Open access • Proceedings Article • DOI:10.1145/1367497.1367537

A comparative analysis of web and peer-to-peer traffic — Source link

Naimul Basher, Aniket Mahanti, Anirban Mahanti, Carey Williamson ...+1 more authors

Institutions: University of Calgary, Indian Institute of Technology Delhi

Published on: 21 Apr 2008 - The Web Conference

Topics: Web traffic, Traffic generation model, Traffic classification, Traffic shaping and Network traffic simulation

A Comparative Analysis of Web and Peer-to-Peer Traffic

Naimul Basher[1], Aniket Mahanti[1], Anirban Mahanti[2], Carey Williamson[1], and Martin Arlitt[1]

[1] University of Calgary, Canada
[2] Indian Institute of Technology Delhi, India

ABSTRACT

Peer-to-Peer (P2P) applications continue to grow in popularity, and have reportedly overtaken Web applications as the single largest contributor to Internet traffic. Using traces collected from a large edge network, we conduct an extensive analysis of P2P traffic, compare P2P traffic with Web traffic, and discuss the implications of increased P2P traffic. In addition to studying the aggregate P2P traffic, we also analyze and compare the two main constituents of P2P traffic in our data, namely BitTorrent and Gnutella. The results presented in the paper may be used for generating synthetic workloads, gaining insights into the functioning of P2P applications, and developing network management strategies. For example, our results suggest that new models are necessary for Internet traffic. As a first step, we present flow-level distributional models for Web and P2P traffic that may be used in network simulation and emulation experiments.

Categories and Subject Descriptors

C.2.2 [**Computer-Communications Networks**]: Network Protocols; I.6.6 [**Simulation and Modeling**]: Model Development

General Terms

Measurement, Performance

Keywords

Web, Peer-to-Peer, Traffic Characterization, Traffic Models

1. INTRODUCTION

In the mid-1990s, a significant proportion of Internet traffic was from applications that used HTTP, the standard protocol for exchanging Web documents. The distinguishing characteristics of Web-dominated Internet traffic include small-sized flows, short-lived connections, asymmetric flow volumes, and well-defined port usage. For the past decade, these characteristics have underpinned the traffic models used in network simulation and emulation experiments.

The introduction of Peer-to-Peer (P2P) file sharing applications, such as Napster in 2000, triggered a paradigm shift in Internet data exchange. P2P applications typically share large multimedia files with individual hosts (called peers), which act as both content providers and consumers. A peer can obtain portions of a file concurrently from multiple peers and/or obtain portions of the same file from a single peer using one or more persistent connections. P2P usage has grown steadily since its inception, and recent empirical studies indicate that Web and P2P together dominate today's Internet traffic [17, 21].

In this paper we use recent packet traces, collected at the gateway of a large university, to extensively characterize and compare traffic generated by Web and P2P applications. Our focus is on characterizing the behaviors of these applications at the flow-level and host-level. The goal of this characterization is to develop flow-level distributional models that may be used to refine models of Internet traffic for use in network simulation and emulation experiments, to provide insights into the similarities and differences between Web and P2P traffic, and to obtain insights into how current P2P applications work.

A distinguishing aspect of our work is the use of recent full-payload packet traces. Popular P2P applications, including BitTorrent, Gnutella, and eDonkey, are known to use dynamic ports, in addition to well-known ports [6,11,20]. Identification of P2P traffic by default port numbers is likely to miss a significant portion of this type of traffic. In fact, our data suggests that as much as 90% of P2P traffic may be on random ports. In this work, we utilize payload-based signature matching to accurately identify P2P traffic.

Our study highlights the evolving nature of Internet traffic due to growing P2P traffic. In addition to studying the aggregate P2P traffic, we also analyze and compare two popular P2P applications: Gnutella and BitTorrent. This study of individual P2P applications aids in understanding the aggregate P2P traffic trends and also helps in understanding how these two applications work. We consolidate our understanding of these traffic types by developing distributional models for each type of traffic; these models can help refine models of Internet traffic. We present high-level results and key observations from our study in Tables 1 and 2. Table 1 summarizes the similarities/dissimilarities between Web and P2P traffic, while Table 2 summarizes the similarities/dissimilarities between Gnutella and BitTorrent traffic.

The remainder of this paper is structured as follows. Our trace collection, traffic identification, and analysis methodologies are described in Section 2. Sections 3 and 4 present flow-level and host-level characterization results, respectively. Section 5 reviews related work. Issues related to trace data collection and analysis are discussed in Section 6. Section 7 summarizes our contributions and lists future work.

WWW 2008, April 21–25, 2008, Beijing, China.
ACM 978-1-60558-085-2/08/04.

Table 1: Key results: Comparing Web and P2P traffic

Characteristics	Web	P2P	Section
Flow Size	Introduces many mice but few elephant flows. Model: hybrid Weibull-Pareto distribution.	Introduces many mice and elephant flows. Model: hybrid Weibull-Pareto distribution.	3.1
Flow Inter-arrival time	Typically short inter-arrival time. Distribution is long-tailed. Model: two-mode Weibull distribution.	Typically long inter-arrival time. Distribution is heavy-tailed. Model: hybrid Weibull-Pareto distribution.	3.2
Flow Duration	Typically short-lived. Model: two-mode Pareto distribution.	Typically long-lived. Model: hybrid Weibull-Pareto distribution.	3.3
Flow Concurrency	Most hosts maintain more than one concurrent flow. Hosts maintain concurrent flows with a few distinct hosts.	Many hosts maintain only one flow at a time. Hosts that maintain more than one flow do so by connecting with many distinct hosts.	4.1
Transfer Volume	Large transfers are dominated by downstream traffic. Heavy-hitters account for a large portion of total transfer and their transfers follow a power-law distribution.	Large transfers happen in either upstream or downstream direction. Heavy-hitters account for a huge portion of total transfer and their transfers follow a power-law distribution.	4.2
Geography	Most external hosts are located primarily in the same geographic region.	External peers are globally distributed.	4.3

Table 2: Key results: Comparing Gnutella and BitTorrent traffic

Characteristics	Gnutella	BitTorrent
Flow Size	Both small and large flows are observed. Elephants are relatively more frequent. Distribution is heavy-tailed. Model: hybrid Lognormal-Pareto distribution.	Small flows are prevalent. Elephants are less frequent, but comparatively large. Distribution is heavy-tailed. Model: hybrid Lognormal-Pareto distribution.
Flow Duration	Typically short-lived. Distribution is heavy-tailed.	Typically long-lived. Distribution is long-tailed.
Flow Concurrency	Peers mostly connect to a single host at a time.	Peers maintain many concurrent flows with a large number of distinct hosts.
Transfer Volume	Transfers are extremely asymmetric and dominated by single direction traffic. Heavy hitters account for less volume of traffic.	Transfers are comparatively less asymmetric and more balanced. Heavy-hitters contribute more traffic volume.
Geography	External peers are mostly concentrated in the same geographic region.	External peers are from regions with broadband connectivity.

2. METHODOLOGY

2.1 Trace Collection and Traffic Identification

The network traffic traces used in this work were collected from the commercial Internet link[1] of the University of Calgary, a large research-intensive university with 28,000 students and 5,000 employees. We used `lindump`[2] running on a dual processor 1.4 GHz Pentium system with 2 GB memory and 70 GB disk space to capture TCP/IP packets via port mirroring.

