2005

Characterization of Web server workload

Amit Sangle
West Virginia University

Characterization of Web Server Workload

by

Amit Sangle

Thesis submitted to the
College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements
for the degree of

Master of Science
in
Computer Science

Dr. Katerina Goseva Popstojanova, Ph.D., Chair
Dr. Jagannathan Vasudevan, Ph.D.
Dr. Hany Ammar, Ph.D.

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia
2005

Keywords: Self-similarity, heavy-tailed distribution, web server workload characteristics

Abstract

Characterization of Web Server Workload

by

Amit Sangle
Master of Science in Computer Science

West Virginia University

Dr. Katerina Goseva Popstojanova, Ph.D., Chair

Realistic and formal mathematical description of web-server workload forms a fundamental step in the design of synthetic workload generators, capacity planning and accurate predictions of performance measures. In this thesis we perform detailed empirical analysis of the web workload by analyzing access logs of nine web-servers. Unlike most previous work that focused on request-based workload characterization, we analyze both request and session characteristics. We perform rigorous statistical analysis to determine the self-similarity of web traffic and heavy-tailedness of the distribution of different session parameters. Our analysis shows that web traffic is self-similar and the degree of self-similarity is proportional to the workload intensity. To increase the confidence in our analysis we use several methods for estimating the degree of self-similarity and heavy-tailedness. Additionally we point out specific problems associated with these methods. Finally, we analyze the impact of robots sessions on the heavy-tailedness of the distribution.

Acknowledgments

I would like to express my gratitude to my advisor Dr. Katerina Goseva Popstojanova for her support and guidance through my thesis. I am also grateful to my other committee members, Dr. Jagannathan and Dr. Hany Ammar for their support.

I would like to thank David Krovich, Lane Department of Computer Science, WVU, David Olsen, WVU Web Services and Brian Kesecker, NASA IV & V Facility for making available the web server logs that were crucial for my research. Thanks to the NASA IV & V Facility, Fairmont, West Virginia for providing financial support for this research. Special thanks to my lab colleagues Fengbin Li , Deepak Jha, Xuan Wang and Ajaydeep Singh for helping me in my research.

Finally, I would like to thank my family and friends for their constant help and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Since the advent of World Wide Web (WWW) in the early 90's there has been an exponential growth in the number of people using and relying heavily on WWW. Different web servers such as Apache, Microsoft IIS, IBM Http Server (IHS) play an important role in enabling Internet users to access information, download data and use web-based applications. With the number of users accessing these web servers continuously growing, factors such as performance, scalability and high availability are prime concerns for organizations hosting web-sites so as to ensure satisfactory Quality of Service (QoS) to the end users. Web users assume that the web-sites they want to access will be available 24/7 and would be performing at satisfactory levels (response time less than few seconds) all the time. Failure to meet these high QoS expectations can result in loss of business directly impacting the profits. It is estimated that the economic loss because of unavailability due to failures or poor performance is in the range of billions of dollars per year in United States alone [4]. In addition to traditional web-based applications, web technology is also used in mission critical applications where reliability and performance become all the more critical because of the real-time needs of such application. Analyzing and predicting the web quality, thus, is very important considering the high consequences of poor QoS.

Key to understanding the web quality is, understanding the characteristics of workload on the web servers. This understanding helps to:

1. Improve server performance - Lots of studies have been studying the impact of web workload

characteristics on the performance. It has been shown that, design of web caching systems [5], network congestion management and detection systems [6] can be improved based on thorough understanding of the workload characteristics.

2. Perform capacity planning - Efficient prediction of workload can help provision additional resources during peak loads.

3. Administer and manage system resources - Maintenance activities such as system scanning, backup's etc. can be performed at non-peak hours.

4. Provide personalized service to users - Understanding the nature of user session characteristics can help provide personalized service to them.

5. Design web workload generators accurately - Synthetic workload generators simulate workload on the web system. One issue with this kind of workload generators is that they can be far away from being realistic. Knowing the characteristics of the workload can help making the synthetic workload closer to the actual workload.

6. Do admission control - Understanding user session behavior can be used to positively impact admission control by dropping unimportant requests.

Lot of empirical studies have been done to understand the nature of web traffic [7, 8, 4, 5]. However, given the diversity of web traffic and rapid development in web technology more up-to-date workload studies are necessary. In this thesis, we characterize the web traffic by analyzing web-access logs from nine different web servers (including three private web servers). Traditionally, the analysis of web servers considers request as the basic unit of analysis and study of characteristics such as bytes transferred per request, inter-arrival time between requests and number of requests per unit time is done. In this thesis, in addition to some of the request-based characteristics, we also study session characteristics. A session represents the interaction of the web-system with a single user. We deem that the study of user session characteristics can be equally or even more useful. We therefore analyze the user session in terms of following session characteristics:

1. *Intra-session characteristics* - bytes transferred within a session, number of requests per session, session length.

2. *Inter-session characteristics* - Number of sessions initiated per unit time, Time between session initiations.

1.2 Motivation and Research Objective

The long-held paradigm in the communication and performance communities has been that the network traffic can be described by certain *Markovian models* (e.g. Poisson process) [1]. These models are subject to accurate analysis and have an efficient control mechanism. The very popular queuing theory is based on these models. However, the seminal study conducted by Leland, Taqqu, Willinger and Wilson [1] in 1994 showed *self-similar* (scale-invariant burstiness) nature of the Ethernet traffic. Since then there has been a considerable interest in the notion of *'self-similarity - long range dependence'* and its application to the network traffic. Numerous empirical studies [5, 7, 8, 4, 9] and simulations [10, 8, 11] were focussed on the study of self-similar nature of the network traffic. Self-similarity of Ethernet traffic has been attributed to the heavy-tailed distribution of file-sizes within a web server [5] and is known to adversely affect the web server performance. A few years down the line, Karagiannis, Faloutsos and Riedi [12], raised certain doubts on the existing methods to estimate self-similarity. They showed that different methods for self-similarity estimation produced conflicting results. Further, they also proved that on the network backbone packet arrivals follow Poisson assumption in their study [13]. In their paper titled *Long Range Dependence: Ten Years of Internet Traffic Modeling* they say:

"As the Internet increases in size and the technologies connected to it change, we must constantly monitor and reevaluate our assumptions to ensure that our conceptual models correctly represent reality"

This along with the conflicting results produced by different studies forms one of the most important motivations to study the characteristics of web traffic.

The aim of the thesis is:

1. To report the characteristics of web workload that have significant performance implications.
2. To perform statistical analysis on request and session characteristics of the workload and evaluate whether the workload is indeed self-similar in nature and shows heavy-tailed distribution.

3. If heavy-tails are present in the workload, what are the possible reasons for this kind of distribution?

4. Make remarks on the existing methods for determining self-similarity and heavy-tailed distribution.

Chapter 2

Background

The great book of nature is written in a mathematical language - *Galileo Galilei*

Network traffic is widely believed to be self-similar in nature. In this chapter we will look at the theory associated with *self-similarity*, *long-range dependence*, and *heavy-tailed distribution*.

2.1 Fractal Geometry and Self-similarity

Science and geometry have always progressed hand in hand. Scientists have always tried to explain natural phenomena's using perfect geometrical shapes such as squares, circles, ellipse etc. For example, earth revolves around the sun in *elliptical* orbit; similarly back and forth motion of a perfect pendulum is represented by a *sine wave*. These natural systems follow deterministic laws of physics and the future of such systems can be predicted from the past. However, there are many simple systems in the universe that follow deterministic laws but still behave unpredictably - the deterministic chaos as described in [14]. Phenomena's such as: the shape of mountains or clouds, how galaxies are distributed in the universe, the way prices vary in stock market cannot be explained by simple geometry of squares, triangles and circles. They need a far more complex geometry. As the need to find answers to such complex phenomena's grew the concept of fractal geometry evolved. B. B. Mandelbrot first pioneered the notion of self-similarity by observing the fractal geometry in nature [14].

Fractals are geometrical shapes that are not regular. They show the same degree or irregularity on all scales. A fractal object looks the same when observed from a long distance or nearby. In other

words, it is self-similar in nature. Self-similarity, thus, can be defined as a phenomenon wherein objects or mathematical forms (e.g. fractals) exhibit features that appear similar in different scales of magnification.

Nature has many objects that are self-similar: broccoli, cauliflower, ferns and many other plants exhibit self-similarity.

One way to explain self-similarity is by using a Sierpinski gasket [15] shown in Figure 2.1.

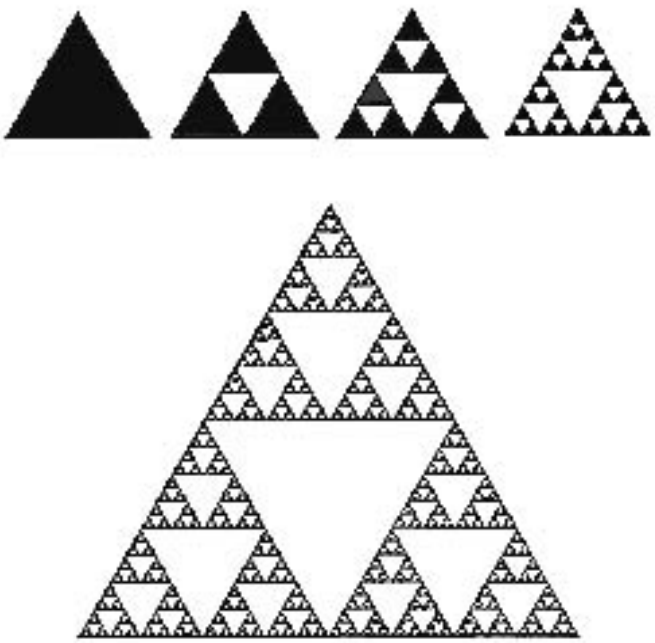

Figure 2.1: Depicting self-similarity - Sierpinski gasket

The Sierpinski gasket can be constructed as follows:

1. Start with a solid equilateral triangle and create four smaller triangles using midpoints of the three sides of the original triangle as the vertices of the new triangle.

2. Remove the interior of the middle triangle. This will form three solid triangles and one white triangle. Each of these new triangles will be an equilateral triangle with side half of the original triangle.

3. Repeat step 2 for each of the solid triangles formed to obtain a similar structure on reduced scale. Continue to do so for all the subsequent triangles formed.

You will see a structure similar to the one seen in Figure 2.1 wherein parts of the objects are *exactly* like the whole - but only in a different scale. This is called "linear" or "deterministic" self-similarity.

2.2 Stochastic Self-similarity and Network Traffic

In the previous section we have looked at linear or deterministic self-similarity that assumes strong recursive regularity. It has been shown that web traffic is self-similar in nature. However, it would be too much to expect "linear" self-similarity in network traffic. Network traffic is random in nature. The stochastic variability in network traffic can be attributed to:

1. Different number of users accessing the network at different time of day.
2. Users accessing data (web-pages, documents etc.) of random size randomly form a particular web-site etc.
3. "Think time" of user accessing particular web-site.

The stochastic (non-linear) self-similarity of network traffic is illustrated visually in Figure 2.2. The figure shows the traffic data from [1] in terms packets per unit time, plotted in five different time scales. Starting with a time unit of 100 sec in (a), each subsequent graph is plotted by increasing the time resolution by a factor of 10 and by concentrating on a randomly chosen subinterval (indicated by a darker shade).

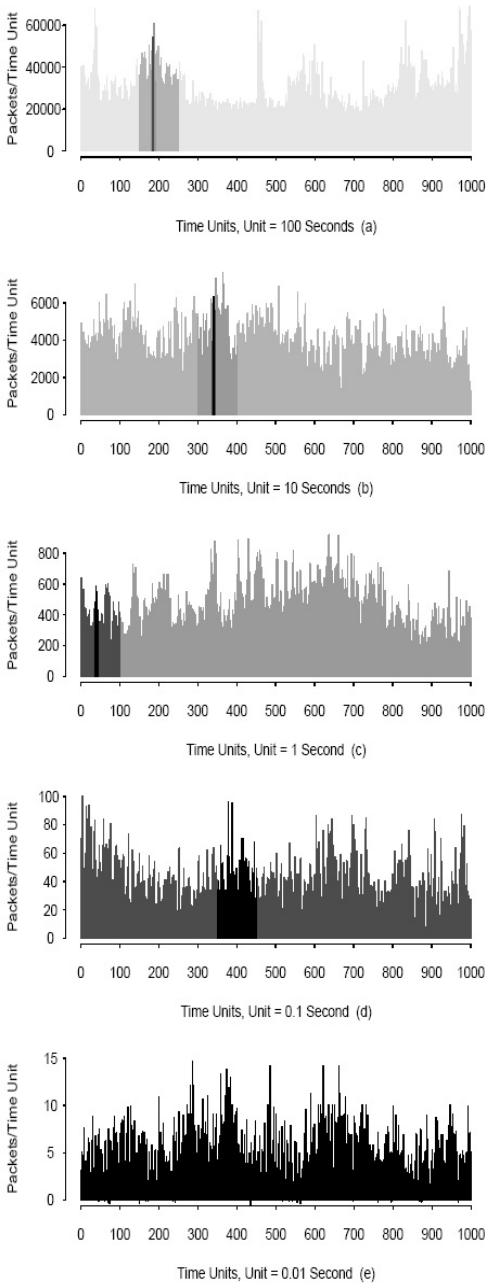

Figure 2.2: Visual proof of Self-similarity, adopted from [1]

Observe that, unlike the deterministic self-similarity, these plots do not have exact resemblance to each other. However, when suitably normalized, we can intuitively say that the graphs look very 'similar' to each other as far as the shape is concerned. Notice that, there is no natural length of bursts, i.e., for every time scale we can see bursty sub-periods separated by less bursty sub-periods.

In contrast, Poisson process, the commonly assumed model for network traffic, has a characteristic burst length, which would smoothed by averaging over a long enough time scale [16]. This is not seen in Figure 2.2.

2.3 Statistics of self-similarity

Figure 2.2 shows the visual proof of self-similar nature of network traffic. In this section we present the statistical aspects of a self-similar process.

2.3.1 Continuous time definition

A statistical process *X(t)* is said to be self-similar with a parameter *'H'* (or H-self similar), if for any real $a > 0$, the process *X(at)* has the same statistical properties as $a^H X(t)$

$$X(at) = a^H * X(t), a > 0 \tag{2.1}$$

Here, a denotes the scaling factor

H, the Hurst Exponent, indicates the degree of self-similarity

Thus, self-similarity implies that a change of the time scale is equivalent to change in state space scale. For a self-similar process $0.5 < H < 1.0$. As H increases from 0.5 to 1.0, the degree of self-similarity increases.

2.3.2 Discrete time definition

Consider a discrete time stochastic process or a time series $X(t), t \epsilon Z$. Here, $X(t)$ represents the traffic volume in terms of number of bytes, number of packets or number of requests per unit time. Let, $X_k(t), t \epsilon Z$ denote the k-shifted process. $X(t)$ is *strictly stationary* if $(X(t_1), X(t_2), ..., X(t_n))$ possess the same joint distribution as $(X(t_1 + k), X(t_2 + k), ..., X(t_n + k))$. Imposing strict stationarity however is too restrictive and we will be interested in *second-order stationarity*, which requires that the autocovariance function $\gamma(r, s) = E[(X(r) - \mu)(X(s) - \mu)]$ satisfies the translational invariance:

$$\gamma(r, s) = \gamma(r + k, s + k)$$

for all $r, s, k \epsilon Z$

This implies that the k-shifted process has the same covariance as original process. Since, by stationarity $\gamma(r, s) = \gamma(r - s, 0)$, we denote autocovariance by $\gamma(k)$. Also, note that the autocorrelation function $r(k)$ is given by the equation $r(k) = \gamma(k)/\sigma^2$, where σ^2 denotes the variance.

To formulate the scale invariance, let us now define the m-aggregated series $X^{(m)} = (X_k^{(m)}; k = 1, 2, 3,)$ by summing the original series $X(t)$ over non-overlapping blocks of size m.

$$X^{(m)}(i) = \frac{1}{m} \sum_{t=m(i-1)+1}^{mi} X(t) \tag{2.2}$$

Then we say that X is H-self-similar if for all positive m, $X^{(m)}$ has the same distribution as X rescaled by m^H.

$$X(t) =_d m^{-H} X^{(m)} \tag{2.3}$$

Considering, second-order stationarity, we can also say that, if X is H-self-similar, it has the same autocorrelation function $r(k)$ as the aggregated series $X^{(m)}$.

2.4 Long-range Dependence

Long-range dependence and self-similarity have been used interchangeably especially in the network traffic model study. A process is said to be long-range dependent if its autocorrelation function exhibits a power-law decay i.e. $r(k) \sim k^{-\beta}$ as $k \to \infty$, where $0 < \beta < 1$.

Power-law decay is slower than exponential decay. Thus, the autocorrelation function decaying as a power-law function, essentially means that there is a stronger correlation between the process and its time shifted version (as compared to exponential decay). Since $\beta < 1$, the sum of autocorrelation values of such series approaches infinity. The implication of this nonsummable autocorrelation is that, if we consider n samples from the series, then the variance does not decrease proportional to $1/n$ buy rather decreases proportionally to $n^{-\beta}$. For short-range dependent process the autocorrelation function is summable.

2.5 Estimating H - Hurst Exponent

In this section we will shortly discuss various existing methods for estimating the value H, the Hurst Exponent. The details of these methods can be found in [17] and [18].

1. *Absolute Value Method* - In this method we plot the aggregation level versus the absolute first moment of the aggregated series $X^{(m)}$ on a log-log scale. This plot is a straight line with slope of $H - 1$, if the data is process is long-range dependent.

2. *Variance Method* - In this method plot the aggregation level versus the sample variance of the aggregated series on a log-log scale. If the series is self-similar with long-range dependence then the plot is a line with slope $\beta = 2(H - 1)$, where $\beta > -1$.

3. *R/S method* - This method uses the rescaled range statistic (R/S statistic). The R/S statistic is the range of partial sums of deviations of a time series from its mean, rescaled by its standard deviation. For a self-similar process, the log-log plot of R/S statistic versus the number of points of aggregated series should be a straight line with slope H.

4. *Periodogram* - This method is based on the power spectrum transforms. In this method we plot the spectral density of the time series versus the frequencies on a log-log scale. The slope provides the estimate of H.

5. *Whittle estimator* - The Whittle estimator is also based on periodogram. It is a non-graphical method which produces an estimate of H with a confidence interval. The method is based on the minimization of a likelihood function, which is applied to the periodogram of the time-series.

6. *Variance of Residuals* - In this method we plot the aggregation level versus the average of the variance of the residuals of the series on a log-log scale. For a long-range dependent series the plot is a straight line with slope $H/2$.

7. *Abry-Veitch method* - This is a wavelet based method for estimation of H [18]. In this method the energy of the series in various scales is studied to estimate the value of H.

Karagiannis, Faloutsos, Molle [2] developed a java based tool - SELFIS for estimating the value of H using the above methods. Sample output of the tool showing the estimate of H using the above methods is shown in Figure 2.3.

