Gnutella	0.78%	0.04%
KaZaA	0.01%	0.00%
DirectConnect	17.14%	0.00%
WinMX	0.00%	0.02%

III. EDONKEY AND BITTORRENT OVERVIEW

In this section we briefly present the main features of eDonkey and BitTorrent.

eDonkey: The eDonkey network belongs to the class of hybrid P2P architecture: it is composed of peers and multiple servers. The servers provide a file search service and maintain a list of addresses of other servers, to be distributed to peers. Each peer logs on to one of the servers (using a TCP connection) and registers its shared files with it. To search a file, a peer sends the query to its main server which replies with a list of matching files and their location. Optionally, the peer can send further queries directly to other servers via UDP. To download a file, a peer establishes direct TCP connections to the peers that are sharing the requested file. During download, files are split into separate pieces. Pieces of the same file can be obtained from several different peers. Finally, a file can be shared by a peer before it is completely downloaded.

BitTorrent: BitTorrent is a file distribution system based on the P2P paradigm. Unlike other popular P2P networks, such as eDonkey or Gnutella, which comes with a file search service, the sole objective of BitTorrent is to quickly replicate a single large file to a set of clients. There is a separate *torrent* for each file that is distributed. A torrent consists of a central component, called a *tracker* and all the currently active peers. The role of the tracker is to act as a rendez-vous point for the peers of the torrent, however it is not involved in the actual distribution of the files. Once a peer joins a torrent, it first contacts the tracker to retrieve a list of active peers. It then cooperates with 20-40 peers chosen at random to replicate the file among each other. Although there are unofficial extensions to support UDP communications, by default, BitTorrent only uses TCP.

IV. P2P TRAFFIC CHARACTERIZATION

P2P traffic can be roughly divided into download traffic and signaling traffic: the first is caused by the transfer of content,

[2]P2P traffic is believed to be hazardous for networks, and our campus network makes no exception. We are aware that a filtering system has been deployed, realizing traffic shaping for the most common P2P applications.

the latter is mainly due to the presence of an overlay network, and possibly a search service.

Because of the large differences between these two types of traffic, we argue that a comprehensive P2P traffic characterization should include this distinction. In fact, even though the download traffic is generally the major share of the total P2P traffic volume, if such distinction is not taken into account, then, because of the large number of signaling conversations, simple statistical measurements are biased.

A way to accurately differentiate download vs. signaling traffic would be to implement a protocol analyzer. Although one can leverage existing tools, e.g. *binpac* [10], to build protocol analyzers, this solution has several drawbacks: (i) it requires specific knowledge of P2P protocols, (ii) it needs access to the payload of each packet, (iii) it has to maintain a state for each conversation.

In particular, we are interested in characterizing the P2P traffic from the point of view of the network, i.e., to gain more insight on the distinctive characteristics of the behavior of aggregates of download and signaling P2P conversations, including the volumes of carried content, the conversations' interarrival times and durations.

We point out that it is not our intention to provide a method that deterministically divide P2P conversations into the two categories. Thus, we follow a novel approach that doesn't rely on the accuracy of the solution based on protocol analyzers, but provides a means for clearly distinguishing two different classes of behaviors and for treating P2P traffic with generality.

We call such classes of behaviors the download and the signaling traffic profiles. To some extents we're abusing this terminology, as it is possible, though not common, that a download conversation exhibits the typical characteristics of signaling traffic and vice-versa. For example, early truncated downloads are not clearly distinguishable from signaling conversations. However, we accept such misclassifications because misclassified flows don't bias our measurements and we want to keep our method simple.

Our approach consists of a way to offer a statistical characterization of P2P traffic through the formalization of a measurement index.

We now define the Content Transfer Index (CTI) of a conversation C as:

$$\frac{F}{f+F} \cdot \frac{\bar{P}}{\mathrm{MSS}(C)} + \frac{f}{f+F} \cdot \frac{\bar{p}}{\mathrm{MSS}(C)} \in [0,1] \ ,$$

where F, f are the lengths of the two flows constituting C, such that $f \leq F$. One can use three different flow features to represent its length: the packet count, the count of payload bytes and the count of headers and payload bytes. However, in this paper, we only present the results obtained by using the count of payload bytes as the flow length. $\bar{P}$ and $\bar{p}$ represent the average number of payload bytes per packet calculated for the flow with length F and f respectively. The maximum segment size (MSS) of the conversation C is expressed with $\mathrm{MSS}(C)$. For the UDP, we assume that the MSS corresponds to the maximum transfer unit (MTU) minus the IP and UDP headers' lengths.