Identifying P2P traffic correctly in the traces is a challenge. One approach, which has been used in some recent P2P characterization studies [17,21,24], is to map network traffic to applications using well-known port numbers. However, many P2P applications including BitTorrent and Gnutella use dynamic port numbers. This necessitated the use of payload signatures [11,20] to identify applications.

We used `Bro` [15], an open source Network Intrusion Detection System, to perform the payload signature matching. The built-in payload "signature matching engine" in `Bro` was used to perform the mapping of network flows to application types. We used the signatures described by Sen *et al.* [20] and Karagiannis *et al.* [11]; details of our payload-based identification scheme can be found in [6]. We identify the start of a TCP flow using connection establishment semantics (i.e., `SYN-SYNACK-ACK` packet transmissions) or by the first packet transmission observed between hosts, and end of a TCP flow after observing a `FIN` or `RST` packet. By default, `Bro` considers a flow terminated if it is idle for more than 900 seconds.

The payload-based identification technique requires traces with relevant application-layer headers. The signature strings for some P2P applications (e.g., Gnutella) can be buried deep inside a packet [6]; therefore, successful string matching requires full-packet payloads. This poses another challenge: the huge storage space required for full-packet trace collection from a high-speed Internet connection for an extended interval (e.g., a day or a week). For our work, we used non-contiguous one-hour traces collected between April 6 and April 30, 2006. The traces were collected each morning (9-10 am) and evening (9-10 pm) on Thursday through Sunday every week (i.e., eight one-hour traces per-week). Although discontinuous traces limit the analysis of long-term traffic behavior, we expect the traces to capture morning/evening and weekday/weekend trends. Our methodology also captured behavioral aspects related to the academic calendar.

2.2 Trace Summary

The traces contain 1.12 billion IP packets totalling 639.4 Gigabytes (GB) of data. In this paper, attention is restricted to only TCP/IP packets because these account for 84.4% of the total packets and 92% of the total bytes in the traces. Furthermore, Web and P2P applications such as Gnutella and BitTorrent use TCP in most cases. In total, we consider 23.3 million TCP flows with 946 million IP packets and 588.3 GB of data.

Table 3 shows the breakdown by application type. Web and P2P dominate in terms of bytes. Although P2P accounts for only 2.8% of the total flows, it accounts for 33.1% of the total bytes. The Unknown category includes HTTPS (port 443), flows without payloads, and flows unclassified by `Bro`. The Others category bundles together the remaining traffic; the main contributors (by bytes) are email (5%), file transfer (3%), and streaming (2%) applications.

[1] At the time of trace collection, the Internet link was a 100 Mbps full-duplex connection.

[2] `http://awgn.antifork.org/codes/lindump.c`

Table 3: Flow and byte count by applications

Application	Flows	% Flows	Bytes (GB)	% Bytes
Web	9,213,424	39.51	204.32	34.73
P2P	646,082	2.77	194.96	33.14
Unknown	9,275,013	39.77	68.42	11.62
Others	4,186,232	17.95	120.61	20.51
Total	23,320,751	100.00	588.31	100.00

Table 4: Flow and byte count for P2P

P2P Systems	Flows	% Flows	Bytes (GB)	% Bytes
Gnutella	137,024	21.21	151.51	77.71
BitTorrent	393,641	60.93	31.88	16.36
eDonkey	79,796	12.35	2.64	1.35
Other-P2P	35,621	5.51	8.93	4.58
Total	646,082	100.00	194.96	100.00

Table 4 categorizes the P2P flows present in our traces by P2P application type. There are approximately 646,000 P2P flows; these account for nearly 195 GB of traffic data. From the table, we notice that BitTorrent has a lower byte-to-flow ratio than Gnutella. Table 4 also reveals that although eDonkey accounts for many P2P flows, the cumulative traffic volume in bytes was relatively small. The Other-P2P category consists of P2P applications that each contributed less than 1% of the identified P2P flows.

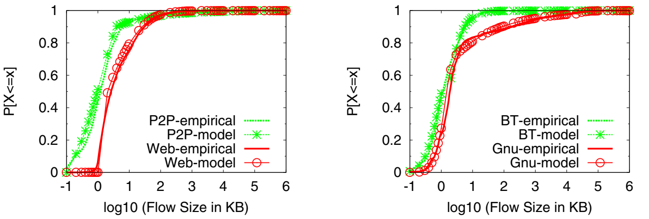
2.3 Characterization Metrics

We consider three flow-level characterization metrics:
Flow Size – the total bytes transferred during a TCP flow. Flows can be categorized as mice [25], buffalo [22] and elephants [13]. We label flows as *mice* if they transfer less than 10 Kilobytes (KB), and as *elephants* if they transfer more than 5 Megabytes (MB) of data. The rest are labeled as *buffalo*.
Flow Duration – the time between the start and the end of a TCP flow.
Flow Inter-arrival time – the time interval between two consecutive flow arrivals.

We consider three host-level characterization metrics:
Flow Concurrency – the maximum number of TCP flows a single host uses concurrently to transfer content.
Transfer Volume – the total bytes transferred to and from a host during its activity period. *Upstream* transfer volume is measured as the total bytes transmitted from an internal host to the external hosts. *Downstream* transfer volume is the total bytes received by an internal host from the hosts external to the network.
Geographic Distribution – the distribution of the shortest distance between individual hosts and our campus along the surface of the Earth. This distance measure is known as the *great-circle*[3] distance.

2.4 Statistical Measures and Models

We use statistical measurements such as mean, median, standard deviation, inter-quartile range (IQR), and skewness to summarize trends of the sample data. Where necessary, we also use the probability density function (PDF), cumulative distribution function (CDF), and complementary CDF (CCDF) of the sample data to obtain further insights.



(a) Web and P2P (b) Gnutella and BitTorrent

Figure 1: CDF of flow sizes

References to the "tail" of the CCDF refer to those values in the upper 10% of the empirical distribution; the remaining 90% of the distribution is referred to as the body. CCDF tails are often studied to determine how quickly or slowly they decay. A distribution where the tail decays more slowly than an exponential distribution is called *long-tailed*. A distribution is *heavy-tailed* if the tail asymptotically follows a hyperbolic shape (i.e., shape parameter $0 < \alpha \leq 2$).

We present statistical models that capture the salient features seen in our data sets. We use the following distributional models: Pareto (CDF: $1 - (\frac{\beta}{x})^{\alpha}$), Weibull (CDF: $1 - e^{-(\frac{x}{\beta})^{\alpha}}$), and Lognormal (CDF: $\Phi\left(\frac{ln\ x-\mu}{\sigma}\right)$) where α and β are shape and scale parameters, μ and σ are mean and standard deviation of the distribution, and Φ is the Laplace Integral; we also present models that are hybrid of the aforementioned distributions, where the model thresholds were determined manually such that the hybrid distribution passed a goodness-of-fit test. We tested the statistical models for accuracy using the Kolmogorov-Smirnov (K-S) goodness-of-fit test. If the statistical model passed the K-S test at the 5% significance level, we considered it to model our empirical data well.[4] Only these models are presented in the paper.