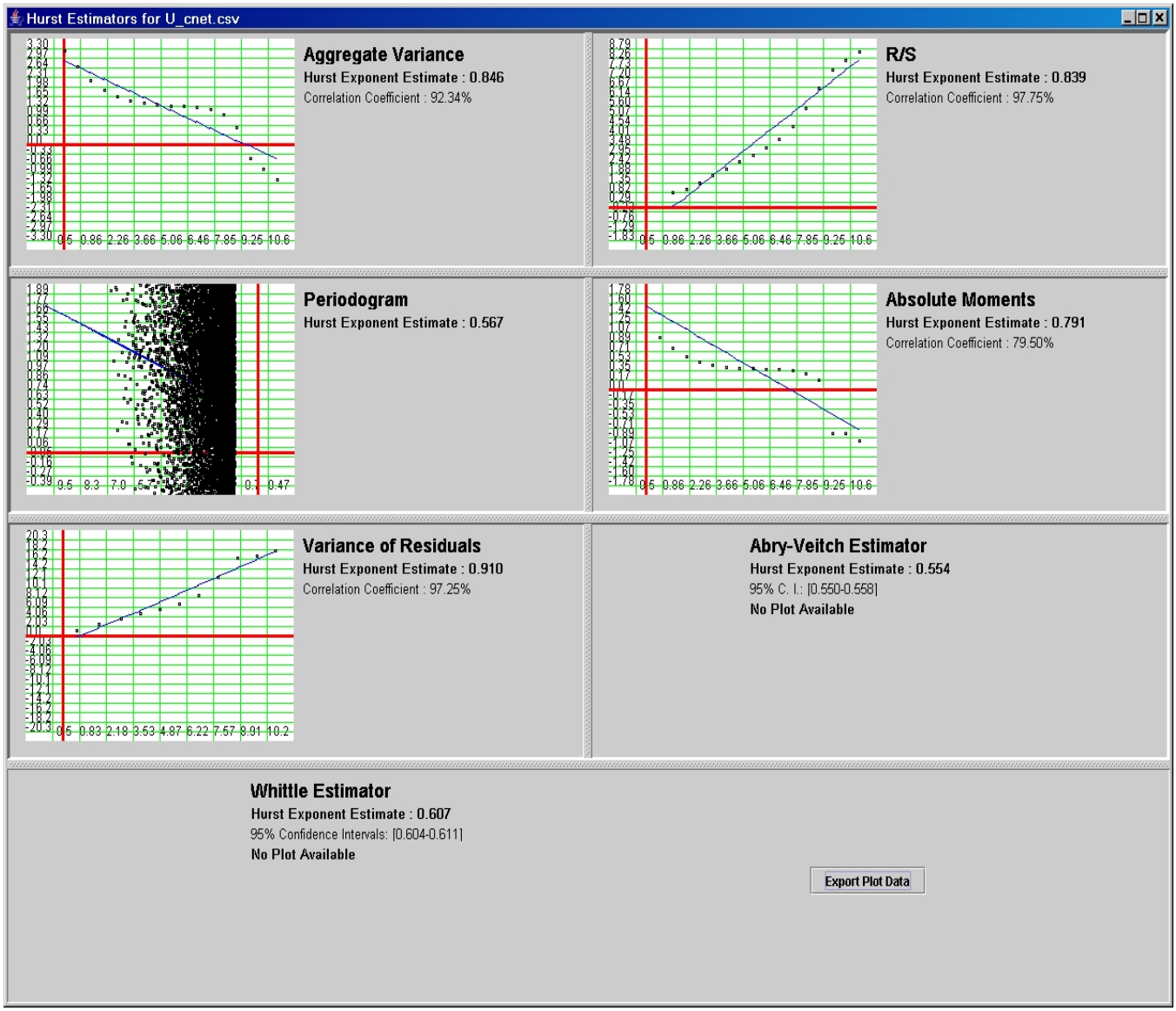

Figure 2.3: Sample Output from Selfis - Hurst Exponent estimators [2]

2.6 Heavy-tailed Distribution

There is an intimate relation between long-range dependence and heavy-tailed distributions. A distribution is said to be heavy-tailed if its Complementary Cumulative Distribution function (CCDF) is given by the equation:

$$P[X > x] \sim cx^{-\alpha}, x \to \infty, 0 < \alpha < 2 \tag{2.4}$$

Here, α is called the tail index of heavy-tailed distribution.

This means that, regardless of the behavior of small values of the random variable, if the asymptotic shape of the distribution is hyperbolic, it is heavy-tailed. This is in contrast with, light-tailed distributions (exponential and Gaussian distribution, for example) which possess an exponentially decreasing tail.

The simplest heavy-tailed distribution is the *Pareto* distribution. The Pareto distribution is hyperbolic over its entire range. The probability density function for Pareto distribution is given by:

$$p(x) = \alpha k^{\alpha} x^{-\alpha-1}, \alpha,\ k > 0, x \geq k$$

and its cumulative distribution function is given by

$$F(x) = P[X \leq x] = 1 - (k/x)^{\alpha}$$

The parameter, k represents the smallest possible value of the random variable.

Heavy-tailed distributions have number of properties that are different from exponential distributions. For $0 < \alpha < 2$, the heavy-tailed distribution has infinite variance and if $0 < \alpha \leq 1$ they also have infinite mean. This means that as α decreases, an arbitrarily large portion of the probability mass may be present in the tail. In other words as α decreases, the distribution becomes more heavy-tailed. In practical terms, a random variable that follows heavy-tailed distribution can give rise to extremely large values with non-negligible probability [16].

2.7 Estimating the index of heavy-tailed distribution - α

In this section we discuss two methods for estimating the value of α, the index of heavy-tailed distribution.

2.7.1 Log Log Complimentary Distribution (LLCD) plot

Let, $\overline{F}(x)$ denote the complimentary cumulative distribution function as given in equation 2.4. Taking logs on both sides of this equation we get:

$$\frac{d\log(\overline{F}(x))}{d\log(x)} = -\alpha, x > k$$

Thus, if we plot the complimentary cumulative distribution function on a log-log scale, we expect a straight line, in the tail ($x > k$).

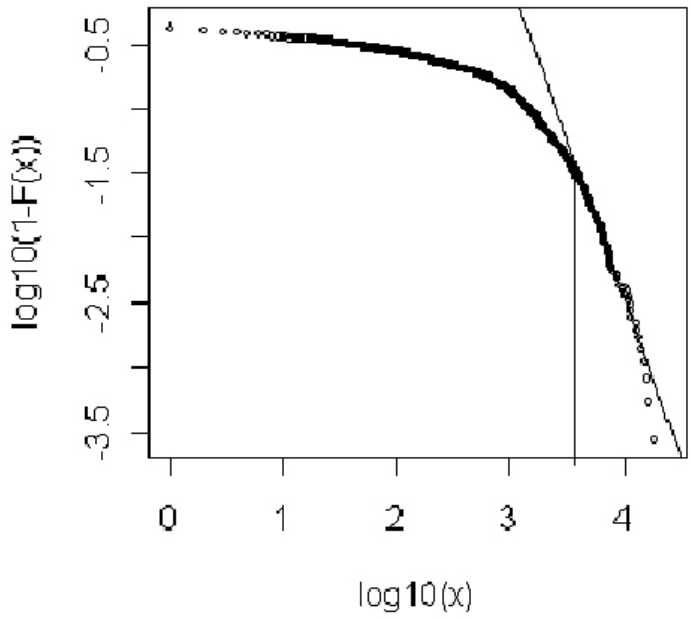

Figure 2.4: Sample Log-Log Complimentary Plot (LLCD) [3]

Figure 2.4. shows a sample Log-Log Complimentary Distribution (LLCD) plot. As seen in the figure the plot appears linear after $x > 3.6$. A linear regression fit to the points after $x > 3.6$ (i.e. a linear regression fit to the tail of the distribution) gives a line with slope -2.47, i.e. $\alpha = 2.47$. The R^2 value of the regression indicates the goodness of fit and it is 0.9 for the above sample plot.

2.7.2 Hills Plots

Another plotting technique to estimate the index of heavy tails is based on the Hills estimator. The Hills estimator is defined as follows:

$$H_{k,n} = \frac{1}{k}\sum_{i=1}^{k} \log \frac{X_{(i)}}{X_{(k+1)}} \tag{2.5}$$

where,

$X_{(i)}, i = 1, 2, 3..., n$ is the distribution under consideration, such that,

$$X_{(1)} > X_{(2)} > X_{(3)} > ... > X_{(n)}$$

k is the number of upper-order statistics used in the estimation.

The rough idea behind using k upper-order statistics is that we want to sample only that part of the distribution which looks most Pareto-like [19].

For getting the Hills plot, we calculate the value of $H_{k,n}$ by varying the value of k from 1 .. $n-1$ and plot:

$$\{(k, H_{k,n}^{-1}), 1 \leq k < n\}$$

Figure 2.5 shows a sample Hills plot drawn using the equation 2.5. As we see from the plot, the value of $H_{k,n}^{-1}$ stabilizes at 1. This stable value of $H_{k,n}^{-1}$ is the value of α, the index of heavy-tailed distribution.

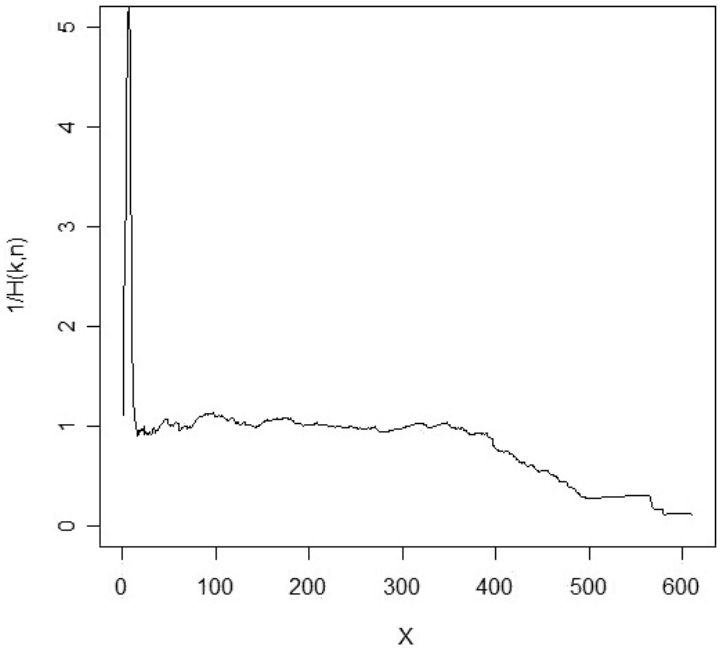

Figure 2.5: Sample Hills plot, $\alpha = 1.0$

Smooth Hills plot, Alternate Hills Plot and Alt Smooth Hills plot

Figure 2.6 (a) shows a Hills plot where it is difficult to estimate the exact value of α. Three techniques which ease the estimation of α: smoothing the Hills estimator, alternate Hills plotting (changing the scale) and combination of smoothing and alternate hill plot are suggested in [19]. Figure 2.6 (b) (c) and (d) shows the *Smooth Hills plot*, *Alternate Hills plot* and the *Alternate Smooth Hills plot*.

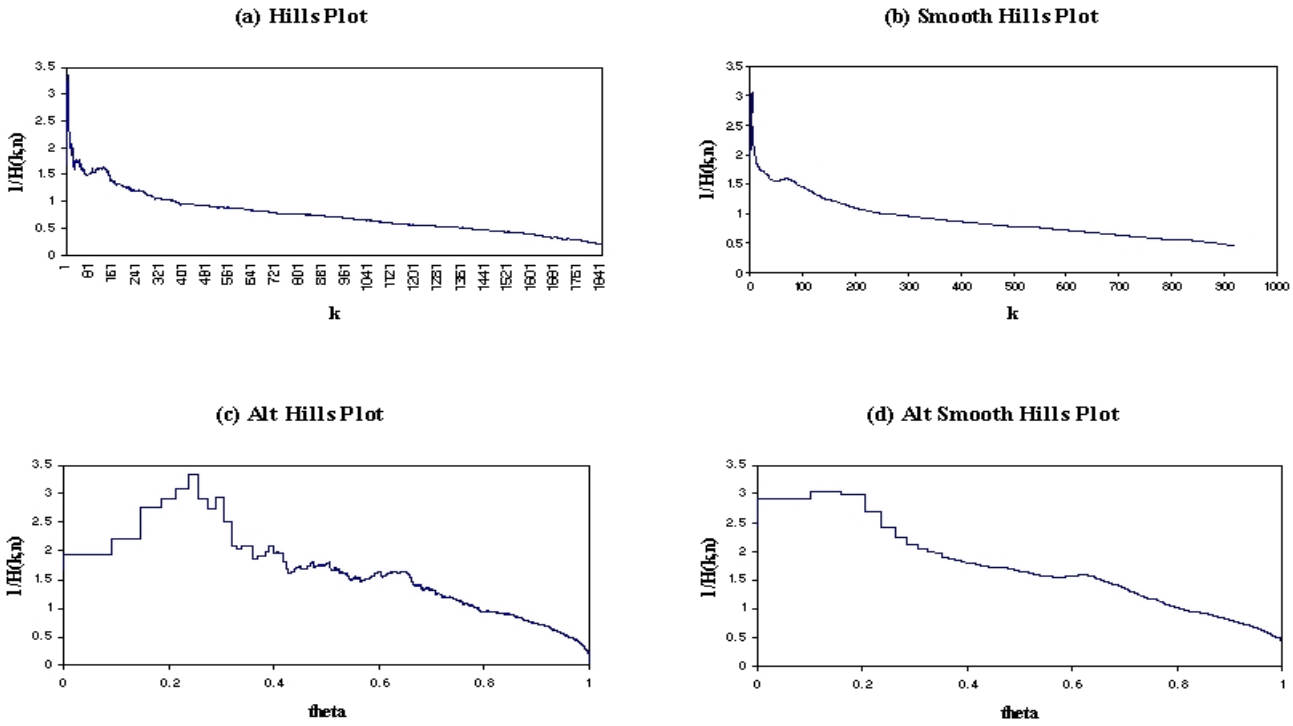

Figure 2.6: (a) Hills Plot (b) Smooth Hills Plot (c) Alternate Hills Plot (d) Alternate Smooth Hills Plot, $\alpha = 1.55$

Smoothing reduces the volatility of the plot and the uncertainty about how to pick the value of α. The smooth Hills estimator is calculated using the formula:

$$SmoothH_{k,n} = \frac{1}{(u-1)k} \sum_{j=k+1}^{uk} H_{j,n} \tag{2.6}$$

The bigger the value of u, more is the variance reduced. However, there is a tradeoff between reduction in the variance and the fact that for bigger values of u, the number of points in the smooth hills plot reduce [19].

In the *Alternate Hills* plot, the same information in the Hills plot is plotted on a different scale. We plot the following graph:

$$\{(\theta, H^{-1}_{\lceil n^\theta \rceil, n}, 0 \leq \theta \leq 1)\}$$

This graph is shown in Figure 2.6 (c). As we see, the initial order statistics get shown more clearly and cover a bigger portion of the displayed space.

The Figure 2.6 (d) shows the *Alternate Smooth Hills* plot, where we plot the Smooth plot (b) on the alternate scale.

From the alternate Hills plot and Alternate Smooth Hills plot we can now clearly see that stabilization of $H^{-1}_{k,n}$ occurs at 1.55, giving us an estimate of α, the index of heavy-tailed distribution.

Chapter 3

Related Work and Our Contribution

Web workload characterization is a widely researched field and and the results of lots of studies characterizing web clients and web servers have been published. In this chapter we review the related work and also put forward our contribution.

3.1 Review of Related Work

3.1.1 Workload Characterization of LAN and WAN Traffic

Since the pioneering study done by Leland, Taqqu, Willinger and Wilson [1] there have been numerous studies on measurement-based traffic modeling [19, 20, 21, 22] where researches have collected traffic traces from actual networks and analyzed them to identify and quantify the traffic characteristics. In this section we review the workload characterization of LAN and WAN traffic.

Leland, Taqqu, Willinger and Wilson [1] in their pioneering work established in a statistically rigorous manner that Ethernet (LAN - Local Area Network) traffic is self-similar in nature and that the degree of self-similarity depends on the level of utilization of the network. Higher network utilization results in higher degree of self-similarity [1]. They analyzed four sets of traffic data for a period ranging from August' 89 to February' 92, each period consisting of 20 to 40 hours of continuous Ethernet traffic. They studied the Ethernet characteristics by analyzing the number of packets in the network per unit time. From the large 20 to 40 hour traffic data for each of the periods they took representative periods indicating low, medium and high Ethernet traffic. For each of these periods they estimated the value of the Hurst Exponent 'H' using R/S method, variance time plot and periodogram plot. They found that the value of 'H' lied in between 0.5

and 1.0 indicating self-similarity of the Ethernet traffic and also found that degree of self-similarity increased from low to high period. Further more, even after aggregation of traffic over longer time intervals the self-similar nature of traffic was preserved (as shown by $0.5 < H < 1.0$) indicating that the assumption of Poisson model for Ethernet traffic was flawed and that aggregation intensifies the burstiness as opposed to making it smooth.

Paxson, Floyd [22] later analyzed 24 traces of wide-area TCP traffic, investigating various arrival processes. In their study, they showed that for interactive TELNET traffic and FTP traffic, connection or session arrivals can be well-modeled as Poisson process with fixed arrival rates. However, TELNET packet arrival and FTP data connections arrival within a single FTP session show significant burstiness. Further, they show that the distribution of number of bytes in each FTP session has a heavy tail i.e. small fraction of the bursts carry most of the FTP data transfer bytes. They also come up a complete model (FULL-TEL) for TELNET traffic, which uses Poisson connection arrivals, log-normal connection sizes (in packets) and Tcplib [11] packet inter-arrivals. They show that, FUL-TEL faithfully represents the actual TELNET workload by comparing the results of synthetically generated workload using FUL-TEL with actual TELNET traffic traces. Further, they look into SMTP and NNTP connection arrivals are show that this traffic cannot be modeled as Poisson.

Willinger, Taqqu, Sherman and Wilson in their paper [20] provide possible explanation for the self-similar or long-range dependent nature of network traffic. They suggest that the superimposition of many ON/OFF sources whose ON-periods and OFF-periods exhibit infinite variance (Noah Effect) produces aggregate network traffic that is self-similar or long-range dependent in nature. In the web world, the sources would be the individual clients generating requests on the server and the ON and the OFF periods are the periods of user activity and inactivity respectively. They further prove statistically the presence of actual Noah Effect in the measured Ethernet traffic at source level using the Log Log Complimentary Distribution (LLCD) plots and Hills estimator discussed in the Chapter 2.

Park, Kim and Crovella in [23] presented the results of their simulations on realistic client/server network environment where traffic sources contended to get access to shared bounded resources. They showed that the degree of self-similarity is directly determined by the degree of heavy-tailed distribution of file sizes in the server. They further show that changes in network resources (bot-

tleneck bandwidth and buffer capacity), network topology and distribution of inter-arrival times between requests do not affect degree of self-similarity. Their study reveals that reliable transmission and flow control mechanism of TCP help preserve the long-range dependency of traffic induced by heavy tailed file-size distribution. In contrast, UDP based unreliable protocol tends to make traffic less self-similar. In their work they also discuss the performance implications of self-similarity. Increased self-similarity degrades the performance because of drastic increase in queuing delays. However, the packet loss rate and retransmission rate increase only gradually with increased self-similarity when reliable protocol like TCP is used.

3.1.2 Workload Characterization of Web Traffic

In this section we review the work related to characterization of Web traffic in terms of requests and sessions.

Request Based Workload Characterization

Crovella and Bestavros [16] extended the studies conducted in [1, 22] to World Wide Web (WWW) and showed evidence of self-similar nature of WWW traffic at request level. They instrumented the Mosaic browser application used in NCSA to capture the user access patterns in terms of user requests for document transfers that included details of timing of requests, transfer lengths, the requested URL and the workstation from which the requests were issued. Their analysis showed that the web traffic shows self-similar characteristics when the demand is high. This result is in conformity with the one showed by Leland, Taqqu, Willinger and Wilson [1] for Ethernet traffic. They further provide possible explanation for this result based on the work related to ON/OFF traffic model in [20]. They consider each workstation as a source of packets. The ON period corresponds to the transmission durations of individual web files and the OFF periods correspond to the interval between transmissions. Using Complimentary distribution plots, Hills Estimator and Hurst Exponent they show that ON periods are significantly heavy-tailed with a value of $\alpha = 1.2$ and $H = 0.7 - 0.8$. They further show that the underlying reason for this is the actual heavy-tailed distribution of file sizes present in the web-server. They do this by surveying file size distribution on 32 different web-sites. Further, the authors study the OFF intervals and observe that the complimentary distribution plots has two distinct slopes. To explain this they

say that the OFF period consists of two different intervals "Active OFF" interval and "Inactive OFF" interval; the "Active OFF" period being the time when the web browser has received the data from the server and is busy interpreting, formatting and displaying the data and the "Inactive OFF" period being the user "think time" where the user is looking and trying the grasp the data displayed on the browser. The data/graphs presented clearly suggest that the heavy-tailed nature of the OFF periods is primarily due to the distribution of the Inactive OFF periods, rather than from the active OFF periods.

Arlitt and Williamson [5] in there study first used the web server logs from six different web servers to characterize the web-server workload. They studied characteristics such as file size, transfer size, file referencing, inter-arrival time between requests, inter-arrival time for individual document requests and geographic locations of web clients by extracting this data from the web access logs. They confirm the results from the previous study that the file size distribution and transfer size are indeed heavy-tailed. Their study also suggests a non-Poisson distribution (in-fact a heavy-tailed distribution) of inter-arrival time between requests. However, they confirm that the request arrival time for individual documents is Poisson. They further suggest the implications of their study in the caching mechanisms used in web systems. These implications can be summarized as follows:

1. *Trade-off Requests Vs Bytes* - A choice to be made in caching designs is to reduce the number of requests presented to web servers (by having more cache hits) or to reduce the volume of Internet traffic in terms of bytes transferred. Their analysis revealed that reduction in Internet traffic does not necessarily reduce the number of requests. Appropriate cache policy is a trade-off between the two and depends on what is the bottleneck resource: CPU on server or network bandwidth.