Hence, given the pair of end points $\{$A, B$\}$, the CTI gives us a measure of the way the content is transferred. At the opposite ends of the spectrum there are two distinct traffic profiles:

- when the conversation is flatter or balanced (i.e., A and B exchange an even quantity of content, mainly using packets whose payload size is far from the MSS), then the CTI's value tends to zero;
- when the conversation is richer of content which is either transferred from a single end point that dominates the conversation (unbalanced), or efficiently exchanged between the end points, then the CTI's value tends to one.

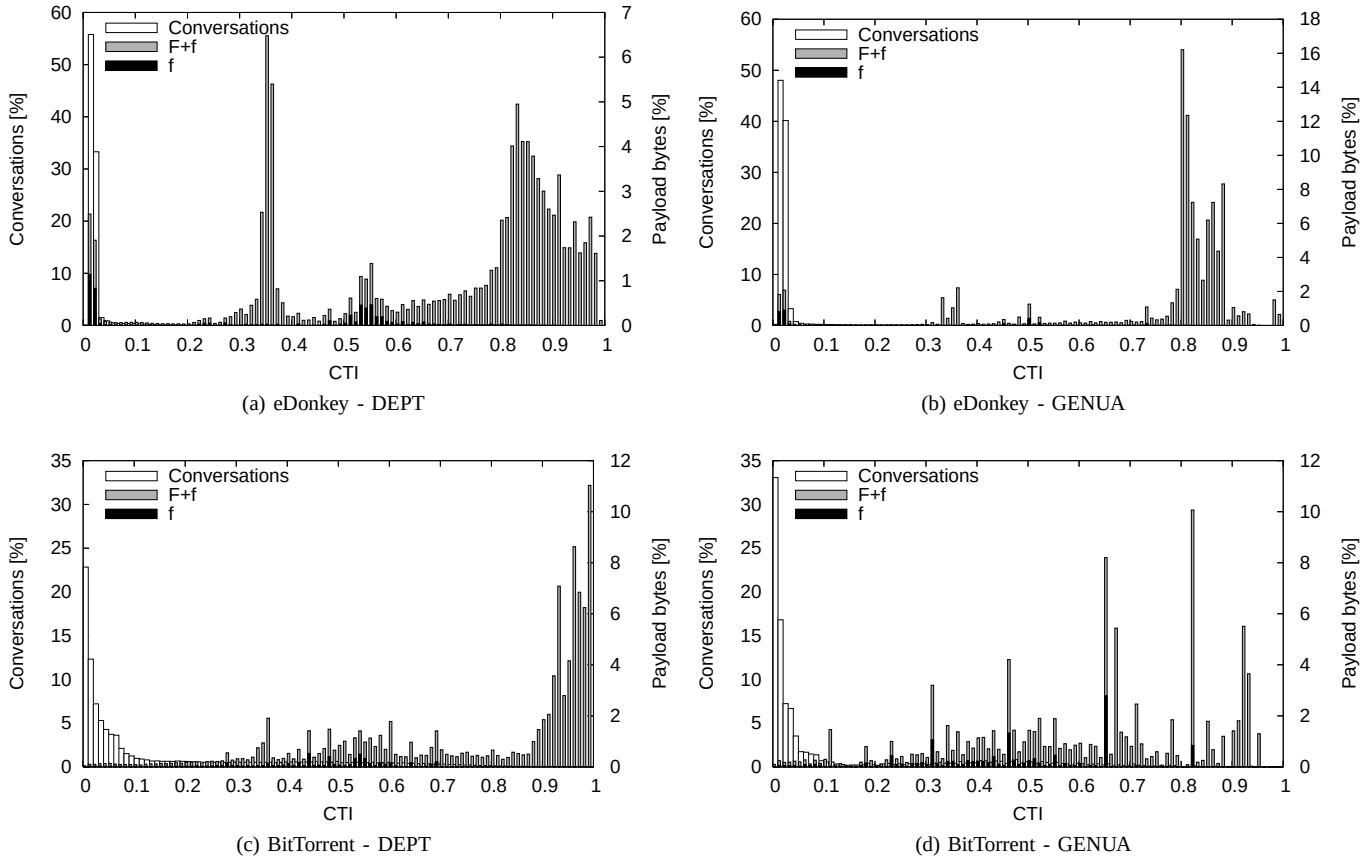
The intuition behind the proposed metric derives from observing that the traffic profiles of download and signaling conversations are quite different.

The idea is that, during a file download, a peer, on average, gets packets filled up to the MTU and sends back fewer packets to acknowledge the received data. Even when the peers are exchanging pieces of a single file with one another (as realized in BitTorrent), causing balanced conversations, still, the average payload size tends to reach the MSS. On the contrary, the signaling conversations are characterized by a flatter profile, consisting of a more even count of exchanged bytes and packets.