3. FLOW-LEVEL CHARACTERIZATION

In order to conduct realistic network simulations, models of flow size, inter-arrival time, and duration are needed. In this section, we present our flow-level characterization results and derive distributional models from the characterization results. Summary statistics for Web and P2P traffic are presented in Table 5. The corresponding statistics for Gnutella and BitTorrent are shown in Table 6.

3.1 Flow Size

3.1.1 Web and P2P Flow Sizes

Table 5 shows that P2P flows have a higher mean flow size and lower median flow size than Web flows. These observations suggest that P2P applications generate many small and many very large-sized flows compared to Web. The CDF of Web and P2P flow sizes in Figure 1(a) corroborates the aforementioned observation.

The preponderance of small-sized P2P flows is somewhat unexpected as P2P applications are typically used to share large audio and video files. There are at least three sources of small-sized flows: extensive signalling, aborted transfers, and connection attempts with non-responsive peers. We also find some very large-sized P2P flows. These few P2P flows are much larger than the occasional large Web transfer. Our analysis indicates that *P2P applications contribute*

[3] `http://en.wikipedia.org/wiki/Great-circle_distance`

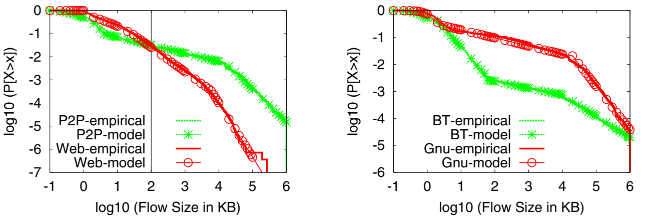
[4] We validated the models using a distribution fitting tool called EasyFit: `http://www.mathwave.com/products/easyfit.html`.

Table 5: Flow-level summary statistics of Web and P2P

Characteristic	Web					P2P				
	Mean	Median	Std. Dev.	IQR	Skewness	Mean	Median	Std. Dev.	IQR	Skewness
Flow size (KB)	21.50	2.53	341.92	7.38	44.03	362.40	1.17	12470	1.89	192.13
Flow Inter-Arrival (sec)	0.11	0.007	3.53	0.016	26.05	1.77	0.18	17.21	0.39	48.69
Flow duration (sec)	13.32	0.40	56.71	1.80	14.48	123.54	24.80	274.37	93.30	7.61

Table 6: Flow-level summary statistics of Gnutella and BitTorrent

Characteristic	Gnutella					BitTorrent				
	Mean	Median	Std. Dev.	IQR	Skewness	Mean	Median	Std. Dev.	IQR	Skewness
Flow size (KB)	1159.40	1.89	15549	2.73	94.68	84.95	0.96	11189	2.10	292.31
Flow Inter-Arrival (sec)	2.30	0.21	22.22	0.51	30.15	2.46	0.42	20.25	0.99	49.78
Flow duration (sec)	89.35	9.70	386.22	25.60	8.12	135.43	33.20	221.41	180.90	3.03



(a) Web and P2P

(b) Gnutella and BitTorrent

Figure 2: CCDF of flow sizes

many mice and elephant flows, and possibly alters the mix of these flow types in today's IP networks. We elaborate on this phenomenon in Section 3.1.3.

We examined the tails of the flow size distributions using CCDF plots. Figure 2(a) presents the CCDF of flow sizes for Web and P2P. In the body of the distribution, P2P flows are smaller than Web flows, but in the tail (specifically, the upper 3.5% of flows after the "crossover" point) P2P flows are larger than Web flows. Also, the tail of the Web flow size distribution decays more quickly than the corresponding P2P distribution. These observations provide further evidence of P2P's large elephant-sized flows.

3.1.2 Gnutella and BitTorrent Flow Sizes

Table 6 indicates that Gnutella flow sizes are larger and more dispersed than BitTorrent flow sizes. The empirical CDF for the two P2P variants in Figure 1(b) shows that both applications generate a similar percentage of small-sized flows (e.g., 5 KB or less). Many of these smaller flows are the result of control information exchanged between peers, which is a byproduct of the distributed nature of P2P protocols. The ratio of large-sized to total flows for BitTorrent is, however, less than that for Gnutella. For example, approximately 5% of BitTorrent flows are larger than 10 KB, whereas 17% of Gnutella flows exceed this size. The characteristics of these large-sized flows are analyzed next.

Figure 2(b) shows the CCDF of flow sizes of Gnutella and BitTorrent applications. Gnutella appears to generate more large-sized flows than BitTorrent. BitTorrent uses *file segmentation* to split an object into multiple equal-sized "pieces" (256 KB each by default), and downloads these pieces from either the same or different peers using parallel flows. In contrast, Gnutella typically downloads the entire object from a single peer. As a result, we observe *fewer large flows in BitTorrent than Gnutella.*

3.1.3 Mice and Elephant Phenomenon

Table 7 shows the percentage of mice and elephant flows among the total flows contributed by different applications.

Table 7: Mice and elephant flow breakdown

Application	Mice		Elephants	
	% Flows	% Bytes	% Flows	% Bytes
Web	75.78	8.89	0.04	15.35
P2P	92.93	0.47	0.81	93.43
Gnutella	83.41	0.14	3.05	93.14
BitTorrent	94.96	1.94	0.08	94.87

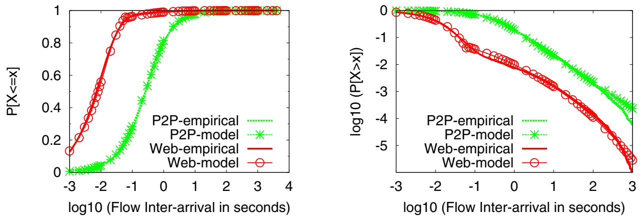
We observe that both categories of application generate many mice flows. Although the mice flows originating from Web applications are less prevalent than those from P2P applications, Web mice flows account for a relatively higher proportion of the total Web bytes than P2P mice flows account for the total P2P bytes. For example, approximately 9% of total Web bytes are from Web mice flows, whereas only 0.4% of total P2P bytes are transferred by P2P mice flows.

Both applications generate a small proportion of elephant flows. Nevertheless, these few elephant flows contribute a significant fraction of the total bytes; the elephant-sized Web flows contributed about 15% of the total Web-generated bytes, while the elephant-sized P2P flows contributed as much as 93% of the total P2P bytes. Network operators may be interested in bandwidth-limiting these long-duration "elephant" flows, or may be interested in assigning these flows lower priority. As P2P applications become more popular, we can expect networks to carry increasingly more elephant flows. Our results also indicate that *P2P elephant flows are significantly larger than Web elephant flows.*

We next analyze mice and elephant flows generated by Gnutella and BitTorrent. While both P2P applications have a similar proportion of mice flows, the BitTorrent mice flows account for a much higher percentage of byte transfers than Gnutella mice flows; that is, Gnutella mice flows are smaller, on average, than BitTorrent mice flows. As mentioned earlier, signalling between peers is a major contributor to the pool of P2P mice flows. Our data suggests that BitTorrent applications have more intense signaling activities compared to Gnutella, resulting in relatively larger mice flows.