2. *Cache replacement policy* - The data that the analyzed revealed the absence of strong temporal locality. This suggests that LFU (Least Frequently Used) would be a more attractive cache replacement policy as compared to LRU (Least Recently Used)

3. *Other cache policies* - They suggest use of cache policies such as: "never cache documents greater than X bytes" (since it uses too much cache space and adversely affects the hit rate),

"never cache a document less than Y bytes" (since it does not save much on bytes transferred by the server)

Bradford and Crovella [10] developed an analytical workload generator tool SURGE (Scalable URL Reference Generator) for HTTP traffic. Since the tool is based on analytical model it provides flexibility to adjust the workloads for varying demands. The workload generator takes into account the self-similar nature and assumes a heavy-tailed distribution for both the ON and the Inactive OFF time as discussed in [16]. It also adheres to the following five statistical properties of web system:

1. The tail of the file-size distribution on server is heavy-tailed. The body of the distribution is assumed to be lognormal.

2. Request sizes (or sizes of files transferred over the network) is assumed to be heavy-tailed.

3. Popularity of file within a server follows Zipf's Law (i.e. popularity of a file is inversely proportional to its rank).

4. Consideration is given to the number of embedded references for a particular web request. Based on the traces from network traffic data, this distribution was assumed to be Pareto.

5. Temporal locality - the likelihood that once a file is requested, it will be requested again, plays an important role since it increases the caching effectiveness significantly. Temporal locality has been shown have log normal distribution by the authors and is assumed to have this distribution is SURGE.

6. The Active OFF times are modelled as Weibull distribution.

Based on the above statistical properties and a solution to few more challenges, results of the workload generator showed that the network traffic generated was self-similar in nature. SPECweb96 a commonly used benchmark for characterizing web workload does not show such characteristics. Also the SURGE generated workload maintains a much larger number of open connections which results in higher CPU utilization as compared to SPECweb96.

Session Based Workload Characterization

Menasc, Almeida, Fonseca, Mendis [8] first characterized web workload in terms of user sessions. Starting from the web access logs session logs were generated. Then, a state transition graph indicating the navigational pattern of user in session was generated for each session. This state transition graph was called Customer Behavior Model Graph (CMBG) and grouping of these CMBG's into clusters was done. Each of the CMBG group identified was then characterized in terms of:

1. Workload intensity: session arrival rate, average think time between requests of the CMBG.

2. Resource usage parameters.

The above methodology was applied to data collected from a simulated electronic bookstore web-site and the sessions were clustered into 6 groups. As a continuation of this work in [24] priority based resource management policies based on CBMG representation and simulated workload were proposed in order to increase the business-oriented metrics such as revenue per second.

In [7], the authors studied web server logs of the 1998 world cup web-site. The showed how the threshold value of session length affects the number of user sessions in a given period. They focused their analysis on some of the session characteristics such as the number of requests per session, session length, and intersession arrival times. The results of the study revealed that caching at web clients, proxies and within the network is changing the workloads seen by web servers. The lack of an efficient, supported and widely adopted cache consistency mechanism is the main cause of these changes and is the primary reason why web caches fail to significantly reduce web server workloads during times of extreme user interest in the content on those servers.

Menasce, Almeida et al. in their work [4] characterized the workload at three different levels: protocol level represented by HTTP request layer, application level represented by function layer and user level represented by session layer. They filtered the access log data collected from two different web-sites: an e-tailer bookstore and an auction site, to consider only requests pertaining to e-business functions (search, register, pay) and ignored the requests related to images etc. The conclusions from their work can be summarized as follows:

1. Most sessions are less than 1000 sec.

2. More than 70% of functions performed are browsing and selection functions the rest small percentage are ordering functions.

3. Request arrival process is self-similar in nature.

4. Session length measured in terms of number of requests per session exhibit heavy-tailed distribution, especially in the presence of robots.

. They further present a study indicating some of the characteristics of robots or web-crawlers:

1. Robots typically have longer session lengths.

2. There is no logical sequence of pages followed by robots. Usually the first page accessed by robots is not the home page.

3. Robots typically do not execute unfeasible functions such as "add to cart" or "make payment".

4. Though it is expected to have a fixed arrival rate for requests within robot sessions this might not be the case as the request arrival rate also depends on the server response time. However, there is no "user think time" involved.

5. Before issuing any request, robots often identify themselves by issuing request to the robots.txt file.

3.1.3 Contradicting Self-Similarity and Heavy-Tailed Distribution in Network Traffic

As we have seen in the previous sections of this chapter, lot of work has been done to show self-similarity and heavy-tailed distribution in web traffic. In this section, we review some work that raises doubts about self-similarity and the methods used to estimate H and α.

Allen Downey [9] in his work suggested that the methods employed for determining the value of α, the index of heavy-tailed distribution could be misleading. He showed that there could be other distributions like "lognormal" that are not heavy tailed, but whose Log-Log Complimentary Distribution (LLCD) plots can appear heavy-tailed. He proposed a new statistical technique for identifying long-tailed distribution based on the curvature of the LLCD plot. Based on this new technique he showed that the TCP packet and connection inter-arrival times are not heavy tailed

for extreme tails contradicting the results shown by Paxson and Floyd in [22]. He also showed that there is not enough evidence in the study by Arlitt and Williamson [5] about the heavy-tailed distribution of inter-arrival time between WWW requests. Using the new method he also concluded that the HTTP and FTP burst lengths and burst sizes appear to be more lognormal than heavy-tailed contradicting the results presented in [16] and [22].

Karagiannis, Faloutsos and Riedi [12] in their paper suggested that the existing methodologies for estimating self-similarity and long-range dependence could give conflicting results. As a result, studies based on these existing methods could arrive at misleading conclusions. To show this they used synthetic data with known Hurst Exponent and compared this value with the one estimated by the known methods for accuracy. Also, to check the sensitivity, they took an artificial non-long-range dependent data and showed that some of the Hurst Exponent estimators can be easily fooled. They made the following conclusions:

1. Whittle is the most robust of all estimators of Hurst exponent that are discussed in Chapter 2.

2. Periodogram estimator also gives satisfying results.

3. The Abry-Veitch estimator seems to overestimate the value of Hurst exponent.

4. R/S plot gives sufficiently accurate estimation of H when $H < 0.8$.

5. The other methods cannot provide sufficient estimations of H.

6. Noise, periodicity and trend affect the estimation of Hurst exponent. Whittle estimator which is supposed to be a good estimator is the most sensitive to these factors.

Some important lessons from their study is that researchers should not rely on a single estimator for deciding on Long range dependence and that reporting of Hurst exponent is meaningful only if it is accompanied by the method used and the confidence interval or correlation coefficient.

Karagiannis, Molle and Faloutsos in their works [25] [13] present the need to revisit the previously held notion of Poisson traffic process in view of the fact that the link speed and number of Internet-connected hosts have increased almost three-folds over the last ten years. They confirm the presence of heavy-tailed distribution of inter-arrival times for network traffic from 1989 in their

study. This is in congruence with the study done by others. However, for the more recent network backbone traffic (from Jan-Aug 2003) they concluded that:

1. Packet arrivals appear Poisson at sub-second time scales.
2. Internet traffic appears non-stationary at multi-second time scales.
3. Internet traffic exhibits long-range dependence at scales of seconds and above.

The above conclusions lead us to believe that the characteristics of internet traffic are continuously changing. As stated in [13]:
"As the Internet increases in size and the technologies connected to it change, we must constantly monitor and reevaluate our assumptions to ensure that our conceptual models correctly represent reality."

3.2 Our Contribution

As we see above, considerable amount of research has been done in the past with the aim of characterizing web-server workload. In this thesis we analyze web access logs from nine different web-sites. The objective is to enhance the understanding on web workload at the request and session levels. We examine rigourously the tails of the distributions of the request and session characteristics and draw conclusions about reasons behind the heavytailed distribution. Our contribution towards this widely researched field is as follows:

1. Most of the work done till date focusses on the request based parameters [16, 23, 5]. The work done on session parameters in [7] presents analysis of parameters such as number of requests per session and session length. In this thesis we perform a detailed statistical analysis on following request and session based parameters:
    - Request parameters:
        - Requests per unit time
        - Request inter-arrival time
    - Intra-session parameters:
        - Bytes transferred per session

  - Number of requests per session
  - Session length
- Inter-session parameters:
  - Sessions initiated per unit time
  - Time between session initiations

2. Unlike the previous work on session based parameters in [7] that performs the statistical analysis in terms of Cumulative Distribution Function (CDF) plots, we do a more detailed statistical analysis and estimate the value of α, the index of heavy-tailed distribution using the Hills plot, Smooth Hills plot, Alternate Hills plot and Alternate Smooth Hills plot to check if the session parameters are well described by heavy tailed distributions. We also compare our results with the results obtained using Log Log Complimentary Distribution (LLCD) plot in [3] to increase the confidence in our estimation of α.

3. We study the impact of workload on the degree of heavy-tailedness. For this, we analyze the workload for the typical low, medium and high periods for nine different web servers including three private web servers.

4. We also perform a preliminary analysis on the effect of robots on the heavy-tailedness of intra-session parameters.

5. For estimating self-similarity of workload we use all the seven method discussed in Chapter 2 and compare the values produced by different methods to confirm some of the results shown in [12].

Chapter 4

Experimental Setup and Our Approach

In this chapter we discuss the approach that we followed for the web workload characterization. We also look at the detailed experimental setup.

4.1 Overview of Experimental Setup

Web servers maintain different kinds of logs viz. *access logs, error logs, SSL logs, referrer logs, agent logs* etc. These logs can reveal a lot of information about the workload characteristics and the errors experienced by users.

Figure 4.1 shows our data extraction and analysis process [26]. We store the log files obtained from the web server administrators in a central log repository. Since, the text format of these log files is not suitable for flexible, customized analysis we store the access logs and error logs in a relational database; each row in the log file being one record in the relational database. We then create sessions from these log entries and extract various session parameters into another database table. Workload characterization, error characterization and reliability analysis is done based on these request and session logs. This thesis focusses on request and session based workload characterization. Error characterization and reliability analysis is not within the scope of this thesis. The logs for workload characterization of web servers were obtained from the following nine web sites:

1. WVU - West Virginia University

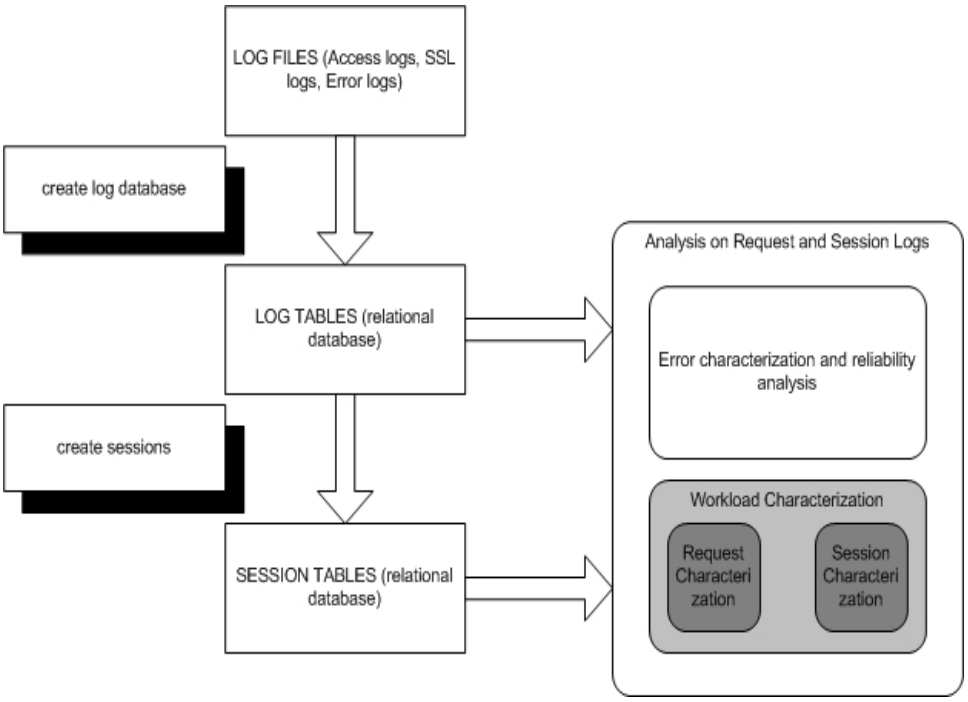

Figure 4.1: Data collection and analysis process

2. ClarkNet - A Commercial Internet Service provider.

3. CSEE - Computer Science and Electrical Engineering department at West Virginia University.

4. Three NASA Public web servers - NASA-Pub1, NASA-Pub2 and NASA-Pub3

5. Three NASA Private web servers - NASA-Pvt1, NASA-Pvt2 and NASA-Pvt3

4.2 Access Log and SSL Log format

In this section we discuss the format of logs maintained on the web servers. Though there are many vendors providing their own flavor of web servers, each of these servers maintain the request logs in a standard format called the Common Log Format (CLF) [27]. A CLF entry in the access log is of the following form:

```
RemoteHost Identity Authorization [Timestamp] "Request Line" Status Bytes
```

The fields in this entry are defined as follows:

- `RemoteHost` - The IP address of the client or the remote host making request to the web server. Settings can be made to record the host name instead of the IP address.

- `Identity` - The RFC 1453 identity of the client determined by the `ident` on the client machine. If the value is not available then a "-" (hyphen) is recorded instead.
- `Authorization` - The userid of the person requesting the document as determined by HTTP authentication. If the document in not password protected this entry will be a "-" (hyphen)
- `[Timestamp]` - The time that the server finished processing the client request. The format of the time stamp field is `[dd/mmm/yyyy:hh:mm:ss zone]`. The logs used in this thesis use one second granularity for recording requests.
- `Request Line` - The request line contains the HTTP method used (e.g. GET, POST), the resource requested (e.g. /index.html) and the protocol used by the client (along with its version - e.g. HTTP/1.0). The format of this field is `"GET /index.html HTTP/1.0"`
- `Status` - The status code that the server sends back to the client. It reveals whether the request resulted in a successful response (codes beginning with 2), a redirection (codes beginning with 3), an error caused by client (codes beginning with 4) or and error on the server side (codes beginning with 5)
- `Bytes` - The size of the object (in bytes) returned to the client, not including the response headers.

A sample entry in the access log of a web server is as follows:

```
12.10.219.28 - - [03/Mar/2003:00:00:01 -0500] "GET /index.html HTTP/1.0" 200 3649
```

This entry indicates that the request for the file `/index.html` by the client with IP address `12.10.219.28`, was successfully satisfied by the server on March, 03, 2003, one second after midnight, eastern time.

For the purpose of web-server workload characterization, the fields `RemoteHost`, `Timestamp` and `Bytes` from the logs are important.

4.3 Our Approach

In this section we discuss in details our approach for web server workload characterization. We first look at some of the details related to creating access log and session tables and then present

the method that we used for the analysis of web server workload.

4.3.1 Creating Access Log Table

As shown in Figure 4.1, all the log files were first dumped into the database table for flexible and customized analysis. We have used Oracle 10g as the database for storing these access logs. Java code was used to read the log files and insert the data in the Oracle tables.

Web servers can be configured so that the requests over SSL can be recorded in a different log file to keep track of SSL related activities. Of the nine web sites that we analyzed, only CSEE server maintained separate logs for SSL activity. Another peculiarity of the CSEE web site was the use of load balancer to distribute the traffic on the web server to two different servers - Bernerslee and Bhelendrof for the purpose of better performance, availability and fault tolerance. Because of this part of the user requests were logged on the Bernerslee server log and part on the Bhelendrof server. Thus for the CSEE server, for a given period, we had 4 logs in which the client requests were logged: Bernerslee access log, Bernerslee SSL log, Bhelendrof access log, Bhelendrof SSL log. For all other servers: WVU, ClarkNet, NASA public and NASA private we only had one access log file for a given period. The data from these logs was put in the access log table, each server having a table of its own. In addition to all the fields from the access log discussed in the previous section, we have few more fields in the access log table:

- `LOG_NAME` - This field indicates the log file from which the values go into the access log table. e.g. SSL_BERN indicates that the it is and SSL log record from the Bernerslee server. We have used this field for manually testing the correctness of process.
- `URI_LEN_ERR_FLAG` - The request field in the access log sometimes contain very long strings (server thousand characters). In such cases, the request string is truncated to 5000 characters and this flag is set to "true" indicating a problem with this access log record.
- `SESSION_ID` - This field is initially kept NULL.

4.3.2 Creating Sessions and Extracting Session Parameters

Our next step, was to create sessions and extract various session parameters. A session represents the interaction of a single user with the web server. A session begins when the user issues

a request for a particular page on the web site for the first time and ends when the user gets the response for his last request and closes the browser or accesses another web site from the browser window. Typically there will be multiple requests in a session. Even, if the user makes a request for a single page during the session, there could be multiple embedded requests corresponding to this single request. For example, accessing a particular HTML page may involve requesting the HTML page and then making requests for accessing the images embedded in the page.

For the purpose of identifying sessions, we define session as a sequence of requests issued from the same IP address with the time between requests less than a threshold value equal to 30 minutes [26]. The two key points in this definition of session are:

1. *Identifying the user by IP address* - This is a reasonable assumption to make despite of the inaccuracies for reasons such as use of proxy server between a group of users and the web server.

2. *The value of threshold as 30 minutes* - We choose this value because increasing this time beyond 30 minutes does not significantly change the number of sessions i.e. the number of sessions for a give period (say, a week) remain constant even after increasing this threshold time beyond 30 mins [26].

Based on the above definition of session, we generated session ID's for every request in the access log table and updated the SESSION_ID field in the access log table. We wrote PL/SQL code for this. We then extracted the following parameters from the access log table and put them in the session table:

Table 4.1: Session Table

SESSION_ID	Unique session ID
REQUEST_COUNT	Total number of requests in that session
SESSION_LENGTH	Total duration of the session in seconds
BYTES_TRFD	Total number of bytes transferred in the session
SESSION_START_TIME	Timestamp of the first request in the session
ERR_400_COUNT_ALL	Number of requests in the session having status code starting with 4
ERR_500_COUNT_ALL	Number of requests in the session having status code starting with 5
E_CNT_400	Number of requests in the session having status code400
E_CNT_401	Number of requests in the session having status code 401
E_CNT_402	Number of requests in the session having status code 402
E_CNT_403	Number of requests in the session having status code 403
E_CNT_404	Number of requests in the session having status code 404
E_CNT_405	Number of requests in the session having status code 405
E_CNT_406	Number of requests in the session having status code 406
E_CNT_407	Number of requests in the session having status code 407
E_CNT_408	Number of requests in the session having status code 408
E_CNT_409	Number of requests in the session having status code 409
E_CNT_410	Number of requests in the session having status code 410
E_CNT_411	Number of requests in the session having status code 411
E_CNT_412	Number of requests in the session having status code 412
E_CNT_413	Number of requests in the session having status code 413
E_CNT_414	Number of requests in the session having status code 414
E_CNT_415	Number of requests in the session having status code 415
E_CNT_416	Number of requests in the session having status code 416
E_CNT_417	Number of requests in the session having status code 417
E_CNT_500	Number of requests in the session having status code 500
E_CNT_501	Number of requests in the session having status code 501
E_CNT_502	Number of requests in the session having status code 502
E_CNT_503	Number of requests in the session having status code 503
E_CNT_504	Number of requests in the session having status code 504
E_CNT_505	Number of requests in the session having status code 505

Each row in the session table 4.1 represents a single session.

The session data in the session table along with the request data in the access log table form the basis for web-server workload characterization.

4.3.3 Workload Characterization Methodology

In this thesis we characterize web server workload in terms of the following request and session based parameters:

1. Request based parameters

- Requests per unit time
- Requests inter-arrival time

2. Session based parameters
   - Intra-session parameters
     - Bytes transferred per session
     - Number of requests per session
     - Session Length
   - Inter-session parameters
     - Sessions initiated per unit time
     - Time between session initiations

Parameters such as requests per unit time and sessions initiated per unit time are time-series based. We estimate self-similarity or long-range dependence of web workload in terms of these parameters. For all the other parameters, we check if their distribution is heavy-tailed.

For determining self-similarity, we used the java based SELFIS tool [2] introduced in Chapter 2. SELFIS has a Graphical User Interface (GUI) and takes one single-column file as in input (via the GUI) to produce the estimate of H using different methods discussed in Chapter 2. We observed that estimation of H using this tool was hardware resource and time intensive. Hence, to make our task of estimating H less tedious, we automated the process, by eliminating the GUI interaction. For this purpose we decompiled the SELFIS java class files using the DJ Decompiler [28] and after understanding the decompiled code, wrote our own java based client which took the the input file as a command line argument and stored the estimates of H in an excel file. We also used JExcelApi - Java API for manipulating excel files [29] for this purpose.