Throughout the rest of the paper, we divide P2P traffic into signaling vs. download by using the value of the conversation's CTI. In particular we use a threshold to distinguish the two traffic types: a conversation with a CTI's value above the threshold is marked as download, while a value below the threshold determines a signaling conversation.

To validate our metric, we classify eDonkey conversations in our DEPT data set into download and non-download. A conversation is marked to belong to the download category if the conversation contains at least one of the eDonkey protocol opcodes '`OP_SENDINGPART`' or '`OP_COMPRESSEDPART`'. We computed the accuracy as the number of correctly classified download conversations over the total count of conversations. We found that using 0.2 as the CTI threshold, we correctly classify 95% of the download conversations (i.e., 95% of the download conversations have CTI greater than 0.2). The value 0.2 appears to the common break point to the graphs in Figure 1 and we choose to use it in the rest of the paper.

The limit of this validation is that we are unable to accurately identify signaling conversations from the ones classified as non-download, because some download conversations might end up being in the non-download category. However, we obtain that only the 2.5% of non-download conversations have CTI's value above the threshold and that there is a difference of one order of magnitude between the average packet size, volume and duration of the non-download conversations above the CTI threshold and those below it.



(a) eDonkey - DEPT

(b) eDonkey - GENUA

(c) BitTorrent - DEPT

(d) BitTorrent - GENUA

Fig. 1. Relationships between the conversations, payload bytes and the CTI of (a) eDonkey conversations in DEPT, (b) eDonkey conversations in GENUA, (c) BitTorrent conversations in DEPT and (d) BitTorrent conversations in GENUA. The histograms are plotted with CTI bin size 0.01.

V. P2P TRAFFIC ANALYZES

A. CTI graphs

Figure 1a, 1b, 1c and 1d show the relationships between the conversations, payload bytes and the CTI of DEPT's eDonkey, GENUA's eDonkey, DEPT's BitTorrent and GENUA's BitTorrent TCP conversations respectively. Each figure shows three overlapping histograms, symbolizing the following figures corresponding to the same CTI range: the number of conversations, the summation of the minimum and maximum length flows of the conversations (denoted with F + f) and the summation of just the minimum length flows (denoted with f). All the graphs clearly show two distinctive profiles: the signaling profile containing most of the conversations is having CTI values below 0.2, whereas the download profile dominated by the payload bytes above 0.2. Also note in the signaling profiles that the conversations are quite balanced (f is almost F).

B. Interarrival times

Table III lists the average, standard deviation and maximum conversation interarrival times.

TABLE III
AVERAGE, STANDARD DEVIATION AND MAXIMUM CONVERSATION INTERARRIVAL TIMES [S] IN DEPT.

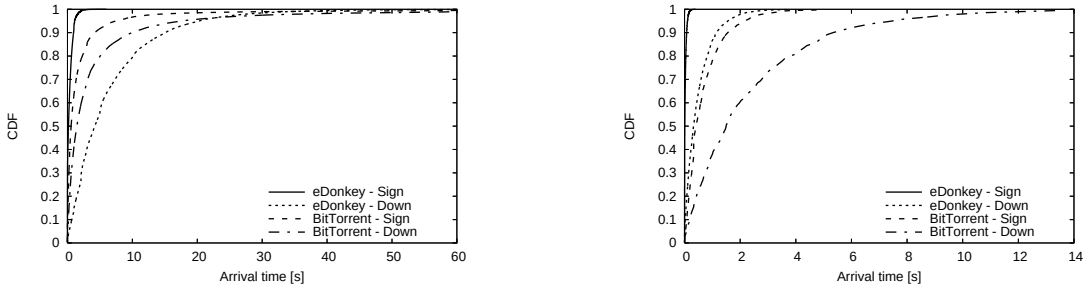
Conversation	Avg.	Std. dev.	Max
eDonkey sign.	0.33	0.43	14.51
eDonkey down.	6.44	7.24	153.58
BitTorrent sign.	4.45	257.70	51812.60
BitTorrent down.	10.61	573.93	73359.40

The CDFs of both the eDonkey and BitTorrent conversation interarrival times reveal an exponential decay, as shown in Figure 2. There is again a significant difference for signaling and download conversations since downloads happen rarely. Note that the graphs for DEPT and GENUA are comparable even though they are two different points of aggregation.

