BitTorrent Traffic Characteristics

David Erman, Dragos Ilie and Adrian Popescu
Dept. of Telecommunication Systems
School of Engineering
Blekinge Institute of Technology
371 79 Karlskrona, Sweden
{*david.erman,dragos.ilie,adrian.popescu*}*@bth.se*

Abstract

This paper reports on a measurement and modeling study of session and message characteristics of BitTorrent traffic. BitTorrent is a Peer-to-Peer (P2P) replication and distribution system developed as an alternative to the classical client-server model to reduce the load on content servers and networks. Results are reported on measurement, modeling and analysis of application and link layer traces collected at the Blekinge Institute of Technology (BTH) and a local ISP in Sweden. Link layer traces and application logs were collected, modeled and analyzed using a dedicated measurement infrastructure developed at BTH to collect P2P traffic.

New results are reported on important session and message characteristics of BitTorrent, i.e., session interarrivals, sizes and durations, request rates and response times. Our results show that session interarrivals can be accurately modeled by a second-order hyper-exponential distribution while session durations and sizes can be reasonably well modeled by various mixtures of the Log-normal and Weibull distributions. Response times have been observed to be modeled by a dual Log-normal mixture, while request rates are modeled as dual Gaussian distributions.

1 Introduction

P2P file-sharing systems have evolved to be some of the major traffic contributors in the Internet [1]. Although an exact definition of "P2P systems" is still debatable, such a system typically represents a distributed computing paradigm where a spontaneous, continuously changing group of collaborating computers act as equals in supporting applications such as resource redundancy, content distribution, and other collaborative actions.

The most obvious and important consequence of the popularity of P2P systems is related to the high traffic volumes they cause. For instance, a recent measurement study showed that P2P traffic made up 80 % of the total traffic in a high speed IP backbone link [2].

BitTorrent, a P2P replication and distribution system, has become extremely popular over the last years. According to Cachelogic, the BitTorrent traffic volume increased from 26 % to 52 % of the total P2P traffic volume during the first half of 2004 [3]. BitTorrent use slightly decreased during 2005, due to the shutdown of several key BitTorrent sites, but BitTorrent traffic still makes out a substantial part of the total P2P traffic (up to 60 % in certain regions) [4]. However, measurement studies on BitTorrent are still fairly rare [5–7]. There are several reasons for this. P2P systems operate on the application layer. This means that either payload decoding or novel classification methods using, e.g., connection pattern characteristics, port usage or session characteristics are needed to indentify the P2P traffic [8]. Payload decoding brings up privacy issues as well as storage and retrival issues. In particular, application-layer message characteristics require storing the entire session payload, a difficult task when sessions may span several days in length and hundreds of GB in size.

The main goals of the paper are towards an understanding of the characteristics of BitTorrent sessions and messages, to be further used in a P2P simulation environment. To facilitate this, we have designed a dedicated measurement system to do traffic measurements on P2P systems [9]. Detailed results are reported on measurement, modeling and analysis of BitTorrent traffic collected at BTH, Karlskrona as well as at a local ISP on a 5 Mbps link. Our results show that session interarrivals can be accurately modeled by a second-order hyper-exponential distribution while session durations and sizes can be reasonably well modeled by various mixtures of the Log-normal and Weibull distributions. Response times are modeled by a dual Log-normal mixture, while request rates are modeled as dual Gaussian distributions.

The rest of the paper is organized as follows. Section 2 briefly describes the BitTorrent system, followed by an overview of the measurement infrastructure and measurements performed for this paper. In Section 4, we present the modeling methodology used in obtaining the models presented in Section 5. Finally, Section 6 concludes the paper and discusses future work.