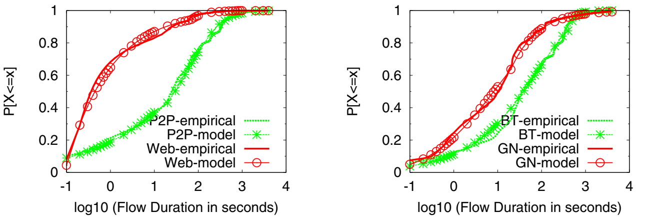
In our data, Gnutella has a much higher percentage of elephant flows than BitTorrent, even though both Gnutella and BitTorrent elephant flows account for a comparable proportion of byte transfers. Thus, on average, BitTorrent elephant flows are larger than Gnutella elephant flows. We believe that the type of files exchanged using these P2P systems can provide an explanation for our observation. A 2005 study by CacheLogic[5] showed that a majority of Gnutella users shared mostly audio files (70%), whereas BitTorrent users shared more video files (47%). Video files are, on av-

[5] CacheLogic. Peer-to-Peer File Type Study, `http://www.cachelogic.com/home/pages/research/filetypestudy.php`



(a) CDF (b) CCDF

Figure 3: Web and P2P flow inter-arrival



(a) Web and P2P (b) Gnutella and BitTorrent

Figure 4: CDF of flow duration

erage, significantly larger than audio files. We believe that the extremely large BitTorrent flows are due to the transfer of multiple pieces of large video files over a single TCP flow.

3.1.4 *Flow Size Models*

In this section, we present statistical models that describe the body and the tail of flow size (S) distribution. These models may be used to generate transfer sizes of TCP flows in network simulations. Figures 1 and 2 plot the statistical models in addition to the empirical distributions. Web flow sizes are well-modeled by a concatenation of bounded Weibull and Pareto distributions:

$$F_{Web}(S) = \begin{cases} 1 - e^{-(\frac{S}{2.7})^{0.38}} & : S < 30KB \\ 1 - (\frac{3}{S})^{1.05} & : 30KB \leq S \leq 5MB \\ 1 - (\frac{200}{S})^{2.35} & : S > 5MB \end{cases}$$

We find that *the tail of the Web flow size distribution is a mix of heavy-tailed and long-tailed distributions.*

Similarly, we find that P2P flow sizes are well-modeled by a hybrid bounded Weibull and Pareto distributions:

$$F_{P2P}(S) = \begin{cases} 1 - e^{-(\frac{S}{1.36})^{0.81}} & : S < 4KB \\ 1 - (\frac{0.005}{S})^{0.35} & : 4KB \leq S \leq 10MB \\ 1 - (\frac{400}{S})^{1.42} & : S > 10MB \end{cases}$$

From the above-mentioned model, we can conclude that *P2P flow sizes are heavy-tailed.*

Both the BitTorrent and Gnutella flow sizes are well-modeled by combining bounded Lognormal and Pareto distributions:

$$F_{BT}(S) = \begin{cases} \Phi\left(\frac{ln\ S - 0.03}{0.95}\right) & : S < 2KB \\ 1 - (\frac{1.07}{S})^{1.4} & : 2KB \leq S \leq 50KB \\ 1 - (\frac{3\times10^{-9}}{S})^{0.25} & : 50KB < S \leq 7MB \\ 1 - (\frac{0.95}{S})^{0.78} & : S > 7MB \end{cases}$$

$$F_{Gnu}(S) = \begin{cases} \Phi\left(\frac{ln\ S - 0.44}{0.73}\right) & : S < 3KB \\ 1 - (\frac{0.04}{S})^{0.3} & : 3KB \leq S \leq 10MB \\ 1 - (\frac{1800}{S})^{1.61} & : S > 10MB \end{cases}$$

We find that *both BitTorrent and Gnutella flow size distributions are heavy-tailed; BitTorrent flow sizes, however, are less heavy-tailed than Gnutella flows.*

3.2 Flow Inter-arrival Times

3.2.1 *Web and P2P Inter-arrival Times*

Analysis of our data (see Table 5) shows that P2P flow inter-arrival times (IAT) are much longer and more dispersed than Web flow IAT. Figure 3 shows the CDF and CCDF of flow IAT for Web and P2P. Web flow IAT are much shorter than those of P2P flows. For example, approximately 97% of Web flow IAT are less than 0.1 second, whereas only 25% of P2P flow IAT are this short.

Another way to understand the difference between the IAT of Web and P2P flows is to study their corresponding flow arrival rates. Web traffic has a higher arrival rate of approximately 80 flows/seconds, compared to P2P traffic, which has arrival rate of only 6 flows/seconds. Another factor contributing to the lower arrival rate and the longer IAT values for P2P flows is the persistent nature of their TCP connections. How these persistent connections are used is discussed in Section 4.1.

We examine the tails of flow IAT for Web and P2P in Figure 3(b). Flow IAT from both applications show similar decay throughout the tails. At the upper tail, we observe sharp decay due to the limited duration of our traces. Flow IAT from individual P2P applications are found to follow similar patterns, and thus are not shown here.

3.2.2 *Inter-arrival Time Models*

We find that Web flow IAT can be modeled by a two-mode bounded Weibull distribution:

$$F_{Web}(IAT) = \begin{cases} 1 - e^{-(\frac{IAT}{0.01})^{0.76}} & : IAT \leq 0.06\,sec \\ 1 - e^{-(\frac{IAT}{3\times10^{-5}})^{0.15}} & : IAT > 0.06\,sec \end{cases}$$

In contrast, P2P flow IAT are well-modeled by a hybrid Weibull-Pareto distribution:

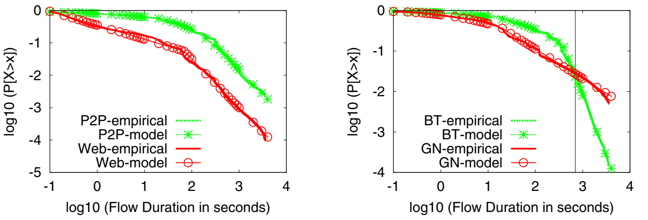
$$F_{P2P}(IAT) = \begin{cases} 1 - e^{-(\frac{IAT}{0.35})^{0.87}} & : IAT \leq 0.1\,sec \\ 1 - e^{-(\frac{IAT}{0.45})^{0.65}} & : 0.1 < IAT \leq 1\,sec \\ 1 - (\frac{0.18}{IAT})^{0.97} & : IAT > 1\,sec \end{cases}$$

These distribution models indicate that *Web IAT are long-tailed, whereas P2P IAT are heavy-tailed.* Our models provide evidence of the inapplicability of memoryless Poisson models for Web and P2P flow arrivals [16].

3.3 Flow Duration

3.3.1 *Web and P2P Flow Durations*

Our statistical analysis (cf. Table 5) indicates the presence of many short-duration flows. Figure 4 shows the CDF of flow durations. From Figure 4(a) we observe that approximately 30% of P2P flows are shorter than 10 seconds in duration. Some of these short-duration transfers are either failed or aborted flows, while other short-duration flows are a byproduct of the P2P applications' signaling behavior. Note that short-duration flows typically transfer a small amount of data, but the converse does not always hold. There are a few long-duration mice flows; these flows arose due to repeated unsuccessful connection attempts by peers. We also



(a) Web and P2P (b) Gnutella and BitTorrent

Figure 5: CCDF of flow duration

observe that a large proportion, approximately 40%, of P2P flow durations are between 20 and 200 seconds. We found that some P2P connections are bandwidth-limited, and thus of long-duration. Bandwidth limitations reflect the available bandwidth between peers (e.g., peers with asymmetric Internet access have limited uplink capacity) as well as flow management on our network (cf. Section 6). Approximately 70% of the Web flows last no longer than 1 second. End users have excellent Internet connectivity in our campus network, and most Web servers are also well-provisioned. Thus, we expect low response times for Web requests. The remaining Web flows that are longer than 1 second are typically responsible for either downloading large objects (e.g., streaming video from `youtube.com`) or transferring multiple objects from Web pages using persistent HTTP/1.1 connections.