C. Durations

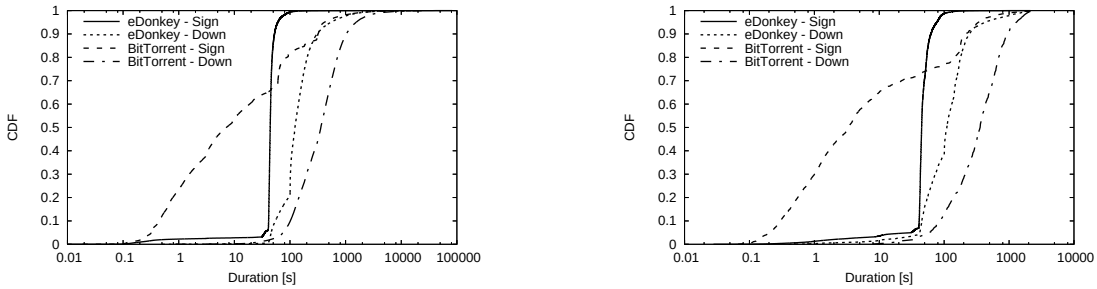
Table IV lists the average, standard deviation and maximum conversation durations.

The CDFs of both the eDonkey and BitTorrent conversation durations, shown in Figure 3, reveal some interesting informa-



(a) DEPT (b) GENUA

Fig. 2. CDF of the observed eDonkey and BitTorrent conversation interarrival times.



(a) DEPT (b) GENUA

Fig. 3. CDF of the observed eDonkey and BitTorrent conversation durations.

TABLE IV
AVERAGE, STANDARD DEVIATION AND MAXIMUM CONVERSATION DURATIONS [S] IN DEPT.

Conversation	Avg.	Std. dev.	Max
eDonkey sign.	59.00	1155.65	600236.74
eDonkey down.	219.27	1098.44	86487.89
BitTorrent sign.	131.68	983.18	87121.08
BitTorrent down.	575.98	1461.66	86746.54

tion. First of all, the curves of the download conversations are very similar for both the protocols and both the traces. This is primarily due to the CTI's capability to distinguish the nature of a conversation, regardless of the specific P2P protocol. The eDonkey signaling conversation durations appear to be concentrated in a small range of values, while the BitTorrent one is distributed in a larger range of small values.

VI. CONCLUSION AND FUTURE WORK

In this paper we have presented a characterization of P2P traffic. We have introduced a new measurement, the CTI, that can be used to distinguish two classes of behavior for the P2P traffic: the download and the signaling traffic profile. We applied the CTI to the eDonkey and BitTorrent conversations in our data sets and we showed that it effectively offers a general characterization of P2P traffic. Finally, we presented a number of statistical measurements that are significantly unbiased because of the distinction in those two profile classes.

In the future, we want to extend the CTI formula to depend on the count of packets in the conversation in order to deal with small unbalanced conversations that would be classified as download but are most likely going to be signaling.

We're also interested to extend this work to different types of P2P applications and apply the CTI to different content such as audio and video.

In a next step, we will define a model based on the presented measures that can be used to generate P2P traffic aggregates. We'll also consider the possibility to build a P2P traffic classifier based on the temporal evolution of the CTI.

ACKNOWLEDGMENT

This work was supported by the project "INTERMEDIA" NoE, in the frame of the EU IST FP6 Program, by MIUR-PRIN project "FAMOUS Fluid Analytical Models Of aUtonomic Systems" and by MIUR-PRIN project "RECIPE Robust and Efficient traffic Classification in IP nEtworks". We also would like to thank CSITA for helping us during the work.

REFERENCES

[1] S. Sen and J. Wang, "Analyzing peer-to-peer traffic across large networks," in *Second Annual ACM Internet Measurement Workshop*, Nov. 2002.

[2] T. Karagiannis, A. Broido, N. Brownlee, kc claffy, and M. Faloutsos, "Is P2P dying or just hiding?" in *IEEE GLOBECOM*, 2004.

[3] S. Sen, O. Spatscheck, and D. Wang, "Accurate, scalable in-network identification of P2P traffic using application signatures," in *Proceedings of the 13th international conference on World Wide Web*, May 2004.

[4] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: Multilevel traffic classification in the dark," in *Proceedings of ACM Sigcomm*, Aug. 2005.
[5] K. Tutschku, "A measurement-based traffic profile of the edonkey filesharing service," in *Proceedings of PAM*, Apr. 2004.
[6] R. Bolla and R. Bruschi, "RFC 2544 performance evaluation and internal measurements for a Linux based open router," in *Proceedings of IEEE Workshop on High Performance Switching and Routing*, Jun. 2006.
[7] l7-filter, http://l7-filter.sourceforge.net.
[8] IPP2P, http://www.ipp2p.org.
[9] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The click modular router," *ACM Transactions on Computer Systems*, vol. 18, no. 3, pp. 263–297, Aug. 2000.
[10] R. Pang, V. Paxson, R. Sommer, and L. Peterson, "binpac: A yacc for writing application protocol parsers," in *Proceedings of ACM Sigcomm Internet Measurement Conference*, Oct. 2006.