2 BitTorrent

BitTorrent is a P2P protocol for content distribution and replication designed to quickly, efficiently and fairly replicate data [10]. It uses *swarming* to distribute load among participating peers. A BitTorrent network, also denoted as a *swarm*, does not provide any resource query, lookup, routing or topology forming functionality. The sole purpose of a BitTorrent swarm is to efficiently disseminate data. Consequently, the system is primarily used to distribute large pieces of content, such as operating system distributions or compressed video files.

A swarm consists of *peers* and at least one *tracker.* The role of the tracker is to provide peers with IP addresses of other peers; it does not participate in the content distribution. Peers are partitioned into *seeds* and *leechers*. A seed is a peer that is in possession of the entire content of the swarm, while a leecher is a peer that still needs to download parts of the content. An initial seed is necessary for a swarm to be successful.

The content distributed in a BitTorrent swarm is divided into *pieces*. Data is requested among peers using piece numbers and byte ranges in these pieces as identifiers. The byte ranges are also known as *subpieces* or *chunks*.

The leech phase of a BitTorrent peer may be partitioned into three different sub-phases. During the first sub-phase, the peer tries to connect to a predefined maximum number of peers. This means that the number of connected peers increases during this phase, thus increasing the number of outgoing piece requests as well. On entering the second phase, the peer has connected to enough peers. During this phase, the number of connected peers and outgoing messages fluctuate around some average value. The final phase is the *end-game mode*, which occurs when a peer only has a few pieces of the content left to download. To quickly obtain the last few pieces, requests are broadcasted to all connected peers, resulting in a dramatic increase in the upstream request rate. This behaviour can clearly be seen in Fig. 3. The phases are separated by dashed lines.

To join a specific BitTorrent swarm, a potential peer must first acquire a set of metadata, known as a *torrent file.* The torrent file contains, among other information, the address to the swarm tracker and a set of Secure Hash Algorithm One (SHA1) hash values for the content pieces. The SHA1 hash for the torrent file itself acts as an identifier of the swarm and is used in both peer handshakes and tracker queries.

Fairness in the BitTorrent system is implemented using a scheme in which peers express desire to download by sending *interested*-messages to peers that have needed pieces. The serving peers may allow or disallow download with *unchoke*- or *choke*-messages respectively. Data transfer takes place by the downloading peer issuing *request*-messages to randomly selected serving peers and the serving peers responding with *piece*-messages. Each *request–piece* message exchange refers to one subpiece. Once all subpieces are downloaded, a *have*-message is broadcasted to all connected peers.

3 Measurements

A mixed methodology for traffic measurements of P2P systems, based on application instrumentation and link layer packet capture, has been developed at BTH [9]. The solution allows accurate application layer measurements of P2P protocols, currently Gnutella and BitTorrent [9,11–14].

The P2P measurement infrastructure developed at BTH consists of peer nodes and protocol decoding software [15]. `Tcpdump` [16] and `tcptrace` [17] are used for traffic recording and protocol decoding.

The BTH measurement nodes run the Gentoo Linux 1.4 operating system, with kernel version 2.6.5. Each node is equipped with an Intel Celeron 2.4 GHz processor, 1 GB RAM, 120 GB hard drive, and 10/100 FastEthernet network interface. The network interface is connected to a 100 Mbps switch in the BTH networking lab, which is further connected through a router to the Swedish University Gigabit Network (GigaSUNET) backbone.

A number of 13 measurements were performed in 3 measurement sets at 2 locations. Measurements 1–8 (set 1) and 13 (set 3) were performed at BTH, while measurements 9–12 (set 2) were performed at a local ISP. Measurements 1–12 were collected as application logs from an instrumented version of the reference BitTorrent client. The data for measurement 13 was collected using both link layer packet capture and application logs. Due to hardware failure, the data for measurement 9 was irretrievably lost, which is why this measurement is absent from the presented models.

Measurement sets 1 and 2 were performed in May 2004, with each measurement ranging in duration from 2 days and 20 hours to 1 week, with some temporal overlap. Measurement set 3 was performed during 1 week in June 2004. In total, about 1.4 GB of compressed log files and 143 GB of packet traces were collected.