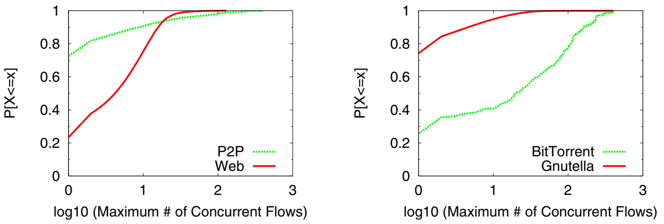
In Figure 5 we analyze the tail of the flow duration distributions. Figure 5(a) shows the CCDF of Web and P2P flow durations. We find that *the probability of long-duration flows is higher for P2P than Web.*

3.3.2 Gnutella and BitTorrent Flow Durations

Summary statistics in Table 6 show that, on average, BitTorrent flows last longer than Gnutella flows; furthermore, the flow durations are dispersed over a wide range of values.

Figure 4(b) shows the CDF of Gnutella and BitTorrent flow durations. This graph reaffirms the aforementioned point. We find that these relatively longer flows of BitTorrent resulted due to its protocol architecture. BitTorrent utilizes a rarest first piece selection policy to exchange data. At any given time, a fixed number of concurrent uploads/downloads are permitted. BitTorrent architecture allows persistent connections between peers and controls downloads/uploads using its piece selection policy which results in connections periodically being idle. Furthermore, concurrent download from a single BitTorrent peer splits the bandwidth available at uploaders for downloading. In contrast, Gnutella can use a single flow for downloading an object and thus does not need to share bandwidth. Occasionally, Gnutella peers may share bandwidth, for example, when the same object is requested by other peers or when different objects are requested by the same peer.

Figure 5(b) shows the CCDF of Gnutella and BitTorrent flow duration. Two observations can be drawn. First, before the crossover point, BitTorrent shows a higher percentage of long-duration flows than Gnutella; however, following the crossover point (upper 2% of flows), the probability of long-duration flows in Gnutella is higher than that in BitTorrent. Second, at the distribution tail, BitTorrent flow durations decay more quickly than Gnutella flow durations. We found earlier that extremely large transfers are not very common in BitTorrent, due to its file segmentation feature. We also found a positive correlation (correlation coefficient is 0.69) between BitTorrent flow size and duration, and therefore, observe a lower proportion of extremely long-duration flows in BitTorrent. Other factors such as file size, swarm population, and availability of pieces in the swarm can also influence the duration of BitTorrent flows. These factors result in the BitTorrent tail being long-tailed instead of heavy-tailed.



(a) Web and P2P (b) Gnutella and BitTorrent

Figure 6: CDF of host flow concurrency

3.3.3 Flow Duration Models

This section outlines the statistical models of flow durations (D) (see Figures 4 and 5). Web flow duration is well-modeled using two bounded Pareto distributions:

$$F_{Web}(D) = \begin{cases} 1-(\frac{0.1}{D})^{0.43} & : D \leq 60\,sec \\ 1-(\frac{10}{D})^{1.5} & : D > 60\,sec \end{cases}$$

The preceding model shows that *Web flow durations are heavy-tailed.* A similar analysis shows that P2P flow durations can be well-modeled by a concatenation of bounded Weibull and heavy-tailed Pareto distribution:

$$F_{P2P}(D) = \begin{cases} 1-e^{-(\frac{D}{88.3})^{0.35}} & : D < 20\,sec \\ 1-e^{-(\frac{D}{57.2})^{0.55}} & : 20 \leq D \leq 300\,sec \\ 1-(\frac{65}{D})^{1.53} & : D > 300\,sec \end{cases}$$

BitTorrent flow durations are well-modeled by a hybrid bounded Weibull and Pareto distributions, whereas Gnutella flow durations are well-modeled by a hybrid bounded Lognormal and Pareto distributions:

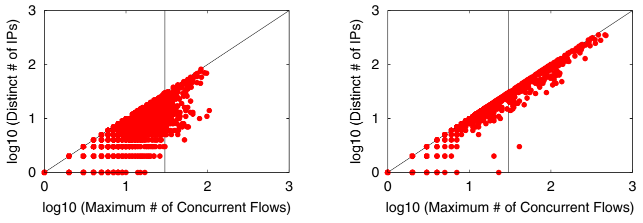
$$F_{BT}(D) = \begin{cases} 1-e^{-(\frac{D}{83.5})^{0.48}} & : D \leq 300\,sec \\ 1-(\frac{200}{D})^{3} & : D > 300\,sec \end{cases}$$

$$F_{Gnu}(D) = \begin{cases} \Phi\left(\frac{ln\ D-2.1}{2.7}\right) & : D \leq 10\,sec \\ 1-(\frac{5}{D})^{0.73} & : D > 10\,sec \end{cases}$$

The above-mentioned statistical distributions show that *BitTorrent flow durations are long-tailed* (tail fits a Pareto distribution with $\alpha > 2$) *but not heavy-tailed.* In contrast, *Gnutella flow durations are heavy-tailed.*

4. HOST-LEVEL CHARACTERIZATION

This section presents a host-level characterization of Web and P2P traffic. This characterization provides information to network administrators for tasks such as bandwidth management and capacity planning, and also provide insights into the functioning of modern P2P systems. The results presented here may also be used to develop synthetic workloads and design realistic network simulations.



(a) Web (b) P2P

Figure 7: Flow concurrency vs distinct IPs

4.1 Flow Concurrency

Figure 6 shows the CDF of host flow concurrency for Web, P2P, Gnutella, and BitTorrent. From Figure 6(a), we observe (surprisingly) that *many P2P hosts in our network maintain only a single TCP connection.* We explain the observation later in this section by analyzing flow concurrency for individual P2P applications. While analyzing the flow concurrency for Web hosts, we ignore the Web servers internal to our network. From the analysis, we find that *a significant proportion of the internal Web hosts maintain more than one concurrent TCP connection.* Web browsers often initiate multiple concurrent connections to transfer content in parallel. This parallel download feature increases the degree of flow concurrency in HTTP-based applications. However, a high-degree of flow concurrency (e.g., above 30) is not typically observed for general Web clients; rather, Web proxies and content distribution nodes account for this high degree of flow concurrency.