For estimating the value of α, the index of heavy-tailed distribution using Hills plots, Smooth Hills plots, Alternate Hills plots and Alternate Smooth Hills plots we wrote another java program. This program takes the file containing the raw data (parameter values) as input and produces the data needed for plotting the Hills plot, Smooth Hills plot, Alternate Hills plot and Alternate Smooth Hills plots.

The above two programs: one for estimating H and other for generating data for estimating α are independent and can be integrated in the future to form an integrated tool that can be used to study two very important phenomena observed in the web traffic viz. self-similarity and heavy-tailed distributions.

Chapter 5

Results

This chapter presents the results of the Web workload characterization. As discussed in earlier chapters we analyze the workload in terms of following request and session characteristics:

1. Request Characteristics
    - Requests per unit time
    - Requests inter-arrival time
2. Intra-Session Characteristics
    - Bytes transferred per session
    - Number of requests per session
    - Session length
3. Inter-Session Characteristics
    - Sessions initiated per unit time
    - Time between session initiations

For the two characteristics based on time series: requests per unit time and sessions initiated per unit time we estimate the Hurst Exponent, H and draw conclusions about self-similar nature of network traffic. For all other characteristics we figure out if their distribution is heavy-tailed by estimating the value of α, the index of heavy-tailed distribution.

We then do some preliminary analysis of the impact of robots on the intra-session parameters.

Our analysis for all the parameters mentioned above is for a period of one week and typical low, medium and high 4 hour intervals.

5.1 Raw Data

As indicated in the earlier chapters, workload from nine different web-sites is analyzed. Table 5.1 shows the summary of the raw data for one week period for the nine data-sets that are analyzed in this thesis.

Table 5.1: Raw Data-set

Data set	Data period	Total requests	Average requests per day	Total sessions	Average sessions per day	Total MB transferred	Average MB transferred per day
WVU	12-Jan-04 to 18-Jan-04	15,785,164	2,255,023	188,213	26,887	36,160,622,401	5,165,803,200
Clarknet	28-Aug-95 to 03-Sep-95	1,654,882	236,412	139,745	19,964	14,454,836,876	2,064,976,697
CSEE	12-Apr-04 to 18-Apr-04	396,743	56,678	34,343	4,906	10,630,592,753	1,518,656,108
NASA-Pub1	12-Apr-04 to 18-Apr-04	3,641	520	970	139	425,751,485	60,821,640
NASA-Pub2	12-Apr-04 to 18-Apr-04	39,137	5,591	3,723	532	325,614,180	46,516,311
NASA-Pub3	12-Apr-04 to 18-Apr-04	5,597	800	644	92	221,624,757	31,660,679
NASA-Pvt1	12-Apr-04 to 18-Apr-04	1,163	166	39	6	17,562,511	2,508,930
NASA-Pvt2	12-Apr-04 to 18-Apr-04	3,203	458	188	27	5,252,705	750,386
NASA-Pvt3	12-Apr-04 to 18-Apr-04	21,799	3,114	1,076	154	95,050,895	13,578,699

Note that for all the data-sets the start day is Monday and the end day is a Sunday. From the above table we can see that there is a large variation in the workload on the different servers that we analyze. NASA-Pvt1, has the lowest workload with 166 requests per day; WVU server, on the other extreme processes almost 15,000 times as much requests per day (2,255,023).

Also note that for WVU, Clarknet and CSEE servers the number of requests, number of sessions and bytes transferred are proportional i.e. higher the number of requests, higher are the number of sessions and higher are the total bytes transferred during that period. However, for some of the NASA web-sites, this is not the case. For example, as compared to NASA-Pub1, the NASA-Pvt3 web-site has significantly larger number of requests; however, the total number of bytes transferred in the NASA-Pub1 web-site is a lot more as compared to NASA-Pvt3. This indicates that the NASA-Pub1 web-site though has less number of users accessing it, lot of activity on the web-site is related to uploading and downloading data.

5.1.1 Low, Medium and High periods

We divide the one week period into 42 equal intervals of 4 hours each and identify the typical low (LOW), medium (MED) and high (HIGH) periods for each data-set. To identify these intervals we use total number of requests as a criteria, i.e, interval with highest number of requests is chosen as HIGH, interval with lowest number of request is chosen as LOW and interval with median requests is chosen as MED. The motivation behind identifying the LOW, MED and HIGH periods is to study the web characteristics at different level of workloads and see the variations, if any. Table 5.2 shows the typical low (LOW), medium (MED) and high (HIGH) 4 hour periods identified for all the data-sets.

Table 5.2: LOW, MED, HIGH workload periods

	LOW	MED	HIGH
WVU	32 (SAT 04:00AM-08:00AM)	34 (SAT 12:00PM-04:00PM)	4 (MON 12:00PM-04:00PM)
Clarknet	38 (SUN 04:00AM-08:00AM)	35 (SAT 04:00PM-08:00PM)	16 (WED 12:00PM-04:00PM)
CSEE	32 (SAT 04:00AM-08:00AM)	27 (MON 08:00AM-12:00PM)	4 (MON 12:00PM-04:00PM)
NASA-Pub1	42 (SUN 08:00PM-12:00AM)	7 (TUE 12:00AM-04:00AM)	9 (TUE 08:00AM-12:00PM)
NASA-Pub2	35 (SAT 04:00PM-08:00PM)	2 (MON 04:00AM-08:00AM)	10 (TUE 12:00PM-04:00PM)
NASA-Pub3	35 (SAT 04:00PM-08:00PM)	31 (SAT 12:00AM-04:00AM)	9 (TUE 08:00AM-12:00PM)
NASA-Pvt1	1 (MON 12:00AM-04:00AM)	18 (WED 08:00PM-12:00AM)	21 (THU 08:00AM-12:00PM)
NASA-Pvt2	1 (MON 12:00AM-04:00AM)	32 (SAT 04:00AM-08:00AM)	3 (MON 08:00AM-12:00PM)
NASA-Pvt3	34 (SAT 12:00PM-04:00PM)	32 (SAT 04:00AM-08:00AM)	27 (FRI 08:00AM-12:00PM)

To identify these intervals we use total number of requests as a criteria, i.e:

1. We divide the entire week into 42 equal intervals of 4 hour each.
2. We then calculate the total number of request during each of these intervals and
3. Choose the interval with highest number of request as HIGH, lowest number of request as LOW and the median as a MED period.

Table 5.3 shows the total requests, total sessions and total bytes transferred during the low, medium and the high periods identified in table 5.2.

From Table 5.3 we see that total bytes transferred and total number of sessions also adhere to these LOW, MED and HIGH periods identified i.e.

$$TotalBytes_{HIGH} > TotalBytes_{MED} > TotalBytes_{LOW}$$

and

$$NumberOfSessions_{HIGH} > NumberOfSessions_{MED} > NumberOfSessions_{LOW}$$

Table 5.3: Total Requests, Number of Sessions, Bytes Transferred in LOW, MED and HIGH periods

	Total Requests			Number of Sessions			Bytes Transferred		
	LOW	MED	HIGH	LOW	MED	HIGH	LOW	MED	HIGH
WVU	47,836	304,725	1,208,238	1,131	3,980	10,287	290,621,917	817,484,273	2,531,008,422
Clarknet	15,130	40,490	75,068	1,332	3,134	5,525	133,310,047	349,055,748	646,134,986
CSEE	1,188	4,783	40,736	366	890	1,586	42,856,743	149,429,616	246,563,612
NASA-Pub1	15	71	180	13	28	44	2,801,560	7,504,972	21,979,503
NASA-Pub2	297	799	2,421	51	93	161	1,541,900	9,469,727	14,885,032
NASA-Pub3	9	121	253	4	18	22	7,41,629	3,230,806	3,561,239
NASA-Pvt1	1	3	129	1	1	4	2,379	16,612	664,964
NASA-Pvt2	2	7	408	1	1	20	1,108	31,915	820,438
NASA-Pvt3	48	438	1,792	3	6	69	149,812	7,131,743	5,926,925

5.2 Request Characteristics

Request is the most basic unit of web workload. Users interact with the web-server by sending requests and in turn receive response a response from the server. If the volume and distribution of requests are estimated reasonably, web-server resources can be effectively managed. In this section we study two request based characteristics: requests per second and request inter-arrival time for the entire weeks data.

5.2.1 Requests per second

Figure 5.1 shows the raw-signal of requests per unit time for one weeks period for the NASA-Pub2 data-set. In this section we estimate the degree of self-similarity of network traffic based on the analysis of the raw-signal for different data-sets for a period of one week. The raw signal forms a time series with each point in the time series indicating the number of requests that the server processes. For this series we estimate the value of H, the Hurst exponent using the SELFIS tool [2] using seven methods discussed in Chapter 2 viz: Aggregate Variance, R/S, Periodogram, Absolute Moments, Variance of Residuals, Abry-Veitch, Whittle. Figure 5.2 shows the output of the SELFIS tool with estimates of H for NASA-Pub2 data-set.

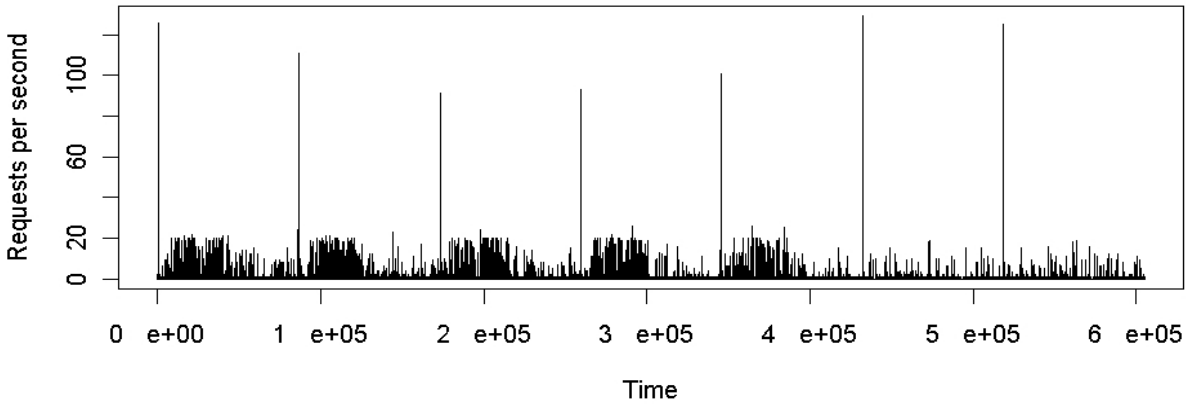

Figure 5.1: Raw data, request per unit time NASA-Pub2

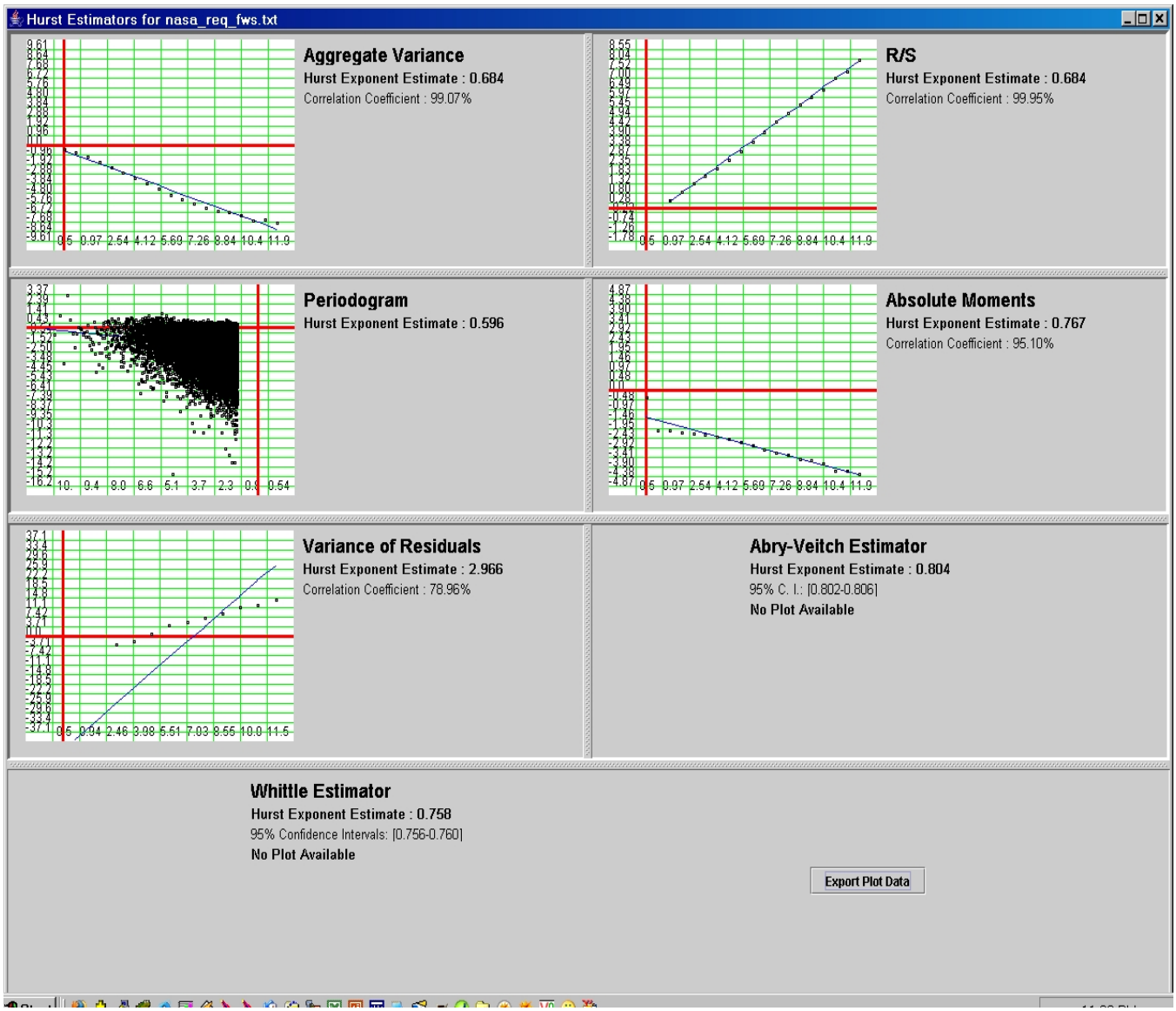

Figure 5.2: Hurst Exponent values - NASA-Pub2 [2]

Table 5.4 shows the estimated values of H for different data-sets including the upper and lower bounds for Abry-Veitch and Whittle estimator. As we can see, almost all methods for all data-sets show a value of H in the range [0.5, 1) indicating that the web-traffic at request level is self-similar in nature.

Table 5.4: Hurst Exponent values, Number of Requests per second

Data-set	Agg. Var.	R/S	Period-ogram	Abs. Mome-nts	Variance of Residuals	Abry-Veitch Lower Bound	Abry-Veitch	Abry-Veitch Upper Bound	Whittle Lower Bound	Whittle	Whittle Upper Bound
WVU	0.927	0.905	0.642	0.837	1.031	0.779	0.781	0.783	0.763	0.765	0.767
Clarknet	0.869	0.879	0.786	0.840	0.941	0.732	0.734	0.736	0.694	0.695	0.697
CSEE	0.867	0.842	0.783	0.876	0.994	0.773	0.775	0.777	0.719	0.720	0.722
NASA-Pub1	0.528	0.573	0.481	0.743	3.389	0.626	0.628	0.630	0.590	0.592	0.594
NASA-Pub2	0.684	0.684	0.596	0.768	2.967	0.802	0.804	0.806	0.757	0.758	0.760
NASA-Pub3	0.581	0.630	0.514	0.743	3.276	0.628	0.630	0.632	0.565	0.567	0.568
NASA-Pvt3	0.686	0.693	0.692	0.731	3.631	0.701	0.703	0.704	0.653	0.655	0.657

Figure 5.3 shows the data from Table 5.4 in a graphical form with the scale on Y-axis ranging from 0.5 to 1.0. Each curve in the graph represents the estimate of H for different data-sets using a particular method. As we can see, Abry-Veitch and Whittle estimator curves are parallel to each other with Abry-Veitch method estimating slightly higher value of H as compared to Whittle estimator. This result is consistent with that shown in [13], which says that Abry-Veitch method over estimates when compared with Whittle method.

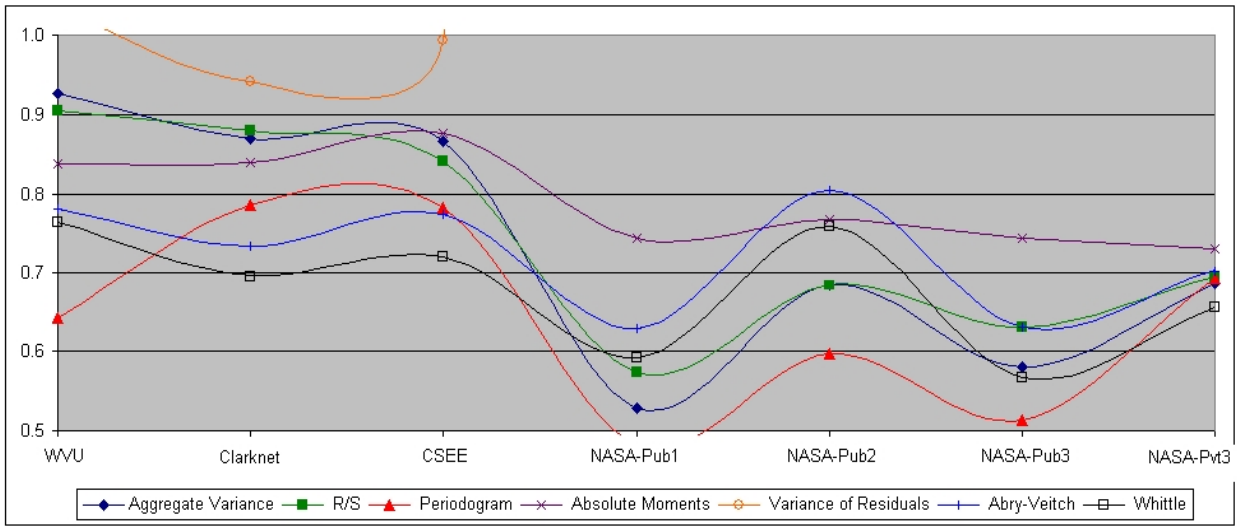

Figure 5.3: Estimate of Hurst Exponent, H (Request per Second)

We can see from Table 5.4 and Figure 5.3 that the Variance of Residuals method estimates a value of $H > 1$ for most of the data-sets. When compared with all the other methods which

estimate $0.5 < H < 1.0$, we can treat the values from this method as outliers. Table 5.5, shows the range and average value of estimate of H excluding the values given by Variance of Residuals method. The data in Table 5.5 is sorted by the total number of requests in descending order (i.e. WVU server process maximum number of requests while NASA-Pub1 server processes the least number of requests). We can clearly see that as the workload increases the average value of estimate of H increases. For WVU, Clarknet and CSEE workloads, $H > 0.8$. For NASA-Pub2 and NASA-Pvt3 where the workload is slightly lower, $0.69 < H < 0.72$ and for the lowest workload (NASA-Pub3 and NASA-Pub1), $H < 0.61$. Thus, we see that web traffic becomes more self-similar as the workload increases. This is consistent with the results shown by Leland, Taqqu *et al.* in [1] for LAN traffic.

Table 5.5: Range and Average for H Sorted by Total Number of Requests

	Total Requests	Range of H	Average H
WVU	15,785,164	0.642-0.927	0.841
Clarknet	1,654,882	0.695-0.875	0.820
CSEE	396,743	0.719-0.876	0.836
NASA-Pub2	39,137	0.596-0.806	0.715
NASA-Pvt3	21,799	0.653-0.703	0.693
NASA-Pub3	5,597	0.514-0.743	0.610
NASA-Pub1	3,641	0.481-0.743	0.590

More analysis involving aggregation of web traffic over different time scales (requests per minute, request per hour etc.) and removal of periodicity in the data-sets has been done and published in [3]. Such detailed analysis is not within the scope of this thesis.

5.2.2 Request Inter-arrival Time

In this section we analyze request inter-arrival time to see if heavy-tails are present in the distribution. Figure 5.4 shows a sample Hills plot for NASA-Pub3 data-set for one weeks period.