The measurements were performed by letting the measurement peer join the measured swarms as a participating peer. The swarms were already present and active when the measurement peer joined. To avoid copyright issues, the data content of the measured swarms was several Linux distributions, ranging in size from 650 MB to 4.3 GB.

4 Modeling methodology

4.1 Model Selection

We have opted to employ a heuristic procedure using several "Gaussian-like" distributions to identify the models for this work.[1] The distributions involved are the Gaussian, Weibull, Log-normal, a binary mixture distribution of Weibull and Log-normal and dual versions of the Gaussian and Weibull distributions. There is a certain amount of concensus in the networking community regarding using the Exponential distribution for session interarrivals and the Log-normal distribution for connection sizes and durations [18].

The general form for a binary mixture density is $f(x;\Theta_1,\Theta_2) = pf_1(x;\Theta_1) + (1-p)f_2(x;\Theta_2)$ where f_1 and f_2 are the component densities with associated parameters Θ_1 and Θ_2, and p is the *mixing weight*. By a dual distribution we refer to a binary mixture distribution, in which the component distributions are the same, i.e., $f_1(x) = f_2(x)$.

In this paper, we use the model notation shown in Table 1, in which λ_i are rates, μ_i and σ_i are means and standard deviations and α_i and β_i are shape and scale parameters.

Table 1. Model notation

Hyper-exponential	$H_2(\lambda_1, \lambda_2, p)$
Gaussian	$G(\mu, \sigma)$
Log-normal	$LN(\mu, \sigma)$
Dual Gaussian	$GG(\mu_1, \sigma_1, \mu_2, \sigma_2, p)$
Dual Log-normal	$LL(\mu_1, \sigma_1, \mu_2, \sigma_2, p)$
Dual Weibull	$WW(\alpha_1, \beta_1, \alpha_2, \beta_2, p)$
Log-normal – Weibull	$LW(\mu, \sigma, \alpha, \beta, p)$

The modeling procedure uses both the minimum distance and MLE methods for parameter estimations [14]. For each measurement, the following procedure is followed:

1. Use MLE to find initial parameter estimates for the Gaussian and Weibull distributions.

2. Minimize the estimation error using the minimum distance method (i.e., minimize over the error percentage described in Section 4.2).

3. Select the distribution and associated parameters yielding in the lowest $E_{\%}$.

4. If the minimum $E_{\%}$ is larger than 2 %, inspect the associated measurement data. If the optimisation algorithm fails to locate the global minimum, the starting values for the optimisation are modified manually.

[1]Except for the session interarrival times, which are modeled using Maximum Likelihood Estimation (MLE) and the Exponential distribution only.

4.2 Fitness Assessment

To determine the fitness of the selected models, we use an absolute fitness measure. The measure, denoted by $E_{\%}$, uses the Probability Integral Transform (PIT) method to transform the test for a specific distribution to a test for uniformity [14].

The error percentage is calculated by

$$E_{\%} = \frac{100}{nE_{max}} \sum_{i=1}^{n} |U_i - \hat{U}_i|,$$

where $E_{max} = \int_0^1 \sup\{U(x), 1-U(x)\}\,\mathrm{d}x = \frac{3}{4}$, $\hat{U}_i$ are the order statistics from the PIT transformed estimated distribution and U_i are the corresponding Uniform order statistics. It is important to mention that this is *not* a statistical significance level, but rather an acceptable margin of error. The error percentage is further discussed in [14].

The following levels are used to classify the degree of fitness for a specific distribution: $E_{\%} \approx (0,1,2,3,4,5) \rightarrow$ (excellent[E], very good[VG], good[G], fair[F], poor[P], fail).

5 BitTorrent Models

The following session and message characteristics have been modeled: session interarrival times, sizes and durations; request rates and piece response times. By session interarrival time, we refer to the time between successively started sessions as seen by the BitTorrent client. The session times have previously been presented in [11], as have measurements 1–3 and 10–13 of the session duration and size.