The CDF of host flow concurrency for Gnutella and BitTorrent is shown in Figure 6(b). We observe that *most Gnutella hosts connect with only one host at a time.* As discussed earlier, Gnutella applications typically download a whole object from another Gnutella host using a single TCP flow. We observed a few Gnutella hosts that maintained more than 10 concurrent TCP connections. These hosts likely acted as “super peers” in Gnutella’s peer hierarchy. In contrast, *most BitTorrent hosts exhibit a high degree of flow concurrency.* Approximately 24% of the BitTorrent hosts use more than 100 concurrent flows. This high degree of concurrency is a natural occurrence in BitTorrent. BitTorrent clients obtain a peer list from a tracker, and then attempt to connect with these peers. Once connections are established, BitTorrent uses its rarest first piece selection policy and tit-for-tat fairness mechanisms to determine how pieces are shared [3]. Typically, only a small number of these concurrent connections actively transfer file pieces.

We also study the correlation between the maximum number of concurrent flows seen at a host and the number of distinct hosts connected at that time. Figure 7 shows scatter plots of flow concurrency versus distinct hosts for Web and P2P hosts. (The plots for Gnutella and BitTorrent are similar to that of P2P, and thus not shown here.) From Figure 7(a) we observe that most of the points are well-below the diagonal. In other words, *the number of concurrent Web flows far exceed the number of Web hosts concurrently contacted.* From Figure 7(b), we observe that *P2P hosts use concurrent flows to connect to many distinct hosts* as illustrated by the concentration of points along the diagonal. This behavior is not unexpected, since P2P protocols such as BitTorrent and eDonkey encourage connectivity with multiple hosts to facilitate widespread sharing of data.

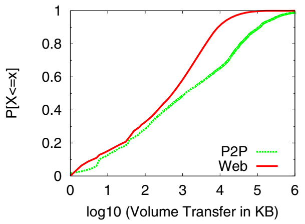


Figure 8: CDF of transfer volume

Table 8: Fair-share ratio in P2P systems

Downstream (MB)	Minimum Fair-share Ratio
< 1	none
1 - 20	0.01
20 - 40	0.25
40 - 60	0.35
60 - 80	0.45
80 - 100	0.55
> 100	0.65

4.2 Transfer Volume

This section studies the transfer activity of hosts in terms of their transfer volume. Figure 8 show the CDF of the transfer volume for Web and P2P hosts. We observe that approximately half of the distinct P2P and Web hosts transfer small amounts of data (e.g., less than 1 MB); these hosts are typically active for less than 100 seconds. We find that these P2P hosts repeatedly yet unsuccessfully attempt to connect with serving peers. Connection requests are unsuccessful for a variety of reasons including insufficient resources or no useful content at the contacted peers. In contrast, Web transfers in this region result from Web browsing, widgets that retrieve information from the Web periodically (e.g., weather updates, stock prices), and downloading small files.

We find that approximately 35% of Web hosts and 15% of P2P hosts transfer data ranging from 1 to 10 MB, and are active mostly for 100 to 1000 seconds. These P2P host transfers are due to sharing small objects, whereas these Web host transfers are due to prolonged Web browsing, downloading software/multimedia files, and HTTP-based streaming. The proportion of hosts that transfer large amounts of data (e.g., 10 MB or more) and are active for over 1000 seconds, is significantly higher in P2P than in Web.

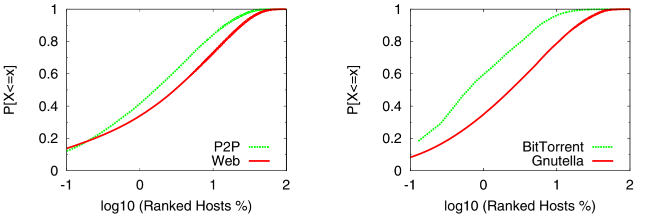
4.2.1 Transfer Symmetry in P2P Systems

Transfer symmetry is a major concern for P2P system developers, who want to encourage fair sharing among participating peers. Many content sharing portals require that users maintain a minimum ratio of upstream to downstream transfer volume, which we refer to as the minimum fair-share ratio. Table 8 shows the minimum ratios of fair-sharing we defined for different levels of downstream traffic. Note that hosts transferring less than 1 MB of data in total are not sharing any content and thus are excluded from our transfer symmetry calculation. In most cases, we used equal-sized bins to assign minimum fair-share ratios; however, for above 100 MB of data transfer, we used a single bin as only 10% of P2P hosts fall in this category.

We divide P2P hosts into three categories (*freeloaders*, *fair-share*, and *benefactors*) according to their transfer ratios (i.e., upstream/downstream ratios) and corresponding minimum fair-share ratios from Table 8. We define freeloaders

Table 9: Transfer symmetry in P2P systems

Systems	Freeloader	Fair-share	Benefactor
Gnutella	56.93%	10.00%	33.07%
BitTorrent	10.30%	39.91%	49.79%



(a) Web and P2P (b) Gnutella and BitTorrent

Figure 9: CDF of ranked hosts

as those hosts who have a transfer ratio less than the minimum fair-share ratio. Benefactors are hosts that have a transfer ratio of 2 or greater. The remaining hosts are in the "fair-share" range.

Table 9 shows the percentage of Gnutella and BitTorrent hosts as freeloaders, fair-share hosts, and benefactors. We find that approximately 10% of BitTorrent hosts are acting as freeloaders, whereas 57% of Gnutella hosts are freeloaders. Benefactors are common in both BitTorrent (∼50%) and Gnutella (∼33%) hosts. Therefore, Gnutella host behavior appears to be dominated by extreme downstream and upstream transfers. We find that approximately 40% of BitTorrent peers and 10% of Gnutella peers reside in the fair share zone. BitTorrent introduced a "tit-for-tat" mechanism to encourage fair sharing among the peers [3]. Every peer in the BitTorrent system is encouraged to upload for obtaining the opportunity to download. Therefore, we observe *more freeloaders in Gnutella and better fairness in BitTorrent.*

4.2.2 *Heavy-hitters*

Figure 9 plots the CDF of hosts ranked by transfer volume (the higher the amount of data transferred, the higher the rank). We find that a few hosts account for much of the volume transferred; we call these hosts *heavy-hitters*.

Figure 9(a) shows that the top 0.1% of Web hosts account for 14% (28 GB) of the total Web transfer. Similarly, the top 0.1% of P2P hosts transfer 12% (24 GB) of the total P2P data. Moreover, top 1% of Web and P2P hosts account for 70 GB (34%) and 82 GB (42%) of the total Web and P2P bytes, respectively. Clearly, *heavy-hitters are present in both Web and P2P.* Examination of the upstream to downstream transfer ratio for the P2P heavy-hitters shows that *most P2P heavy-hitters are either freeloaders or benefactors.*

Figure 10 shows the transfer volume of ranked Web and P2P hosts. We observe that *the total amount of data transferred by the top* 10% *Web and P2P hosts follows a power-law distribution (with* $\alpha \approx 0.27$*)*; we emphasize that the power-law does not apply to the body and the tail of the ranked distribution. The only difference seen between the applications is the total transfer volume; top-ranked P2P hosts transfer an order of magnitude more data than top-ranked Web hosts.