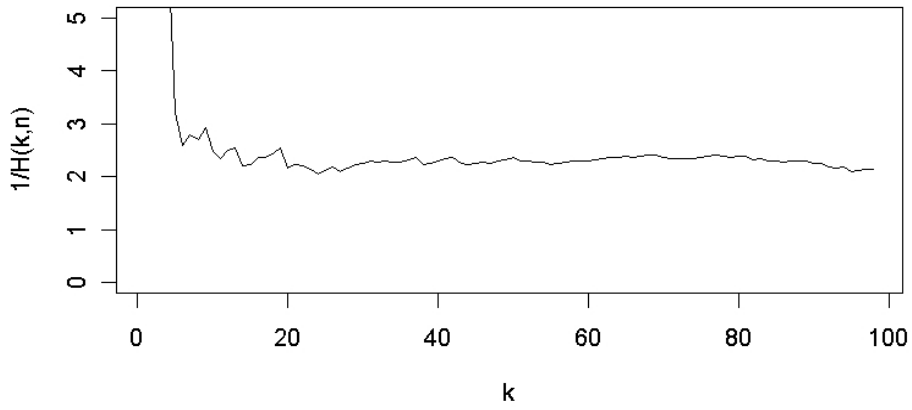

Figure 5.4: Hills Plot ($\alpha = 2.3$) - NASA- Pub3 Request Inter-Arrival Time, One Week

Table 5.6 shows the value of α for all the data-sets as obtained from Hills plots. As we can see for most of the data-sets we were not able to estimate the value of α since the Hills plots did not stabilize. This could be because the distribution of request inter-arrival time does not fit well with Pareto model. As indicated in [3] we assume a uniform distribution of inter-request arrival time whenever there are more than one requests initiated during the same second. This uniform distribution assumption can also be flawed and can result in the inability in estimation of α. However, with the web servers recording requests at one second granularity this cannot be avoided. Apache 2.0 has provisions of recording requests at millisecond granularity and data collected from 2.0 server can result in more accurate analysis.

Table 5.6: α, Request Inter-arrival Time - Low, Medium, High and entire week

	LOW	MED	HIGH	Week	Total Sessions
WVU	3.9	NA	NA	5.9	15,785,164
Clarknet	NA	NA	NA	NA	1,654,882
CSEE	NA	NA	NA	2.9	396,743
NASA-Pub2	1.5	1.9	2.1	3.2	39,137
NASA-Pvt3	2.3	NA	NA	1.0	21,799
NASA-Pub3	NA	NA	NA	2.3	5,597
NASA-Pub1	NA	NA	NA	NA	3,641
NASA-Pvt2	NA	NA	NA	NA	3,203
NASA-Pvt1	NA	NA	NA	NA	1,163

5.3 Session Characteristics

Sessions represent the interaction of the users with the web-server. All requests within one session come from one user. As discussed in Chapter 4, we take 30 minutes as a threshold value for session length i.e. any request from a particular user that comes within 30 minutes of the previous request from the same user is treated to be a part of the same session.

In this section we analyze three intra-session characteristics: total bytes transferred in a session, number of requests per session and session length. We also analyze one inter-session characteristic: sessions initiated per unit time.

Sessions per unit time is a time series and as with requests per unit time we estimate the value of H, the Hurst exponent for this parameter. For all other parameters, we figure out whether the distribution is heavy-tailed by analyzing the data for entire week and the typical LOW, MED and HIGH 4 hour periods identified in table 5.2. As shown in Table 5.1 the total number of sessions in the week for NASA-Pvt1 and NASA-Pvt2 is too low (39 and 188) to draw any conclusions statistically. Hence we exclude these two data-sets from our analysis for one week.

Table 5.7: Raw data for HIGH period

	Number of requests	Number of sessions	Total bytes transferred
WVU	1,208,238	10,287	2,531,008,422
Clarknet	75,068	5,525	646,134,986
CSEE	40,736	1,586	246,563,612
NASA-Pub1	180	44	21,979,503
NASA-Pub2	2,421	161	14,885,032
NASA-Pub3	253	22	3,561,239
NASA-Pvt1	129	4	664,964
NASA-Pvt2	408	20	820,438
NASA-Pvt3	1,792	69	5,926,925

Table 5.7 shows the number of request, number of sessions and total bytes transferred for a typical HIGH period as identified in Table 5.2 for all the data-sets. As we see, NASA-Pub1, NASA-Pub3, NASA-Pvt1, NASA-Pvt2 and NASA-Pvt3 do not have enough number of sessions even for the HIGH period. Hence, for the 4 hour session analysis we exclude all these data-sets.

5.3.1 Intra-session Characteristics

Bytes Transferred per Session

Each request within a session is responsible for some data transfer between the client and the server. In this section, we characterize the total bytes transferred per session for the typical low, medium and high periods identified in table 5.2 and the entire week. Figure 5.5 shows a sample Hills plot (for CSEE HIGH period) used in estimation of α, the index of heavy-tailed distribution. Based on similar Hills plots we estimate the value of α for all the data-sets for all the periods.

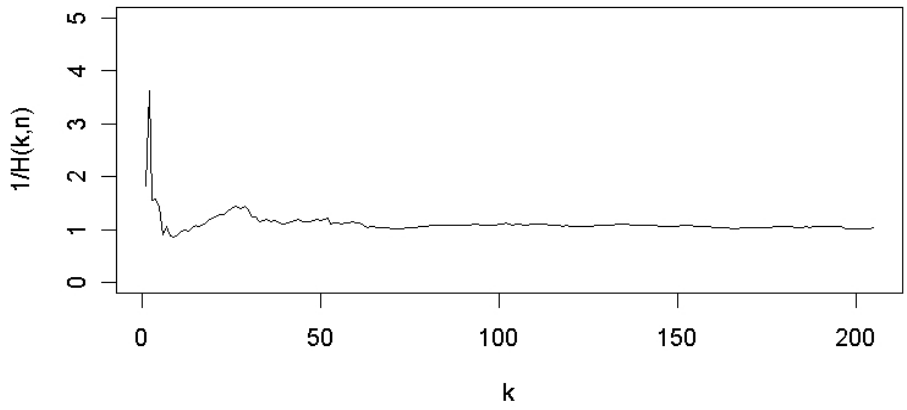

Figure 5.5: Hills Plot ($\alpha = 1.06$) - CSEE Bytes Transferred per Session, HIGH

Table 5.8 shows the summary of values of α. The same information is shown in bar charts in Figures 5.6 and 5.7.

Following observations can be made:

1. For all the periods low, medium, high and entire week the $\alpha < 2$, indicating that the distribution of bytes transferred per session is heavy-tailed. The reason for this is the heavy-tailed distribution of file-sizes in the web-server as indicated in [23].

2. [30] gives an estimate of α for the same data-sets for a different week. The values published in [30] are consistent with the values shown of α for one week shown in Table 5.8.

3. For WVU, Clarknet, CSEE and NASA-Pub2 data-sets: α for low, medium and high periods is in the same range. Also, this value of α is consistent with the value of α for one week. This indicates that the thickness of the tail of the distribution for bytes transferred per session is independent of the workload and the duration under consideration.

4. For, one weeks period, distribution for bytes transferred is more heavy-tailed for lightly loaded NASA and CSEE servers as compared to WVU and Clarknet servers.

Table 5.8: α Bytes Transferred per Session - Low, Medium, High and Entire Week

	LOW	MED	HIGH	Week	Total Bytes
WVU	1.1	1.32	1.63	1.4	36,160,622,401
Clarknet	1.7	1.89	1.86	2.0	14,454,836,876
CSEE	0.8	0.84	1.06	1.0	10,630,592,753
NASA-Pub1	NA	NA	NA	1.15	425,751,485
NASA-Pub2	NA	0.52	1.78	1.1	325,614,180
NASA-Pub3	NA	NA	NA	1.0	221,624,757
NASA-Pvt3	NA	NA	NA	1.2	95,050,895
NASA-Pvt1	NA	NA	NA	NA	17,562,511
NASA-Pvt2	NA	NA	NA	NA	5,252,705

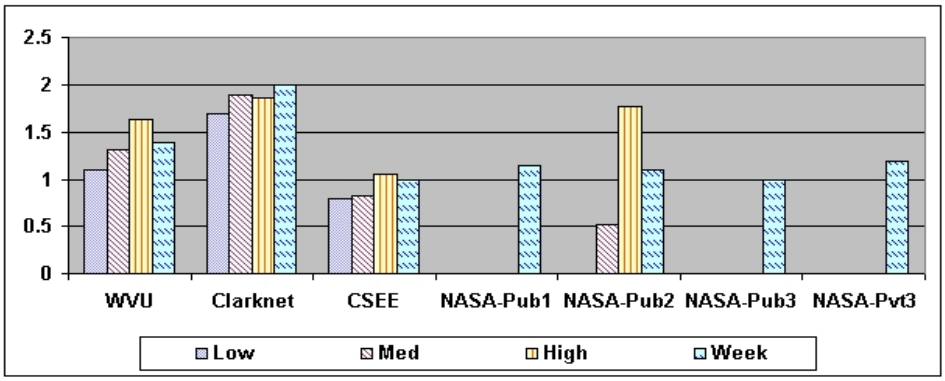

Figure 5.6: Bytes Transferred per Session, α value for each data-set

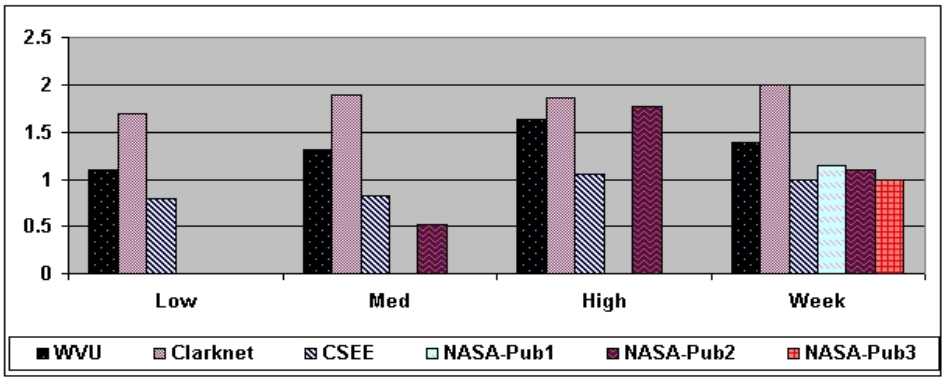

Figure 5.7: Bytes Transferred per Session, α value for each period

Number of Requests per Session

A user session consists of one or more requests. In this section we characterize the total number of requests per session. Figure 5.8 shows a sample Hills plot (for CSEE one week period) used in estimation of α, the index of heavy-tailed distribution. Based on similar Hills plots we estimate the value of α for all the data-sets for all the periods.

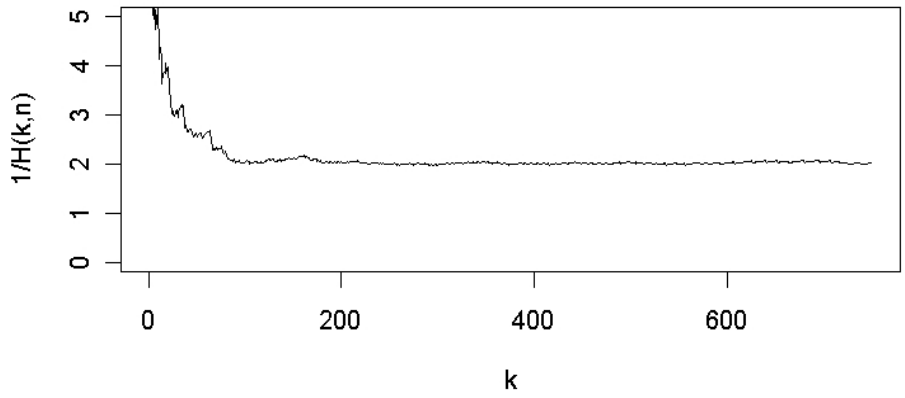

Figure 5.8: Hills Plot ($\alpha = 2.0$) - CSEE Number of Requests per Session, One Week

Table 5.9 shows the summary of values of α. Figures 5.9 and 5.10 show the bar chart for same data.

Following observations can be made:

1. *One week period* - The value of α for heavily loaded web-sites (WVU, Clarknet and CSEE) is greater than 2 indicating the presence of pareto-like distribution (which is not heavy-tailed). For all other data-sets, $\alpha < 2$, indicating a heavy-tailed distribution of number of requests within a session. The degree of heavy-tailedness is more for the lightly loaded NASA-Pub1 and NASA-Pub3 web-sites as compared to the heavily loaded web-sites. We believe that, this is because the impact of sessions with higher number of requests (that cause heavy-tailedness) is more on the lightly loaded servers that have lesser number of sessions.

2. *Low, medium and high periods* - The value of α for low, medium and high periods for WVU, Clarknet, CSEE and NASA-Pub2 servers is consistent with the value of α for the entire weeks

period with some tolerance. We will see later in the chapter that accurate estimation of α is a challenging task and is subject to the analysts interpretation.

Table 5.9: α, Number of Requests per Session - Low, Medium, High and Entire Week

	LOW	MED	HIGH	Week	Total Requests
WVU	1.7	2.0	1.9	2.1	15,785,164
Clarknet	2.32	1.8	1.9	2.6	1,654,882
CSEE	2.0	1.93	2.33	2.0	396,743
NASA-Pub2	NA	1.6	1.62	1.9	39,137
NASA-Pvt3	NA	NA	NA	1.6	21,799
NASA-Pub3	NA	NA	NA	1.4	5,597
NASA-Pub1	NA	NA	NA	1.3	3,641
NASA-Pvt2	NA	NA	NA	NA	3203
NASA-Pvt1	NA	NA	NA	NA	1,163

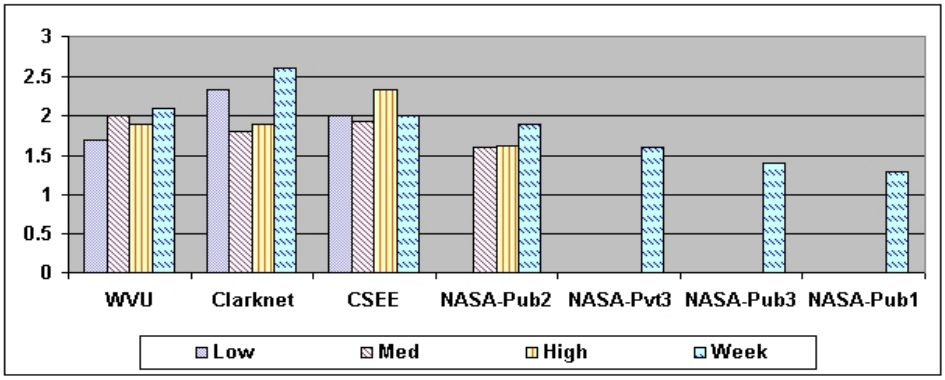

Figure 5.9: Number of Requests per session, α value for each data-set

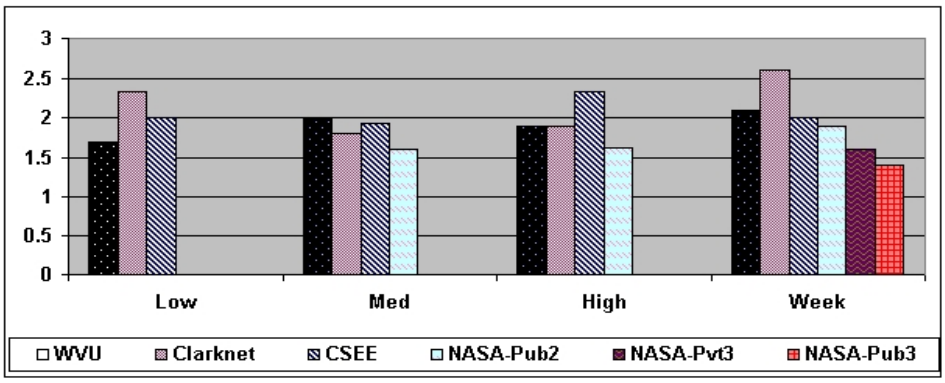

Figure 5.10: Number of Requests per session, α value for each period

Session Length

Session length is the amount of time a user stays connected and interacts with the web-server. Figure 5.11 shows a sample Hills plot (for Clarknet HIGH) used in estimation of α, the index of heavy-tailed distribution. Based on similar Hills plots we estimate the value of α for all the data-sets for all the periods.

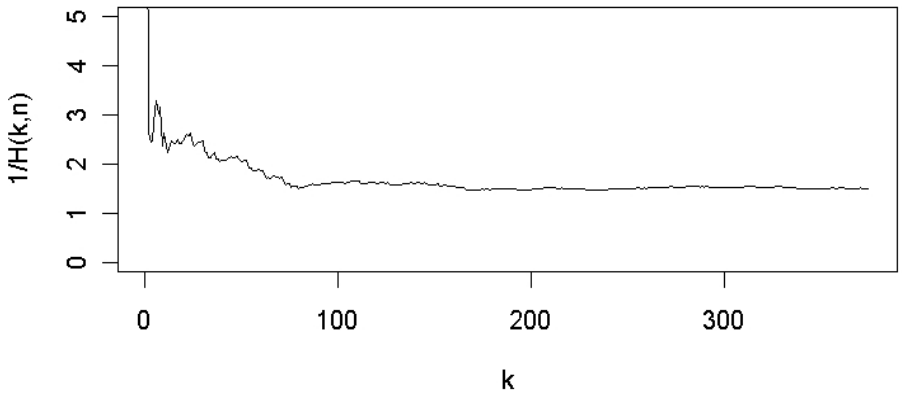

Figure 5.11: Hills Plot ($\alpha = 1.5$) - Clarknet Session Length, HIGH

Table 5.10 shows the summary of values of α. Figures 5.12 and and 5.13 show the bar chart for same data.

Following observations can be made:

1. Web-sites with higher traffic (WVU, Clarknet) have lower value of α ($\alpha < 2$) and have a heavy-tailed distribution for session length indicating that large sessions exist with higher probability for these data-sets. Web-sites with relatively lower traffic (CSEE, NASA-Pub2, NASA-Pvt3 and NASA-Pub3) do not show heavy-tailed behavior and have a value of α greater than 2.

2. From Figure 5.12 we can see that the value of α increases as we move from low to high period i.e. periods with lower traffic show more heavy-tailed distribution of session length.

Table 5.10: α, Session Length - Low, Medium, High and entire week

	LOW	MED	HIGH	Week	Total Requests
WVU	1.02	1.55	1.58	1.8	15,785,164
Clarknet	NA	1.27	1.5	1.8	1,654,882
CSEE	NA	1.73	1.8	2.2	396,743
NASA-Pub2	NA	NA	1.39	2.2	39,137
NASA-Pvt3	NA	NA	NA	2.1	21,799
NASA-Pub3	NA	NA	NA	3.4	5,597
NASA-Pub1	NA	NA	NA	NA	3,641
NASA-Pvt2	NA	NA	NA	NA	3203
NASA-Pvt1	NA	NA	NA	NA	1,163

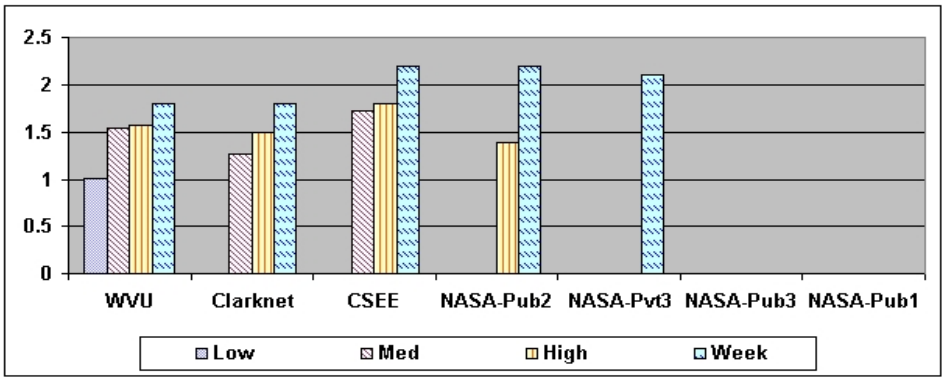

Figure 5.12: Session Length, α value for each data-set

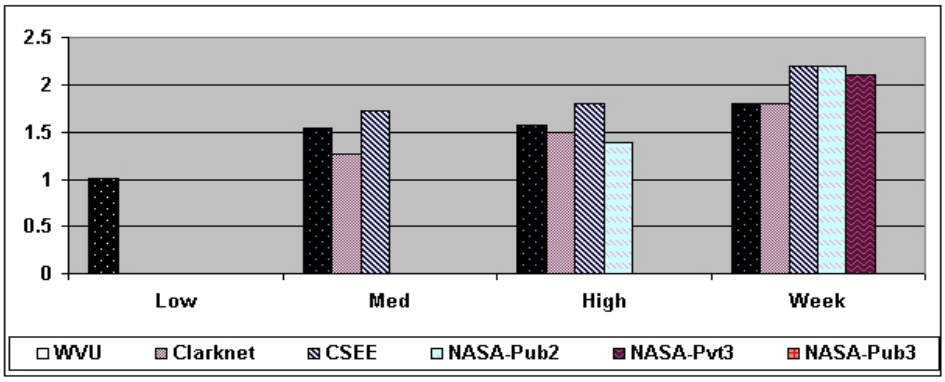

Figure 5.13: Session Length, α value for each period

Summary of Intra-Session Characteristics

In this section we summarize the results for the intra-session characteristics.