For the tables reported in this section, the hash-character (#) refers to the measurement number. Also, note that all Complementary Cumulative Distribution Function (CCDF) plots are on a $\log_{10}-\log_{10}$ scale, while Empirical Probability Density Function (EPDF) plots are on a linear scale. EPDF and CCDF x axes refer to the characteristic being modeled (e.g., session durations), while y axes refer to complementary distribution and density respectively. In the CCDF plots, the dashed line indicates the fitted model, and the solid line the CCDF of the measured data. For the EPDF plots, the solid thick line represents the fitted model. Also, to make the EPDF plots more clear, the upper tails have been cropped.

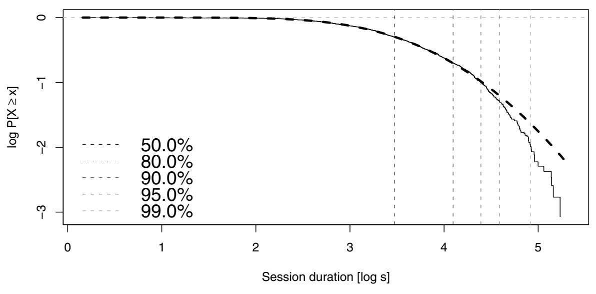
5.1 Session Interarrival Time

BitTorrent session interarrival times have been modeled by a second-order hyper-exponential distribution (Table 2).

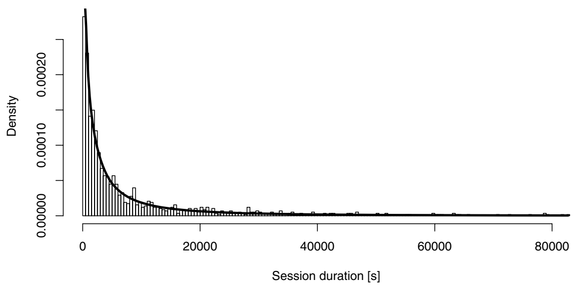
Table 2. Session interarrival modeling results

#	Model	$E_{\%}$
1	$H_2(5.93, 16.96, 0.22)$	2.07 %
2	$H_2(11.58, 75.56, 0.79)$	0.42 %
3	$H_2(5.66, 36.53, 0.66)$	0.49 %
4	$H_2(53.72, 1.68, 0.25)$	2.79 %
5	$H_2(55.38, 1.62, 0.22)$	2.79 %
6	$H_2(47.98, 1.27, 0.29)$	3.94 %
7	$H_2(41.88, 0.52, 0.30)$	2.05 %
8	$H_2(51.42, 1.68, 0.43)$	2.79 %
10	$H_2(55.80, 1.28, 0.33)$	3.76 %
11	$H_2(1.40, 8.02, 0.02)$	2.20 %
12	$H_2(9.30, 582.24, 0.83)$	3.85 %
13	$H_2(5.60, 41.75, 0.59)$	1.87 %

All 13 measurements pass according to the criteria described in Section 4.2. In particular, measurements 2 and 3 show very good fitness degrees. It is also interesting to note that the interarrival means ($\bar{H}_2 = \lambda_1/p + \lambda_2/(1-p)$) of measurements 1–3 and 10–13 are similar (8–14 ms), and that they are about 3–4 times shorter than the other measurements. This indicates that data in these swarms (The RedHat Fedora Linux distribution) is more popular than in the other swarms. Fig. 1 shows the session duration CCDF and EPDF with the associated model for measurement 6.



(a) CCDF



(b) EPDF

Figure 1. Session duration CCDF and EPDF for measurement 6

5.2 Session Duration During Seed Phase

It has previously been observed that, for all measurements, about 95 % of all sessions never transmit or receive any content [11]. Therefore, we have opted to only model sessions that transmit at least one sub-piece, i.e., 16 kB. The modeling results for session durations show a passing fitness degree using both a single Weibull and a single Lognormal distribution, except for measurement 12, which fails for the Weibull case. Table 3 presents the modeling results that resulted in the best fit.