Figure 9(b) shows the CDF of ranked transfer volume for Gnutella and BitTorrent hosts. We find that the top 1% of BitTorrent hosts transfer 20 GB (60% of total BitTorrent traffic), whereas the top 1% of Gnutella hosts account for 53 GB (35% of total Gnutella traffic). Our data suggests that

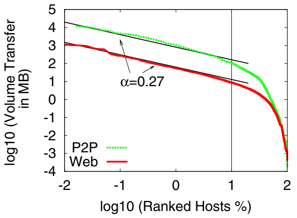
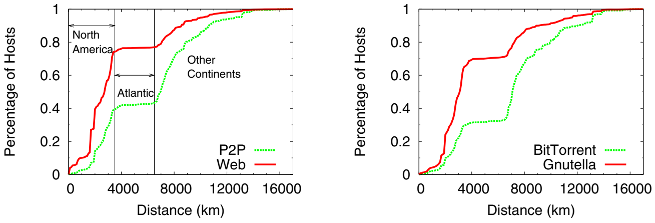


Figure 10: Transfer volume of ranked host



(a) Web and P2P (b) Gnutella and BitTorrent

Figure 11: Geographic distribution of hosts

BitTorrent heavy-hitters account for a much larger fraction of that application's total bytes than Gnutella heavy-hitters do for their total bytes. We also found that the transfer volume of top-ranked Gnutella and BitTorrent hosts did not follow a power-law distribution.

4.3 Geographic Distribution

This section discusses the geographic distribution of hosts external to the campus network. We calculated the great-circle distance between individual hosts and our campus using a geolocation database[6]. This database provides the geographic coordinates, country name, and city name for an IP address range.

Figure 11 shows the geographical distribution of the external hosts. Note the plateau between 3,500 and 7,000 kilometers represents the Atlantic ocean. Figure 11(a) shows the geographical distribution of the external Web and P2P hosts. Most of the external Web hosts, approximately 75%, are in North America; Asia and Europe each account for 10% of the external Web hosts. The results here are not surprising. We know that most of the external Web hosts are Web servers. O'Neill *et al.* [14] had shown that in 1999 and 2002, 49% and 55% of the public Web sites, respectively, were associated with entities located in the United States. In addition, we believe that cultural pecularities may also affect the results. A majority of our campus Web users are English-speaking, and thus they are more likely to visit Web sites located in predominantly English-speaking countries.

In contrast to the geographic distribution of external Web hosts, we found that approximately 40% of P2P hosts are located in North America, 30% in Europe, 18% in Asia, 6% in Australia, and 5% in South America. This indicates that *connectivity between P2P hosts does not appear to strongly rely on host locality*, rather it depends on resource availability during the connection establishment phase. The non-interactive nature of P2P applications makes latency only a secondary concern; the primary goal is to find the requested file. In addition, our results suggest that files being shared using these systems transcend geographic divides.

[6] MaxMind: GeoIP City Database, `http://www.maxmind.com/app/city`

The host geographical distribution for the P2P variants are shown in Figure 11(b). It shows that majority of external Gnutella hosts (∼70%) are from North America. Approximately 18% of the Gnutella hosts are located in Europe and the rest are in Asia (6%), Australia (2.3%), and South America (2.3%). This suggests that either Gnutella peers prefer to connect with hosts that are in close proximity or that Gnutella clients are widely used in North America for file-sharing. In contrast, only 30% of external BitTorrent hosts are located in North America. Among the rest, approximately 40% of BitTorrent hosts are located in Europe, 18% in Asia, 6% in Australia, and 3% in South America. We know BitTorrent hosts connect to peers from a peer-list provided by trackers. We believe that the list from trackers is created based on host bandwidth availability in a swarm and thus, we see a bias towards regions with high broadband penetration. We did observe, however, that *although BitTorrent peers connect to other distant peers for obtaining content, most of the successful transfers originate from the peers located in the same geographic region.*

5. RELATED WORK

Web traffic has been extensively characterized. Many studies concentrate on the user-level behavior such as the size and number of request/response messages, and Web application-specific properties such as page complexity and document referencing(e.g., [1, 2]). Flow-level properties of Web traffic have also been studied (e.g., [4, 16]). One key observation from prior work is that Poisson arrival process may not be appropriate for Web flows [4,16]. Our data reaffirms this observation, and also shows that Poisson models may not be appropriate for modeling P2P flow arrivals.

There are many studies of popular P2P systems in the literature, including Napster [19], KaZaA [8], Gnutella [19, 21, 26], BitTorrent [9, 10, 18], and eDonkey [17, 24]. These studies have focussed on different aspects of P2P systems such as query traffic [12], data traffic [8,21], flow characteristics [17,24], peer behavior [21], system architecture [9,10,18], and system dynamics (e.g., churn) [23, 26]. In this section, we discuss closely related prior work.

Saroiu *et al.* [19] studied Gnutella and Napster systems using traces collected using crawling techniques. They observed Gnutella hosts had high-bandwidth, high-latency, and low user-activity periods when compared to Napster hosts. Sen and Wang [21] studied DirectConnect, Gnutella, and FastTrack traces from a large ISP's network. They found that the traffic volume, peer connectivity, and mean bandwidth usage distributions are extremely skewed, which is similar to our observations. Recently, Zhao *et al.* [26] analyzed traffic from modern Gnutella systems. They observed a significant decrease in free-riders over the past few years. Our results, however, indicate pronounced free-riding in Gnutella. We believe free-riding needs to be further studied.

Guo *et al.* [9] analyzed and modeled BitTorrent systems based on traces collected from a popular tracker site. They found that swarm popularity decreases exponentially over time, and that the distribution of swarm population is heavily skewed. Pouwelse *et al.* [18] studied performance, robustness, and content integrity of BitTorrent systems.

Tutschku [24] and Plissonneau *et al.* [17] analyzed eDonkey traffic observed on the protocol's standard port. Tutschku found that eDonkey flow sizes follow the lognormal distribution, that flow IAT are exponentially distributed, and that eDonkey flows do not appear to alter the mice-elephant mix of flows. Similar to our observations, Plissonneau *et al.* found that eDonkey systems generates many short duration flows, have significant unfairness, and do not exploit geographic locality when exchanging data. Plissonneau *et al.* did not present any traffic models in their work.

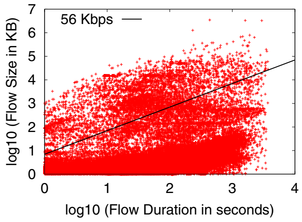


Figure 12: Flow size versus flow duration

Our study complements prior work on Web and P2P traffic analysis. We used recent traces that reflect the emerging traffic trends in a large edge network, and employed application signature matching to identify Web and P2P traffic accurately. We explored the similarities and differences in flow-level and host-level characteristics of Web and P2P flows, and developed models for both types of traffic.

6. DISCUSSION

In this section, we discuss two related issues: identification of P2P traffic and impact of network traffic management.

Many recent P2P characterization studies (e.g., [17, 21, 24]) have relied on identification by port numbers. Our full payload packet traces allow us to apply application signature matching to identify P2P traffic that would otherwise not be identified had we only relied on port numbers for traffic identification. We believe that future characterization of P2P traffic should not rely solely on port numbers for identification of this traffic. Because collection of traces with payloads poses unique challenges (e.g., processing cost, longer-term data collection) and are often difficult to obtain, alternative approaches are necessary. For example, recently proposed machine-learning techniques that use only flow statistics (see [6, 7] and the references therein) or heuristics-based techniques [5, 11] that leverage characteristic behavior of P2P applications may be suitable candidates for identifying P2P traffic.