1. As compared to other intra-session parameters, bytes transferred per session shows heavier tail for all data-sets for all periods. This can be attributed to the heavy-tailed distribution of file-sizes in the web-server.

2. Figures 5.10 and 5.13 show the value of α for number of requests per session and session length parameter respectively. If we look at the one week data for these parameters we can clearly see that for all the data-sets for which number of requests per session show lower value of α, the value of α for session length is higher and vice versa. Lower value of α for session length for WVU, Clarknet data-sets indicates the presence of longer sessions with significant probability in these data-sets. However, these longer sessions have significant lower number of requests and are not a part of the tail for number of requests. The sessions in the tail of number of requests are completely different from the sessions in the tail of session length. A more rigorous clustering based analysis can confirm this result if it shows two distinct clusters: one having longer sessions with less number of requests and the second having sessions with more number of requests but of shorter duration. However, such an analysis is not within the scope of this thesis.

5.3.2 Inter-session Characteristics

Sessions Initiated per unit Time

Figure 5.14 shows the raw signal for sessions initiated per unit time for one weeks period for WVU data-set. In this section we study the degree of self-similarity of network traffic in terms of number of user sessions initiated per unit time (second). We estimate the value of H, the Hurst exponent using the SELFIS tool [2]. Figure 5.15 shows the sample output of the estimate of Hurst exponent for WVU data-set using seven methods discussed in the Chapter 2 viz: Aggregate Variance, R/S, Periodogram, Absolute Moments, Variance of Residuals, Abry-Veitch, Whittle.

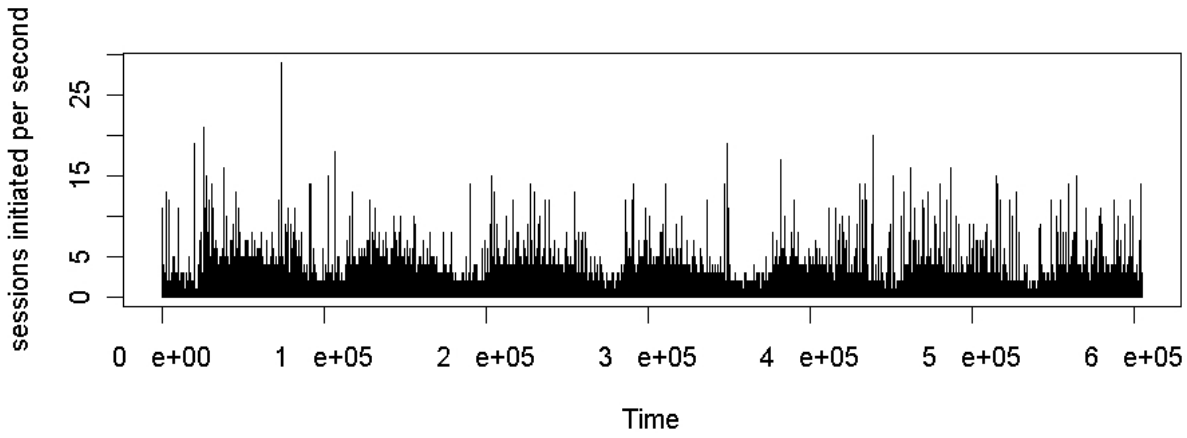

Figure 5.14: Raw data, sessions initiated per unit time WVU

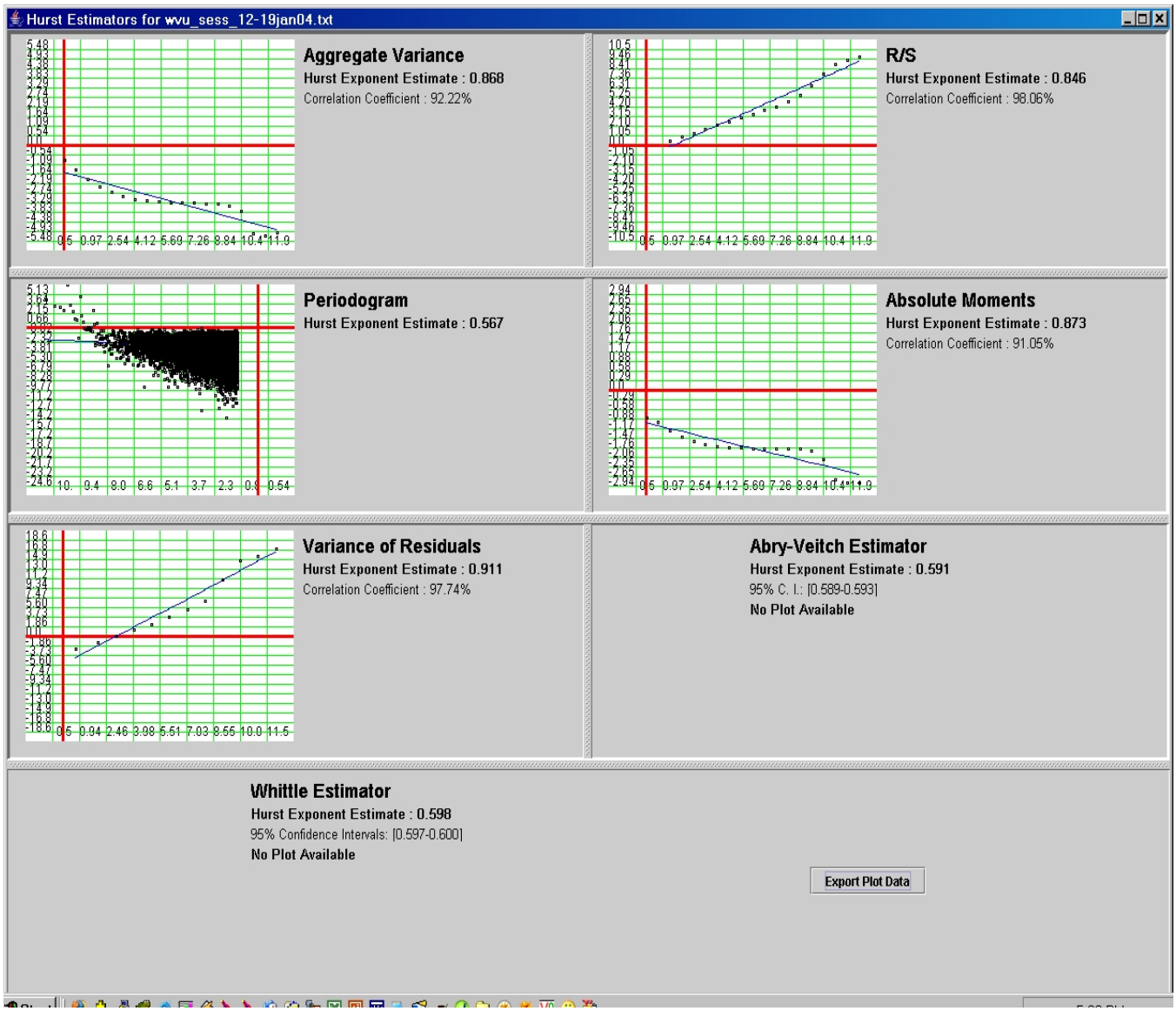

Figure 5.15: Hurst Exponent values - WVU [2]

Table 5.11 shows the summary of estimates of H for different data-sets including the upper and lower bounds for Abry-Veitch and Whittle estimator. As we can see, almost all methods for all data-sets show a value of H in the range [0.5, 1) indicating that the session initiation process is self-similar in nature.

Table 5.11: Hurst Exponent values, Sessions Initiated per second

Data-set	Agg. Var.	R/S	Period-ogram	Abs. Mom-ents	Variance of Residuals	Abry-Veitch Lower Bound	Abry-Veitch	Abry-Veitch Upper Bound	Whittle Lower Bound	Whittle	Whittle Upper Bound
WVU	0.868	0.847	0.567	0.874	0.912	0.591	0.593	0.594	0.597	0.599	0.601
Clarknet	0.813	0.796	0.510	0.820	0.832	0.542	0.545	0.546	0.552	0.553	0.555
CSEE	0.808	0.744	0.554	0.843	1.948	0.553	0.555	0.557	0.533	0.535	0.536
NASA-Pub1	0.561	0.557	0.501	0.729	3.732	0.542	0.544	0.546	0.503	0.504	0.506
NASA-Pub2	0.661	0.594	0.537	0.773	3.003	0.574	0.576	0.577	0.530	0.532	0.533
NASA-Pub3	0.570	0.573	0.514	0.765	3.886	0.554	0.555	0.557	0.510	0.512	0.514
NASA-Pvt3	0.706	0.656	0.497	0.873	3.928	0.536	0.538	0.540	0.505	0.507	0.509

Figure 5.16 shows the same data in table 5.11 in a graphical form with the scale on Y-axis ranging from 0.5 to 1.0. As we can see, Abry-Veitch and Whittle estimator curves are parallel to each other with Abry-Veitch method estimating slightly higher value of H as compared to Whittle estimator. This result is consistent with that shown in [13], which says that Abry-Veitch method over-estimates when compared with Whittle method. Also we can see that the estimates of H using the Periodogram method are comparable with the estimates from Whittle method. Excluding the WVU and Clarknet data-sets (heavily loaded web-servers) R/S and Aggregate variance method estimate the value of H in the same range as Whittle or Abry-Veitch methods; for WVU and Clarknet data-sets these methods estimate the value of H slightly higher than the other methods.

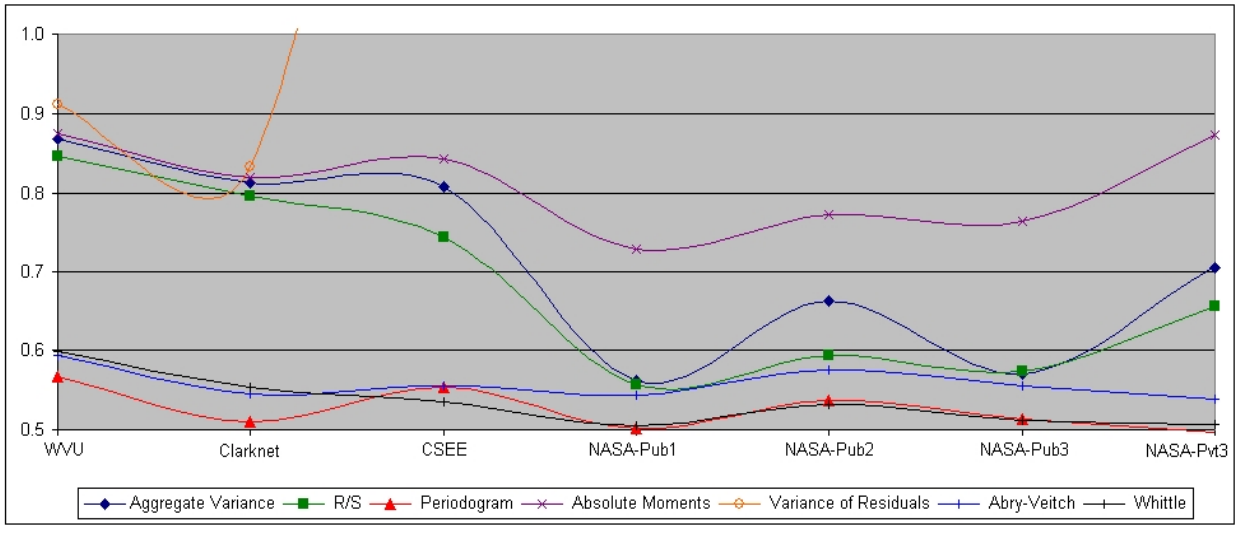

Figure 5.16: Estimate of Hurst Exponent, H (Sessions Initiated per Second)

As with number of requests per second (see section 2.1), the Variance of Residuals method estimates a value of $H > 1$ for most of the data-sets. We ignore the result of this method in the summarized results shown in Table 5.12. The table shows the range and average value of estimate of H. The data in Table 5.12 is sorted by the total number of sessions during the weeks period. We can clearly see that as the number users (sessions) increase, the value of H increases. For heavily loaded WVU, Clarknet and CSEE servers $0.65 < H < 0.75$; for moderately loaded NASA-Pub2 and NASA-Pvt3 servers $0.60 < H < 0.65$; for lightly loaded NASA-Pub2 and NASA-Pub1 $H < 0.6$.

Table 5.12: Range and Average for H Sorted by Total Number of Sessions

	Total Sessions	Range of H	Average H
WVU	188,213	0.567-0.874	0.725
Clarknet	139,745	0.510-0.820	0.673
CSEE	34,343	0.535-0.843	0.673
NASA-Pub2	3,723	0.501-0.729	0.612
NASA-Pvt3	1,076	0.532-0.773	0.630
NASA-Pub1	970	0.512-0.765	0.566
NASA-Pub3	674	0.497-0.873	0.582

More analysis involving aggregation of web traffic over different time scales (session initiated per minute, sessions initiated per hour etc.) and removal of periodicity in the data-sets has been done and published in [3]. Such detailed analysis is not within the scope of this thesis.

Time Between Session Initiations

Users continuously access and leave the web-site. In this section we analyze the time between successive session initiations. This is the difference in time between 2 consecutive sessions from same or different users. Figure 5.17 shows a sample Hills plot for CSEE data-set for one weeks period.

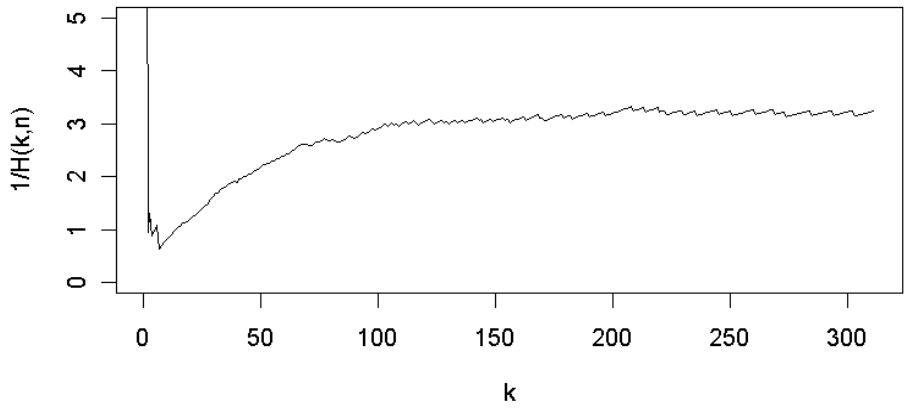

Figure 5.17: Hills Plot ($\alpha = 1.0$) - CSEE Time Between Session Initiations, One Week

Table 5.13 shows the value of α for all the data-sets as obtained from Hills plots. For all the cells in the table marked POISSON, the distribution for this parameter has been found to fit well with Poisson model. These results have been published in [3] and are presented in this thesis for sake of completeness. We did not estimate the value of α for these data-sets since heavy-tailed distribution is not a property of Poisson process.

For the remaining data-sets the value of α is greater than 2 indicating that the distribution of time between session initiations is pareto-like. As indicated in [3] we assume a uniform distribution of this parameter whenever there are more than one sessions initiated during the same second. This uniform distribution assumption can be flawed and can result in inaccurate estimation of α. However, with the web-servers recording requests at one second granularity this cannot be avoided. Apache 2.0 has provisions of recording requests at millisecond granularity and data collected from 2.0 server can result in more accurate estimation of α.

Table 5.13: α, Time Between Session Initiations - Low, Medium, High and entire week

	LOW	MED	HIGH	Week	Total Sessions
WVU	2.9	4.1	NA	5.5	188,213
Clarknet	NA	4.3	4.9	5.5	139,745
CSEE	POISSON	POISSON	1.8	3.2	34,343
NASA-Pub2	POISSON	POISSON	POISSON	2.9	3,723
NASA-Pvt3	NA	POISSON	POISSON	1.9	1,076
NASA-Pub1	NA	POISSON	POISSON	NA	970
NASA-Pub3	NA	POISSON	POISSON	2.3	644
NASA-Pvt2	NA	NA	NA	NA	188
NASA-Pvt1	NA	NA	NA	NA	39

5.4 Challenges in Estimating α, the Index of Heavy-tailed Distribution

In section 3 we showed the estimate of value of α, the degree of heavy-tailedness of a distribution. Estimating α by no means is always trivial, especially if the Hills plot does not stabilize. In this section we discuss a few techniques we used to estimate the value of α correctly with some degree of confidence. We used the following three techniques:

1. Hills Plot with zooming in on the tail of the distribution
2. Smooth Hills and Alternate Hills and Alternate Smooth Hills plots
3. Smooth , Alternate and Alternate Smooth Hills plots combined with LLCD plots

In this section we discuss each of these methods with specific examples.

5.4.1 Hills Plot

In Chapter 2, we discussed the theory behind estimating α, the index of heavy-tailed distribution using Hills plot.

Figures 5.18 and 5.19 show Hills plot for NASA-Pub3 bytes transferred for one week period and Clarknet bytes transferred for one week period respectively.

As we can see from Figures 5.18 and 5.19 the Hills plot clearly stabilizes and the value of α is 1.0 and 2.0.

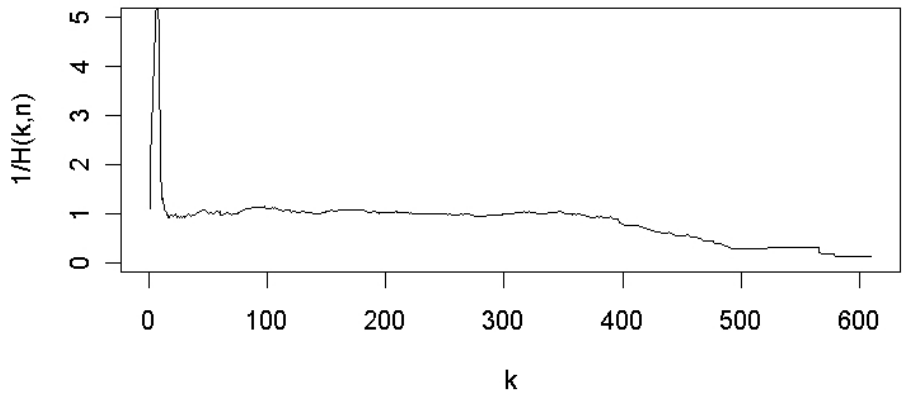

Figure 5.18: Hills Plot ($\alpha = 1.0$) - NASA-Pub3 Bytes Transferred, One Week

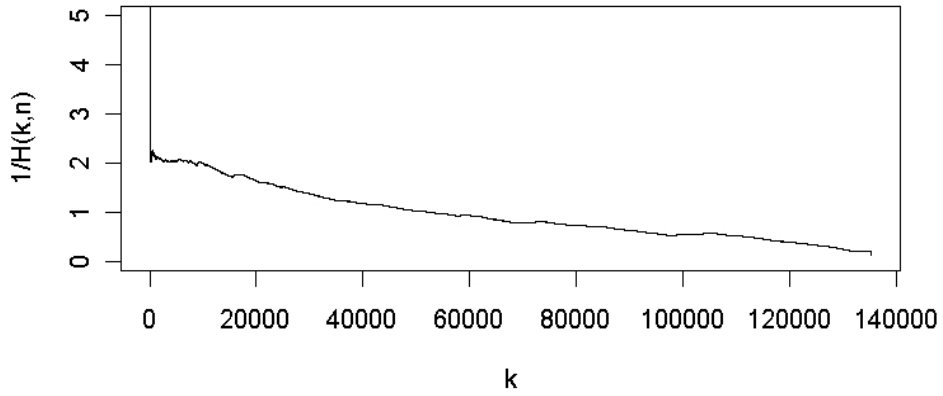

Figure 5.19: Hills Plot ($\alpha = 2.0$) - Clarknet Bytes Transferred, One Week

Sometimes simple zooming in on the upper tail of the distribution in the Hills plot helps in estimating α with confidence. The upper tail essentially is the left part (initial few points) of the Hills Plot.