It is interesting to note that the parameters for the Lognormal models are quite similar. One reason may be that that session durations are more of a swarm characteristic than the session interarrivals. This is an expected result, since network conditions have less impact on session durations than interarrival times (due to, e.g., load, bandwidth limiting, NAT or firewalling). However, it may also be related to the version of the unchoking algorithm used. The choking algorithm in the instrumented client tends to favor peers with high download rates, which means that the downloading peer will be allowed to download faster. However, this will also tend to lock out slower peers, forcing them to stay in the swarm longer.

Table 3. Session duration modeling results

#	Model	$E_{\%}$
1	$W(0.65, 5.10)$	1.49 %
2	$LN(8.18, 1.40)$	1.35 %
3	$LN(8.15, 1.51)$	1.31 %
4	$W(0.76, 5.00)$	0.92 %
5	$W(0.80, 5.49)$	1.34 %
6	$LN(8.02, 1.66)$	0.68 %
7	$W(0.71, 3.42)$	1.13 %
8	$W(0.37, 1.26)$	3.61 %
10	$LN(7.62, 1.73)$	1.82 %
11	$LN(8.07, 1.70)$	1.87 %
12	$LN(7.08, 1.69)$	3.15 %
13	$LN(7.93, 1.69)$	1.67 %

5.3 Upstream Session Size During Seed Phase

As with session durations, only sessions that transmit at least one sub-piece are selected for modeling.

BitTorrent session sizes have been observed to show varying degrees of correlation with session durations ($\rho = 0.25-0.67$) [11]. A certain degree of model similarity between the metrics is thus expected, but also indicate that there are long sessions that request little or no data, alternatively that short sessions transmit large amounts of data (the *mice* and *elephants* effect [18]). A similar effect, due to the BitTorrent choking algorithm is pointed to in [19], albeit during the leecher phase.

In Table 4, we report the modeling results for session sizes. As Table 4 shows, the similarity is mainly in the distribution types, i.e., the Log-normal, Weibull and mixtures of these. We note that measurement 12 fails the 5 % limit with a value for $E_{\%}$ of over 6 %.

Table 4. Session size modeling results

#	Model	$E_{\%}$
1	$LL(14.4, 2.89, 14.4, 3.23, 0.02)$	3.97 %
2	$LL(15.0, 2.61, 15.1, 3.07, 0.01)$	2.64 %
3	$LN(14.8, 3.24)$	3.02 %
4	$LL(15.5, 2.79, 15.8, 3.39, 0.15)$	2.97 %
5	$LL(15.3, 2.90, 15.6, 3.46, 0.05)$	2.42 %
6	$LN(14.6, 3.27)$	2.51 %
7	$LW(17.9, 36.2, 0.53, 4.18 \cdot 10^7, 0.95)$	2.82 %
8	$LN(14.0, 30.6)$	3.41 %
10	$W(0.42, 6.29 \cdot 10^6)$	3.79 %
11	$LN(13.8, 2.84)$	1.11 %
12	$LL(12.5, 2.65, 12.7, 3.98, 0.60)$	6.24 %
13	$LN(14.3, 3.35)$	2.99 %