A consequence of increased use of P2P applications is the deployment of bandwidth management solutions in edge networks. Any analysis of network traffic, therefore, needs to be aware of the potential implications of traffic management as some characteristics of interest such as flow duration and flow concurrency may be affected by flow management. At the University of Calgary, traffic is managed using a commercial packet shaping device. The packet shaper (to the best of our knowledge) employs a combination of application signatures and port numbers to identify traffic. At the time of trace capture, the network policy in place was to group together all *identified* P2P flows (except those from the student residences) and collectively limit their bandwidth to 56 Kbps. Figure 12 shows a scatter plot of P2P flow size and duration for our trace. The scatter plot includes a straight line that marks the 56 Kbps boundary; P2P flows (i.e., points) above this line represent an achieved flow throughput exceeding 56 Kbps. We should also note that points below this line do not necessarily imply that the flow's bandwidth was limited by the traffic shaping device. Flow rates may be below this line for other reasons such as multiplexing of flows, flow control, or congestion control mechanisms. The key observation from the plot is that we do not observe a strong positive correlation between flow size and duration. This suggests that some P2P flows are indeed identified and limited by the packet shaping device. Nevertheless, we do see many points above the 56 Kbps threshold; these P2P flows clearly escaped detection by the traffic shaper.

The final comment we make is regarding the representativeness of our observations and models. Our study is based on observations from one vantage point, and on a network that employs some form of bandwidth management. Clearly, there is a need to study traffic from different networks to validate the models we propose and also to develop general models for Web and P2P traffic. Nevertheless, we believe that our results are still useful as they provide a snapshot of Web and P2P traffic characteristics from a large edge network, and thus should be representative of other large edge networks with similar population and network management policies. In cases where the network differs significantly in design or management policy, our methodology can be applied to develop representative models.

7. CONCLUSIONS

This paper presented an extensive characterization of Web and P2P traffic using full packet traces collected at a large edge network. We considered three flow-level metrics, namely flow size, flow IAT, and flow duration, and three host-level metrics, specifically flow concurrency, transfer volume, and geographic distance. We observed a number of contrasting features between Web and P2P traffic. Typically, Web flows are short-lived whereas P2P flows are long-lived. Both Web and P2P host transfers are asymmetric; however, P2P host transfers are dominated by both upstream and downstream traffic, but not both. Web hosts maintain a high degree of flow concurrency, whereas many P2P hosts maintain a single flow at a time. Finally, P2P traffic exacerbates the "mice and elephants" phenomenon in Internet traffic. Flow-level distributional models were developed for Web and P2P traffic; these models can be used in network simulation and emulation experiments. We believe much work remains. Traffic from other networks should be studied to facilitate development of general models for Web and P2P traffic. Similarly, traffic from other non-Web applications, for example P2P streaming applications such as PPLive, P2P VoIP, and other P2P applications, should be examined, and their impact on Web-based applications studied.

Acknowledgements

The authors thank Jeffrey Erman, the anonymous WWW reviewers, *i*CORE, and NSERC.

8. REFERENCES

[1] M. Arlitt and C. Williamson. Internet Web Servers: Workload Characterization and Performance Implications. *ToN*, 1997.

[2] F. Campos, K. Jeffay, and F. Smith. Tracking the Evolution of Web Traffic: 1995-2003. In *MASCOTS*, 2003.

[3] B. Cohen. Incentives Build Robustness in BitTorrent. In *P2PECON*, 2003.

[4] M. Crovella and A. Bestavros. Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes. *ToN*, 1997.

[5] T. Dang, M. Perenyi, A. Gefferth, and S. Molnar. On the Identification and Analysis of P2P Traffic Aggregation. In *Networking*, 2006.

[6] J. Erman, A. Mahanti, M. Arlitt, I. Cohen, and C. Williamson. Offline/Realtime Traffic Classification Using Semi-Supervised Learning. In *Performance*, 2007.

[7] J. Erman, A. Mahanti, M. Arlitt, and C. Williamson. Identifying and Discriminating Between Web and Peer-to-Peer Traffic in the Network Core. In *WWW*, 2007.

[8] K. Gummadi, R. Dunn, S. Saroiu, S. Gribble, H. Levy, and J. Zahorjan. Measurement, Modeling, and Analysis of a Peer-to-Peer File-Sharing Workload. In *SOSP*, 2003.

[9] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang. Measurements, Analysis, and Modeling of BitTorrent-like Systems. In *IMC*, 2005.

[10] M. Izal, G. Urvoy-Keller, E. Biersack, P. Felber, A. Hamra, and L. Garcs-Erice. Dissecting BitTorrent: Five Months in a Torrents Lifetime. In *PAM*, 2004.

[11] T. Karagiannis, K. Papagiannaki, and M. Faloutsos. BLINC: Multilevel Traffic Classification in the Dark. In *SIGCOMM*, 2005.

[12] A. Klemm, C. Lindemann, M. K. Vernon, and O. P. Waldhorst. Characterizing the Query Behavior in Peer-to-Peer File Sharing Systems. In *IMC*, 2004.

[13] T. Mori, M. Uchida, R. Kawahara, J. Pan, and S. Goto. Identifying Elephant Flows through Periodically Sampled Packets. In *IMC*, 2004.

[14] E. O'Neill, B. Lavoie, and R. Bennett. Trends in the Evolution of the Public Web: 1998 - 2002. *D-Lib Mag.*, 2003.

[15] V. Paxson. Bro: A System for Detecting Network Intruders in Real-Time. *Com. Net.*, 1999.

[16] V. Paxson and S. Floyd. Wide-Area Traffic: The Failure of Poisson Modeling. *ToN*, 1995.

[17] L. Plissonneau, J. Costeux, and P. Brown. Analysis of Peer-to-Peer Traffic on ADSL. In *PAM*, 2005.

[18] J. Pouwelse, P. Garbacki, D. Epema, and H. Sips. The Bittorrent P2P File-sharing System: Measurements and Analysis. In *IPTPS*, 2005.

[19] S. Saroiu, P. Gummadi, and S. Gribble. Measuring and analyzing the characteristics of Napster and Gnutella hosts. *Multi. Sys.*, 2003.

[20] S. Sen, O. Spatscheck, and D. Wang. Accurate, Scalable In-Network Identification of P2P Traffic. In *WWW*, 2004.

[21] S. Sen and J. Wang. Analyzing Peer-to-Peer Traffic Across Large Networks. *ToN*, 2004.

[22] A. Soule, K. Salamatian, N. Taft, R. Emilion, and K. Papagiannaki. Flow Classification by Histograms: or How to go on Safari in the Internet. In *SIGMETRICS*, 2005.

[23] D. Stutzbach and R. Rejaie. Understanding Churn in Peer-to-Peer Networks. In *IMC*, 2006.

[24] K. Tutschku. A Measurement-Based Traffic Profile of the eDonkey Filesharing Service. In *PAM*, 2004.

[25] Z. Zhang, V. Ribeiro, S. Moon, and C. Diot. Small-time Scaling Behaviors of Internet Backbone Traffic: An Empirical Study. In *INFOCOM*, 2003.

[26] S. Zhao, D. Stutzbach, and R. Rejaie. Characterizing Files in the Modern Gnutella Network: A Measurement Study. In *MMCN*, 2006.