Figures 5.20 (top) and 5.21 (top) show the Hills plots for CSEE, number of requests per session for one week period and Clarknet, session length for the HIGH period respectively. In the bottom plots of these figures we have zoomed in on the tail of the distribution.

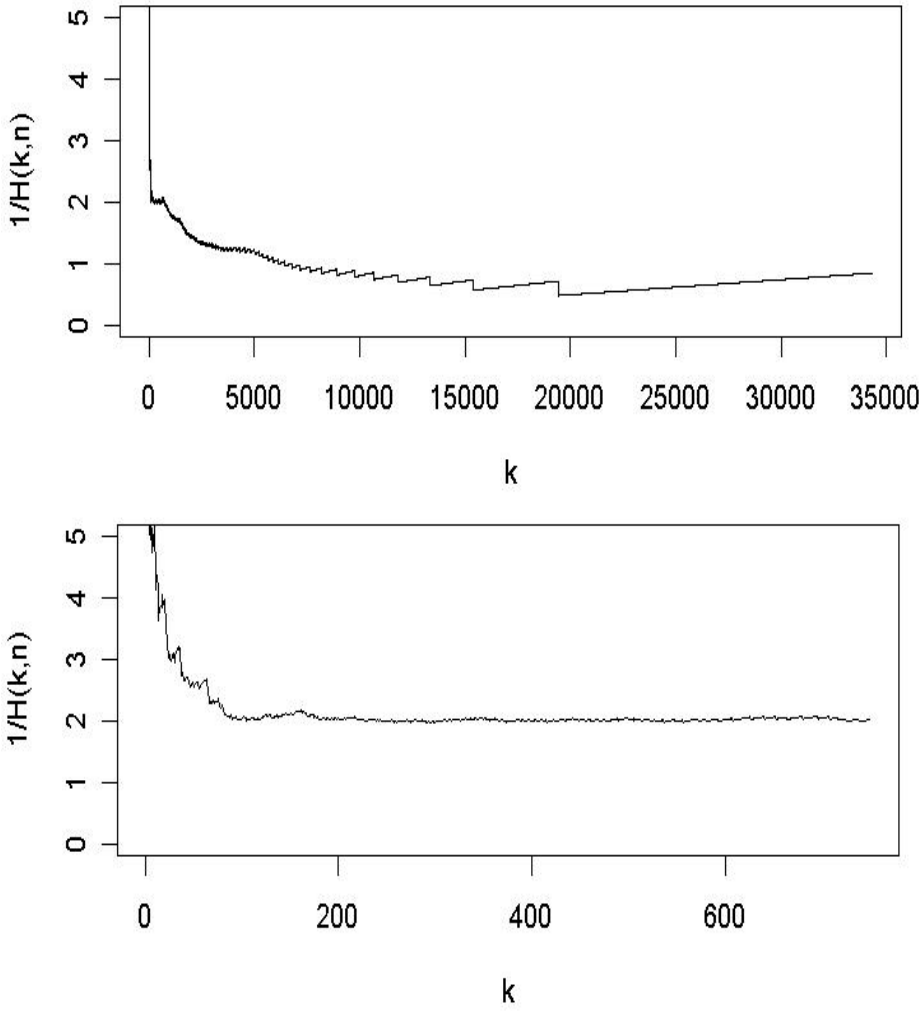

Figure 5.20: Hills Plot ($\alpha = 2.0$) - CSEE Number of Requests per Session, One Week

In Figure 5.20 (bottom) we choose to zoom in on the first 750 points of the distribution. We can now clearly see that the Hills plot stabilizes and can confidently estimate $\alpha = 2.0$ with 2.1 % (750 out of 35000) points in the tail.

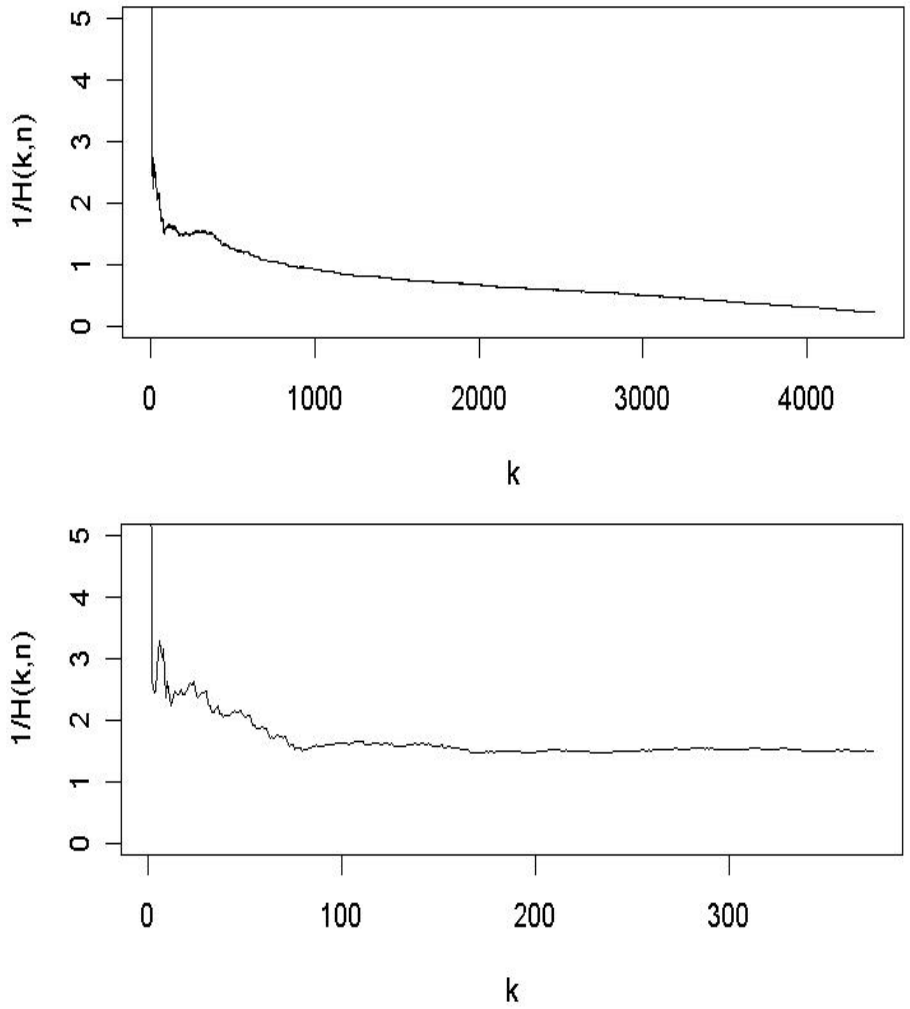

Figure 5.21: Hills Plot ($\alpha = 1.5$) - Clarknet Session Length, HIGH

In Figure 5.21 (bottom) we choose to zoom in on the first 350 points of the tail. We can now clearly see that the Hills plot stabilizes and can confidently estimate $\alpha = 1.5$ with 7.4 % (350 out of 4700) points in the tail.

5.4.2 Smooth Hills, Alternate Hills and Alternate Smooth Hills

In Chapter 2, we discussed the theory of Smooth Hills, Alternate Hills and Alternate Smooth Hills plots. To summarize:

1. *Smoothing* reduces the variability in the Hills plots by considering each point in the Hills plot as the average of a number of points.

2. In *Alternate Hills* plot, we plot the Hills plot in a different scale so that the tail of the distribution is magnified.

3. In the *Alternate Smooth Hills* plot we apply the Alternate Hills technique to the data from the Smooth Hills plot.

Figures 5.22 and 5.23 show the Hills plot (top left), Smooth Hills plot (top right), Alternate Hills plot (bottom left) and Alternate Smooth Hills plot (bottom right) for Nasa-Pub2 request inter-arrival MED period and WVU time between session initiations one week respectively.

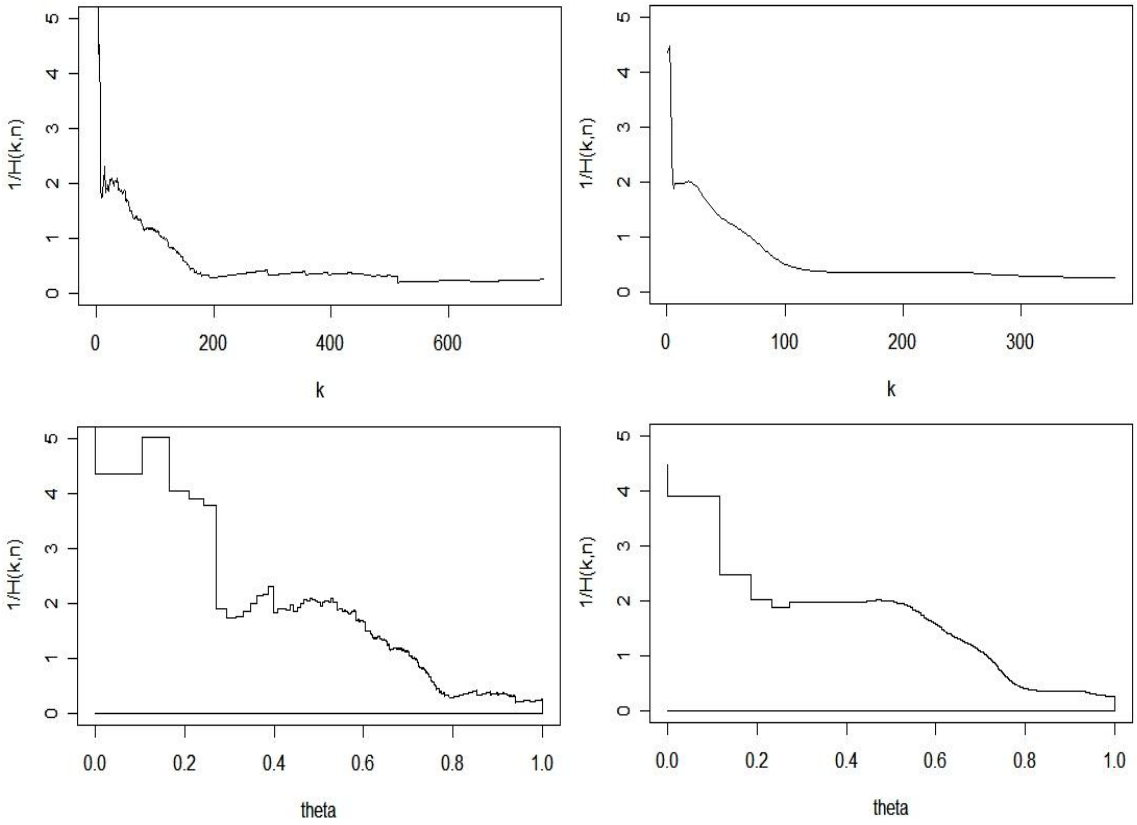

Figure 5.22: Hills, Smth Hills, Alt Hills, Alt Smth Hills plots ($\alpha = 1.9$) - NASA-Pub2 Request Inter-Arrival, MED

As we see in Figure 5.22, Hills plot does not stabilize at all making the estimation of α difficult. Zooming also does not help in this case since we just have around 800 points. The Alternate Hills plot (bottom left) and Alternate Smooth Hills plot (bottom right) stabilize at 1.9 and help getting a correct estimate of α.

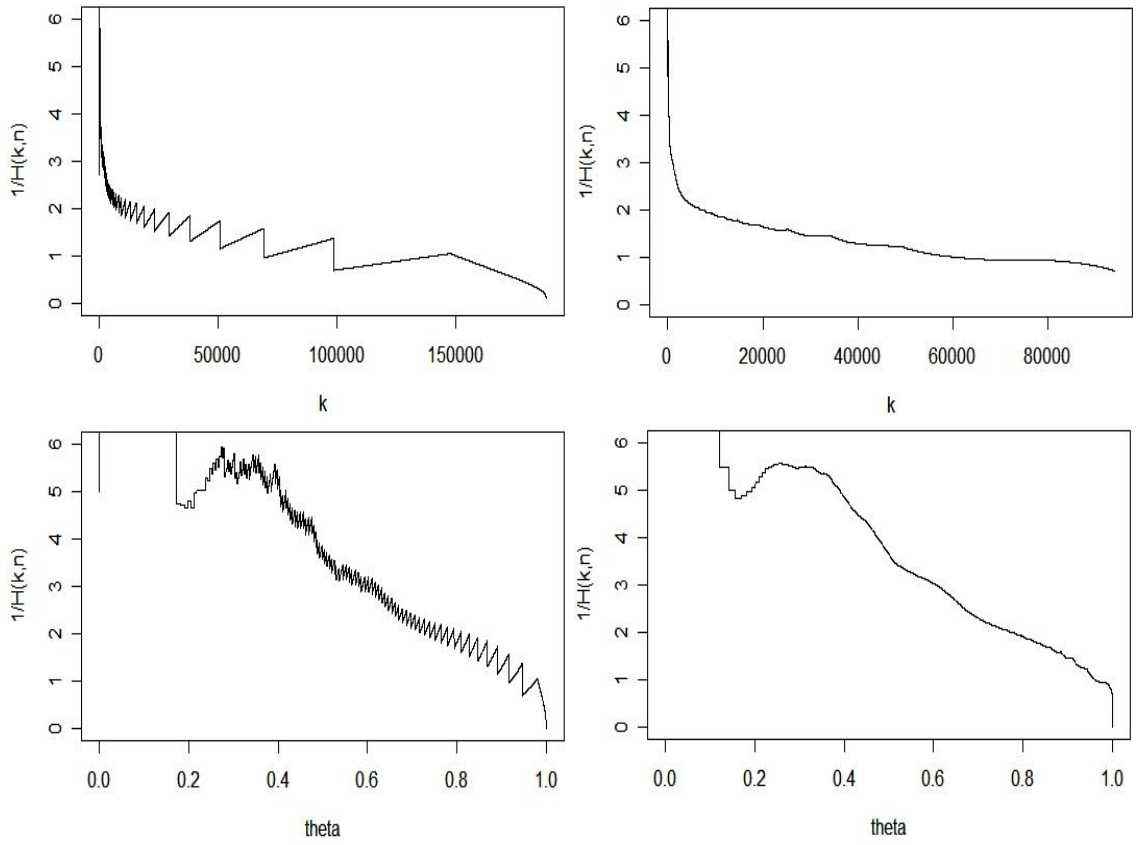

Figure 5.23: Hills, Smth Hills, Alt Hills, Alt Smth Hills plots ($\alpha = 5.5$) - WVU Time Between Session Initiations, One Week

In Figure 5.23 as well we see that the Hills plot does not stabilize. The Alternate Smooth Hills plot (bottom right) stabilize at 5.5 and help getting a correct estimate of α.

5.4.3 Smooth Hills, Alternate Hills and Alternate Smooth Hills combined with LLCD plot data

In the previous section in Figure 5.23 we saw that the Alternate Smooth Hills stabilizes at 5.5 and we choose this as a value of α.

To have some confidence in our technique we use the Log Log complimentary (LLCD) plot and the associated data which consists of:

1. Estimate of α
2. Confidence measure of the linear fitting (R^2) - Closer the value of R^2 is to 1 better is the confidence of the estimate of α.

The estimates of α using LLCD for our data-set can be found in [3].

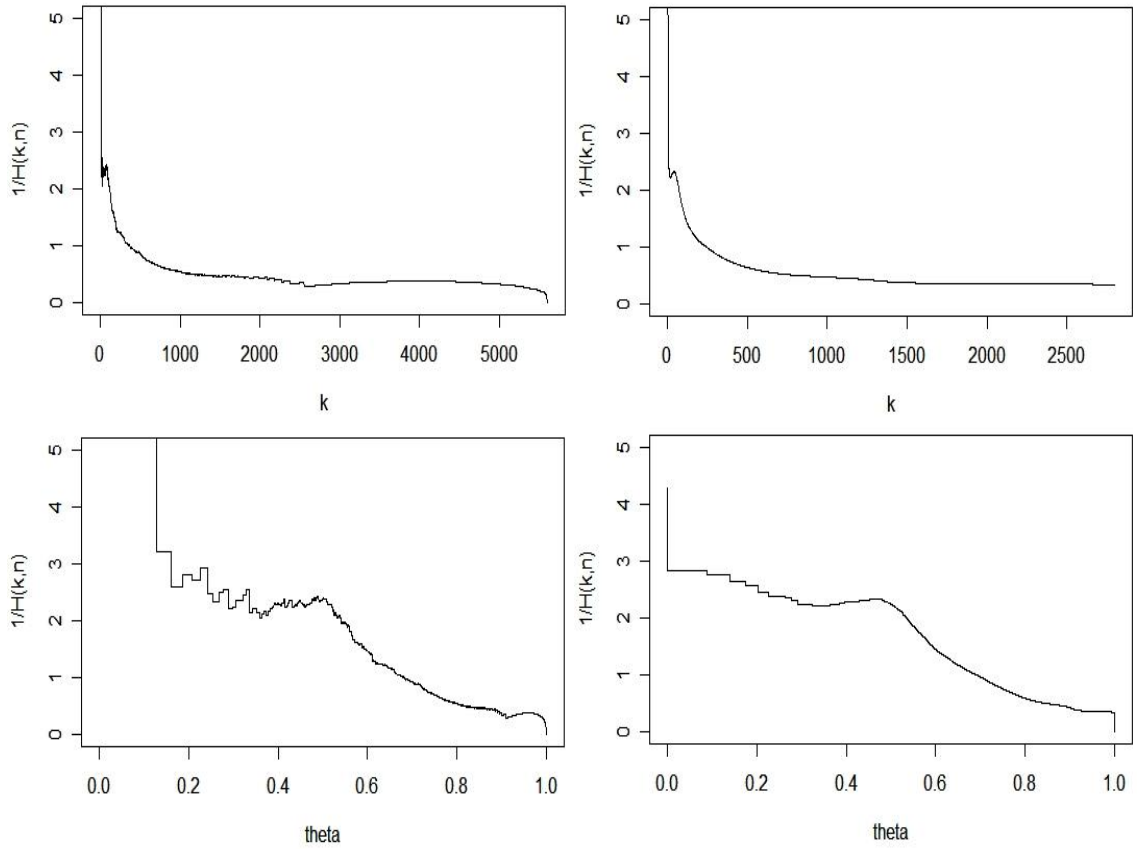

Figure 5.24: Hills, Smth Hills, Alt Hills, Alt Smth Hills plots ($\alpha = 2.3$) - NASA-Pub3 Request Inter-Arrival, One Week

Figure 5.24 shows the Hills, Smth Hills, Alt Hills and Alt Smth Hills plots for NASA-Pub3 request inter-arrival for one week period. We choose $\alpha = 2.3$. The LLCD plot gives an estimate of $\alpha = 2.2$ with the value of $R^2 = 0.986$ [3]. From the data of LLCD plot we can be confident about our estimate ($\alpha = 2.3$).