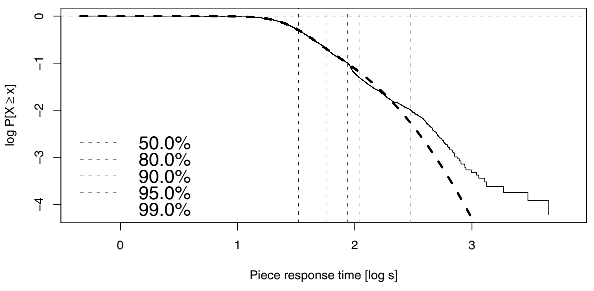
5.4 Piece Response Times

The piece response time is analogous to the piece download time for the measurement peer, i.e., the time between the measurement peer sending the first request for a piece until the *have*-message for the piece is sent by the measurement peer. A dual Log-normal distribution has been selected for the modeling. The results are presented in Table 5. Over 90 % of the measurements pass at a fitness degree of *Good* and higher, and almost 75 % at a degree of *Very good* or higher. Measurement 7 fails the 5% limit with a value for $E_{\%}$ of over 8%.

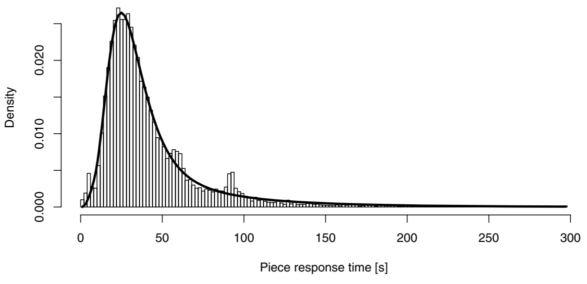
Fig. 2 shows the piece response time CCDF and EPDF for measurement 10. The upper tail in Figure 2(a) is indicative of the tail shape for all measurements. We note that there is a linear tendency, indicating that further analysis and modeling using a true heavy-tailed distribution, such as

Table 5. Piece response time modeling results

#	Model	$E_{\%}$
1	$LL(1.79, 0.48, 2.92, 0.48, 0.68)$	1.25 %
2	$LL(1.78, 0.40, 2.84, 0.61, 0.59)$	2.18 %
3	$LL(1.87, 0.50, 2.34, 0.70, 0.29)$	1.23 %
4	$LL(3.31, 0.53, 4.07, 0.95, 0.70)$	1.39 %
5	$LL(3.36, 0.30, 3.76, 0.88, 0.41)$	1.49 %
6	$LL(3.09, 0.26, 3.69, 0.78, 0.28)$	0.98 %
7	$LL(1.32, 1.40, 1.25, 1.39, 0.52)$	8.31 %
8	$LL(3.66, 0.40, 4.63, 1.38, 0.60)$	1.68 %
10	$LL(3.39, 0.41, 3.81, 0.84, 0.56)$	0.32 %
11	$LL(3.32, 0.38, 3.55, 0.75, 0.54)$	0.45 %
12	$LL(3.23, 0.63, 3.53, 1.08, 0.60)$	0.91 %
13	$LL(2.64, 0.88, 2.75, 0.80, 0.63)$	1.45 %



(a) CCDF



(b) EPDF

Figure 2. Piece response time CCDF and EPDF for measurement 10

a Pareto distribution, may yield further improvements in the model.

5.5 Upstream Request-rate During Leech Phase

The *request*-messages and their responses are the major bandwidth contributors in a BitTorrent session. It is therefore important to model the request behaviour of an individual peer.

A subset of the data was used for modeling the upstream leech phase request rates, to ensure that the end-game mode and startup-phase of the peer do not skew the model. The following censoring procedure was used to locate the steady-state portion of the request rate time series. The upper limit is set at 99 % of the time series, thus excluding the end-game mode requests [14]. To locate the lower limit, we calculate a smoothed version of the time series using a Lowess smoother [20]. The lower limit is then given by the location of the first minimum of the smoothed time series. Fig. 3 shows both the located censoring limits (dashed vertical lines) and the smoothed time series (solid gray line).

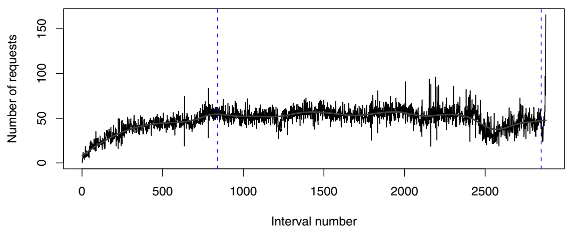


Figure 3. Request rates during leech phase for measurement 12

The resolution used for modeling the message *rates* is one second. Also, the number of back-to-back messages has not been modeled, i.e., messages with interdeparture times of 0 s are excluded from the models.

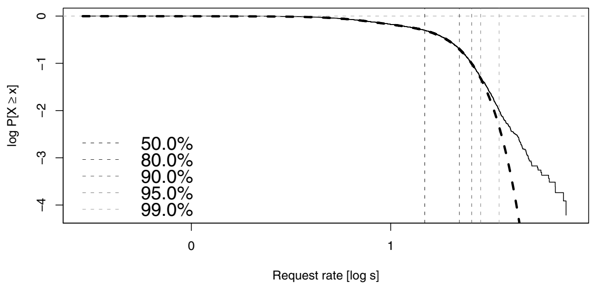
Table 6 reports the results of the request rate modeling. In certain cases, such as for measurement 11 (Figure 4(b)), the bi-modality of the dual Gaussian distribution is more pronounced than for the other measurements. The modes correspond to a changing number of outgoing requests due to an increase or decrease in number of connected peers.

6 Conclusions

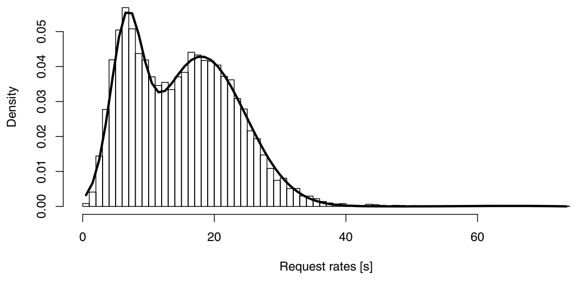
We have reported on a measurement and modeling study of BitTorrent. Results show that BitTorrent session interar-

Table 6. Request rate modeling results

#	Model	$E_{\%}$
1	$GG(87.3, 27.8, 88.4, 7.51, 0.16)$	0.59 %
2	$GG(76.7, 25.6, 78.5, 8.24, 0.28)$	0.44 %
3	$GG(95.2, 36.2, 112, 13.9, 0.35)$	0.88 %
4	$GG(16.0, 3.45, 20.6, 4.49, 0.68)$	0.44 %
5	$GG(13.2, 2.88, 17.7, 4.43, 0.74)$	0.43 %
6	$GG(9.71, 2.73, 14.8, 4.12, 0.35)$	0.45 %
7	$GG(34.1, 16.5, 43.7, 10.4, 0.39)$	1.58 %
8	$GG(7.78, 2.51, 20.0, 11.2, 0.93)$	0.84 %
10	$GG(10.0, 3.69, 22.9, 8.07, 0.84)$	0.81 %
11	$GG(6.58, 2.28, 17.9, 6.89, 0.26)$	0.31 %
12	$GG(11.1, 3.81, 29.1, 8.85, 0.68)$	0.80 %
13	$GG(48.2, 12.4, 53.8, 5.83, 0.38)$	0.57 %
13 [2]	$GG(33.8, 8.03, 38.0, 4.68, 0.47)$	0.36 %



(a) CCDF



(b) EPDF

Figure 4. Request rate CCDF and EPDF for measurement 11

rivals can be accurately modeled by a second-order Hyper-exponential distribution.

Session sizes are primarily modeled as a Log-normal distribution, and the Weibull distribution also provides a good model. Session durations are composed of mixtures of the Weibull and Log-Normal distributions, but show less accuracy than session duration models.

Piece response times are accurately modeled by a mixture of two Log-normal distributions, while request rates show similar dual behaviour with the Gaussian distributions.

Our future work is on modeling message-level characteristics on a per-flow basis as well as the number of requests per request message batch and verification of the presented models by further measurements.

References

[1] T. Karagiannis, A. Broido, N. Brownlee, K. Claffy, and M. Faloustos, "File sharing in the Internet: a characterization of P2P traffic in the backbone," University of California, Riverside, USA, Tech. Rep., 2003.

[2] N. B. Azzouna and F. Guillemin, "Experimental analysis of the impact of peer-to-peer applications on traffic in commercial IP networks," *European Transactions on Telecommunications: Special Issue on P2P Networking and P2P Services*, 2004.

[3] Cachelogic, "The true picture of peer-to-peer file sharing," http://cachelogic.com/research/p2p2004.php, 2004.

[4] ——, "P2P in 2005," http://cachelogic.com/research/p2p2005.php, 2005.

[5] M. Izal, G. Urvoy-Keller, E. Biersack, P. Felber, A. A. Hamra, and L. Garcés-Erice, "Dissecting BitTorrent: Five months in a torrent's lifetime," in *Passive and Active Measurements (PAM2004)*, 2004.

[6] D. Qiu and R. Srikant, "Modeling and performance analysis of bittorrent-like peer-to-peer networks," University of Illinois at Urbana-Champaign, USA, Tech. Rep., 2004.

[7] J. Pouwelse, P. Garbacki, D. Epema, and H. Sips, "The BitTorrent P2P file-sharing system: Measurements and analysis," *4th International Workshop on Peer-to-Peer Systems (IPTPS'05)*, February 2005.

[8] T. Karagiannis, A. Broido, M. Faloustos, and K. Claffy, "Transport layer identification of P2P traffic," *IMC'04*, 2004.

[9] D. Ilie, D. Erman, and A. Popescu, "Traffic measurements of P2P systems," *Swedish National on Computer Networking Workshop (SNCNW04)*, November 2004.

[10] B. Cohen, "BitTorrent," http://www.bittorrent.com/, March 2006.

[11] D. Erman, D. Ilie, and A. Popescu, "BitTorrent session characteristics and models," in *Technical Proceedings*, D. Kouvatsos, Ed. HET-NETs '05 - 3rd International Working Conference on Performance Modelling and Evaluation of Heterogeneous Networks, 2005.

[12] D. Erman, D. Ilie, A. Popescu, and A. A. Nilsson, "Measurement and analysis of BitTorrent traffic," in *Nodic Teletraffic Seminar (NTS) 17*, August 2004.

[13] D. Ilie, D. Erman, A. Popescu, and A. A. Nilsson, "Measurement and analysis of Gnutella signaling traffic," in *IPSI 2004*, September 2004.

[14] D. Erman, "Bittorrent traffic measurements and models," October 2005, licentiate thesis, Blekinge Institute of Technology.

[15] D. Erman, D. Ilie, and A. Popescu, "Peer-to-peer traffic measurements," Blekinge Institute of Technology, Karlskrona, Sweden, Tech. Rep., 2005.

[16] V. Jacobsen, C. Leres, and S. McCanne, "Tcpdump," http://www.tcpdump.org, August 2005.

[17] S. Ostermann, "Tcptrace," http://www.tcptrace.org, August 2005.

[18] V. Paxson and S. Floyd, "Why we don't know how to simulate the internet," in *Winter Simulation Conference*, 1997, pp. 1037–1044. [Online]. Available: citeseer.csail.mit.edu/paxson97why.html

[19] A. Legout, G. Urvoy-Keller, and P. Michiardi, "Understanding BitTorrent: An experimental perspective," INRIA Sophia Antipolis / INRIA Rhne-Alpes - PLANETE INRIA France, EURECOM - Institut Eurecom, Tech. Rep., November 2005.

[20] W. Cleveland, "Robust locally weighted regression and smoothing scatter plots," *Journal of American Statistical Association*, vol. 74, pp. 829–836, 1979.