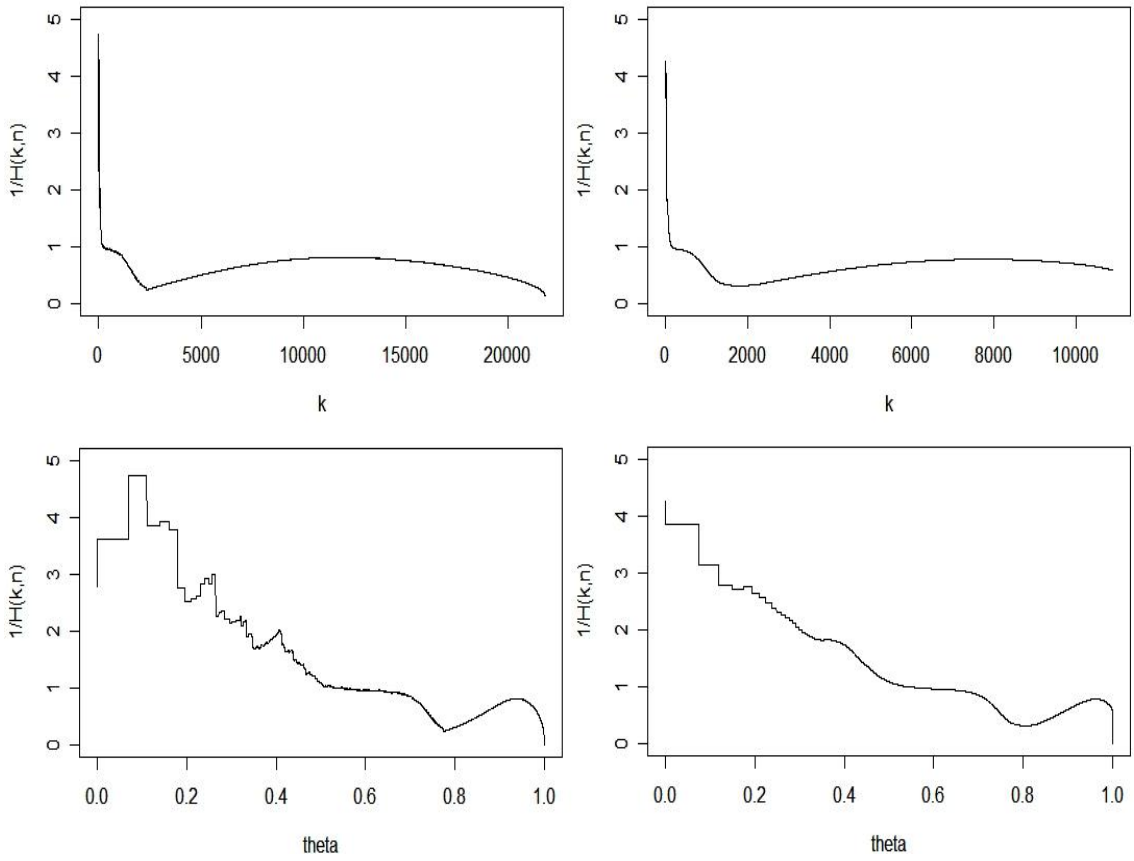

Figure 5.25: Hills, Smth Hills, Alt Hills, Alt Smth Hills plots ($\alpha = 1.0$) - NASA-Pvt3 Request Inter-Arrival, One Week

Figure 5.25 shows the Hills, Smth Hills, Alt Hills and Alt Smth Hills plots for NASA-Pvt3 request inter-arrival for one week period. The Alternate Hills plot (bottom left) shows stabilization at $\alpha = 1.0$. However the Alternate Smooth Hills (bottom right) shows two plateaus ($\alpha = 1.0$ and $\alpha = 1.9$). This leads to confusion. To choose the value of α we resort to the LLCD plot data. We fit in 2 lines in the LLCD plot one corresponding to $\alpha = 1.0$ and the other for $\alpha = 1.9$. The R^2 values for $\alpha = 1.0$ and $\alpha = 1.9$ are 0.974 and 0.95 respectively. Hence we choose $\alpha = 1.0$.

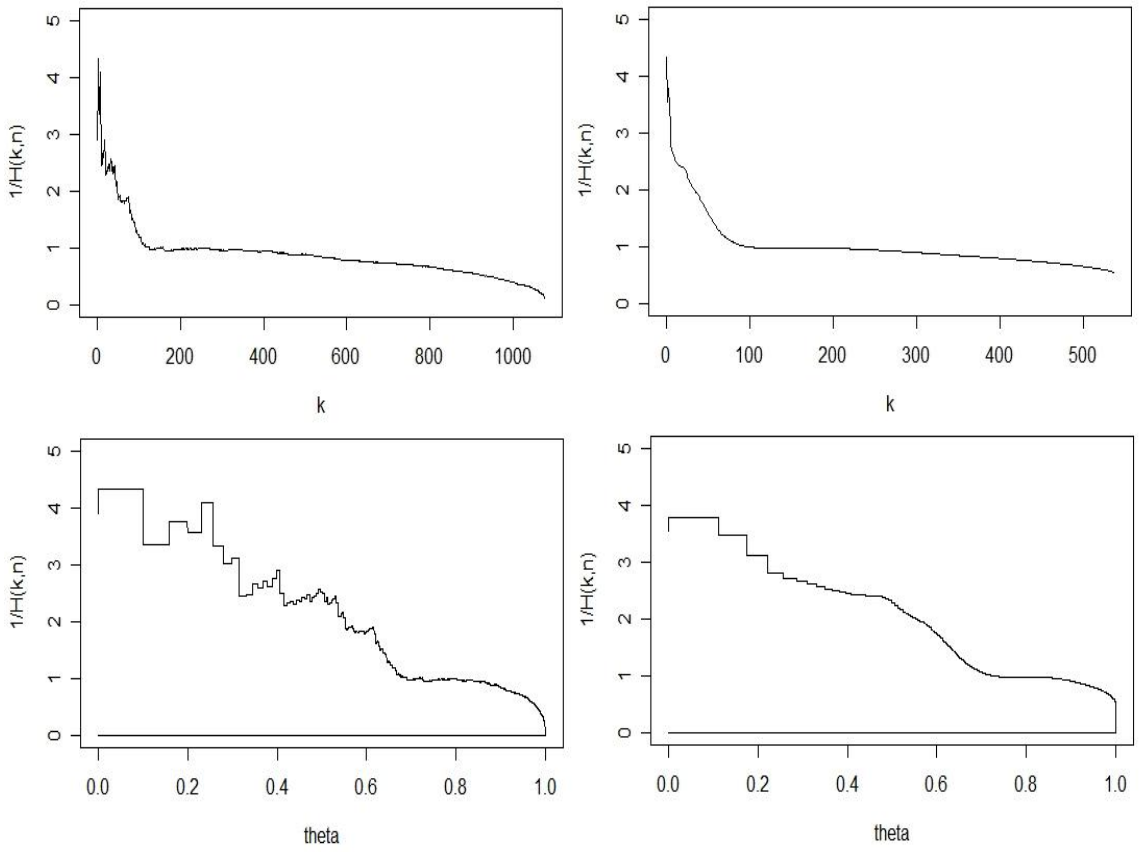

Figure 5.26: Hills, Smth Hills, Alt Hills, Alt Smth Hills plots ($\alpha = 1.9$) - NASA-Pvt3 Time Between Session Initiation, One Week

Figure 5.26 shows the Hills, Smth Hills, Alt Hills and Alt Smth Hills plots for NASA-Pvt3 request time between session initiation for one week period. In this case the Alternate Smooth Hills plot (bottom right) $\alpha = 1.9$. However the Alternate Hills plot (bottom left) shows two plateaus ($\alpha = 1.9$ and $\alpha = 1.0$). Again we resort to the LLCD plot data for choosing α confidently. We fit in 2 lines in the LLCD plot one corresponding to $\alpha = 1.9$ and the other for $\alpha = 1.0$. The R^2 values for $\alpha = 1.9$ and $\alpha = 1.0$ are 0.944 and 0.90 respectively. Hence we choose $\alpha = 1.9$.

5.4.4 Estimating α can still be challenging

In the previous few sub-sections we saw the techniques we used for estimating the value of α. There are certain instances of Hills plot where none of the above techniques help. One such Hills plot along with the Smooth Hills, Alternate Hills and Alternate Smooth Hills plot is shown in Figure 5.27. As we see none of the Hills plots stabilize. The reason for this could be that the

distribution might not be pareto-like or heavy-tailed. We should try to fit some other distribution for this type of data.

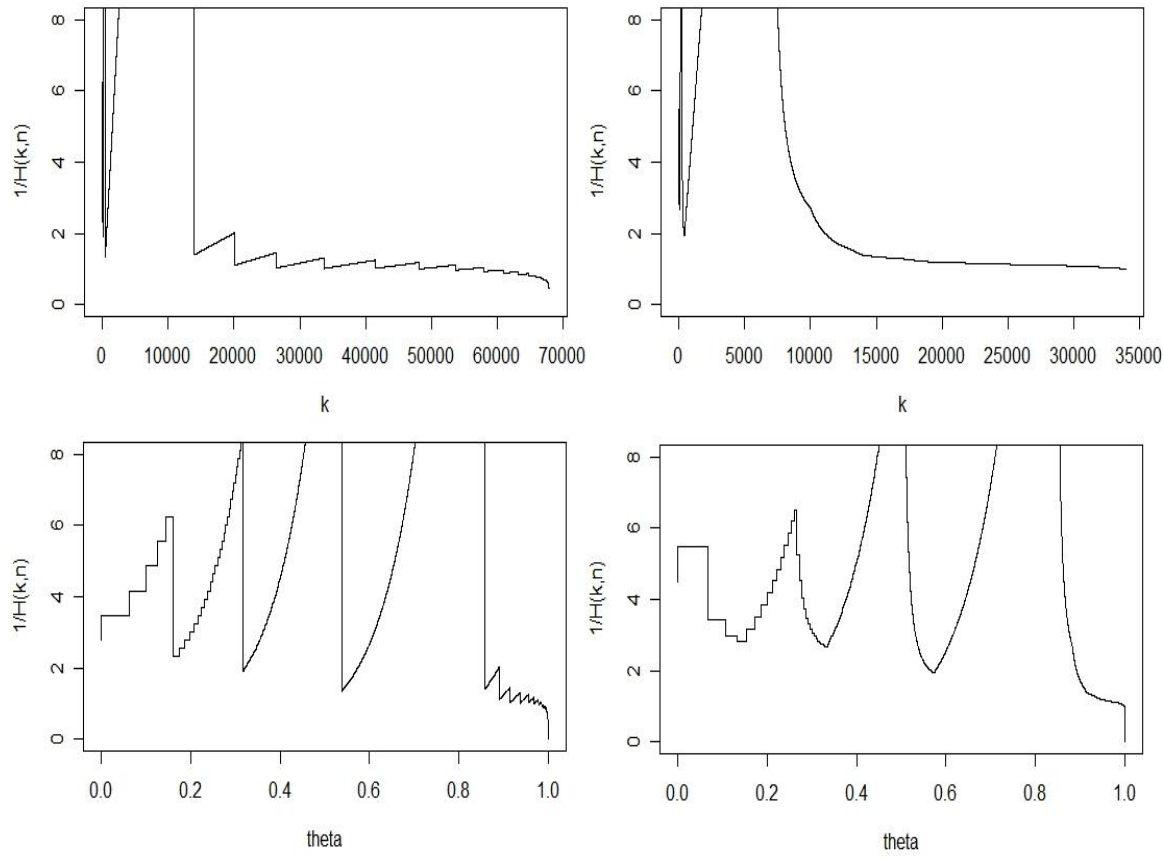

Figure 5.27: Hills, Smth Hills, Alt Hills, Alt Smth Hills plots Not Stabilizing - Clarknet Request Inter-Arrival, HIGH

5.5 Effect of Robots on Session Characteristics

A robot is a program that traverses the web's hypertext structure by retrieving a document and recursively retrieving all the documents that are referenced.

Robots are typically used by search engine web-sites like Google, Altavista. They crawl the web and retrieve data from different web-sites. This data is later processed (indexed and ranked) and is used to provide results for the search queries made by the users of such web-sites. Thus, robots enable a vital function of searching the vast information available on the web. However, from the perspective of a web-server hosting a web-site, robots are agents and just like any other user they generate workload and consume web-server resources. Studying the impact of robots on

the web-server workload characteristics thus becomes important.

Well behaving robots before trying to traverse the web-sites always visit the 'robots.txt' to know what parts of the web-server they can access. This makes the following entry in the web-server's access log file:

```
1.1.1.1597 --- [30/Mar/2004:06:00:50 -0500] "GET /robots.txt HTTP/1.0" 200 23
```

Based on this entry in the log file we identified the robot sessions. To study the effect of these robots sessions on the workload characteristics, we removed these sessions from the data-set and again looked at the distribution of bytes transferred per session, number of requests per session and session length.

Table 5.14 shows the comparison of values of α for the intra-session characteristics with and without robots for one weeks data. A caveat here is that, we have just removed the well-behaving robots (i.e. the robots that access the robots.txt file). There are many robots that do not access the robots.txt file. Identifying such robots is a big challenge.

Table 5.14: Effect of Robots on α

	Bytes per session		Number of Requests		Session Length	
	α_{Robots}	$\alpha_{AfterRemovingRobots}$	α_{Robots}	$\alpha_{AfterRemovingRobots}$	α_{Robots}	$\alpha_{AfterRemovingRobots}$
WVU	1.4	1.4	2.1	2.1	1.8	1.8
Clarknet	2.0	2.0	2.6	2.6	1.8	1.8
CSEE	1.0	NA	2.0	NA	2.2	NA
NASA-Pub1	1.15	1.65	1.3	1.85	NA	NA
NASA-Pub2	1.1	1.3	1.9	2.0	2.2	2.2
NASA-Pub3	1.0	1.2	1.4	1.7	NA	NA
NASA-Pvt3	1.2	1.87	1.6	1.9	2.1	2.1

Following observations can be made from the table:

1. Robots have an impact on the degree of heavy-tailedness on bytes transferred and number of requests per session for all NASA web-sites. As seen in the table 5.14 the tail becomes less heavy (i.e. α increases)after removing the robot sessions from the data-set. In other words, robots make these sessions characteristics more heavy-tailed. This is because most of the robot sessions were present in the upper tail of the data-set. However, observe that the range of α for these parameters remain the same after removing the robots sessions i.e. α lies in the same range (0,1], (1, 2) or [2, ∞) as with or without robot sessions present in the data-set. We can thus infer that though robots make these session parameters more heavy-tailed, the behavior as far as infinite mean and infinite variance remains unchanged.

2. Well-behaved robots do not seem to impact the degree of heavy-tailedness for session lengths for NASA web-sites (NASA-Pub2 and NASA-Pvt3). For both these web-sites robot sessions were not present in the upper tail but were a par of the body, thus, having no impact on degree of heavy-tailedness.

3. Robots did not impact the degree of heavy-tailedness for WVU web-site. This is in spite of the presence of robot sessions in significant numbers in the tail of the data-set. This observation is contrary to the one that we made for NASA web-site. The workload (number of requests and number of sessions) is very high for WVU data-set. We think that though robot sessions contribute towards heavy-tailed distribution of session parameters their effect is not significant when compared to the large number of non-robot sessions that also contribute towards the heavy-tailedness of the distribution.

4. For Clarknet web-site just 2 robot sessions were identified and those robot sessions were not in the tail. Hence, the degree of heavy-tailedness remained unchanged as indicated by value of α after removing those sessions.

5. For CSEE web site we could not detect any robots since there were no requests for robots.txt file. The robots on CSEE web-site, if at all present, were not well-behaved.

Figure 5.28 and 5.29 show the Hills Plot for WVU Bytes Transferred and NASA-Pub1 Bytes Transferred. The plot in the top in both the figures show the Hills plot in presence of robots while that at the shows the Hills plot after removing the robots. We can see that both the Hills plots in figure 5.28 stabilize at the value of $\alpha = 1.4$ indicating that robots do not affect the heavy-tailedness of Bytes transferred per session in WVU data-set. Figure 5.29 shows the Alternate Hills plot for NASA-Pub1 data-set for Bytes Transferred per session parameter. We can see that the top Alt Hills plot stabilizes at $\alpha = 1.15$ while the bottom one where we removed the robot sessions stabilizes at $\alpha = 1.65$.

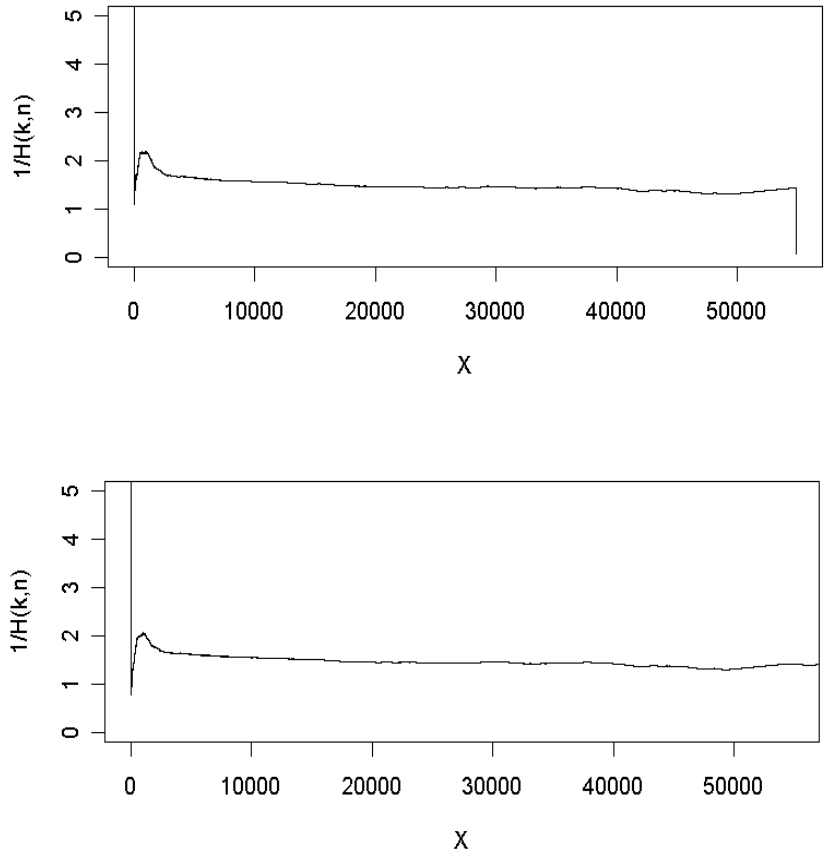

Figure 5.28: Effect of Robot on WVU Bytes Transferred per Session. top - Hills plot with robots, bottom - Hills plot after removing robots

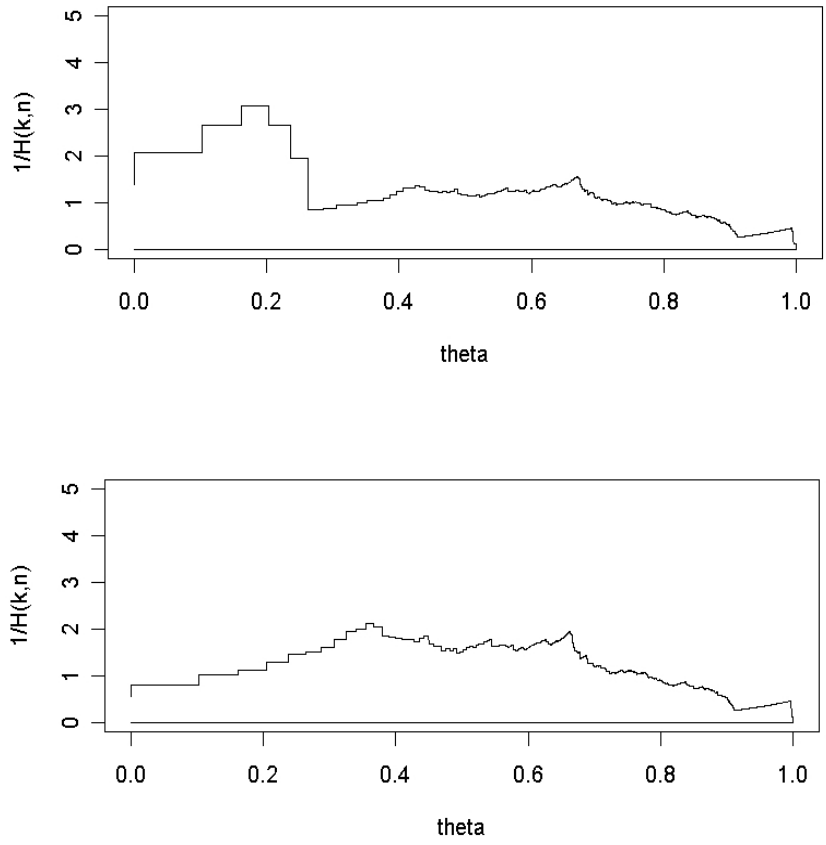

Figure 5.29: Effect of Robot on NASA-Pub1 Bytes Transferred per Session. top - Alt Hills plot with robots, bottom - Alt Hills plot after removing robots

Chapter 6

Conclusion

Understanding the workload characteristics of web server is one of the key steps in the design of an efficient web-server. It provides the basis for developing realistic synthetic workload generators, capacity planning, designing caching mechanisms, admission control policies and accurate predictions of performance measures. Continuously evolving web technology and exponential growth in the number of users makes the workload characterization process all the more important for developing robust web systems.

In this thesis we characterize the web server workload in terms of request and session parameters. We used number of requests per unit time and inter-request arrival time as the request based parameters for workload characterization. Session were characterized using several intra-session and inter-session parameters. Intra-session characteristics include: bytes transferred per session, number of requests per session, and session length. Number of sessions initiated per second and time between session initiations are the two inter-session characteristics analyzed. Based on the data collected from the access logs of nine different web servers, rigorous statistical analysis of these parameters was performed. We explored two important phenomena: self-similarity and heavy-tailed distribution of web-server workload in terms of these parameters.

The results showed that web traffic is self-similar in nature and that the degree of self-similarity is proportional to the workload intensity; higher the workload more self-similar is the web traffic. The rigorous analysis of the intra-session parameters indicated that the tails of the distributions matched with that of the Pareto distribution. Furthermore, many of these parameters exhibited heavy-tails, i.e., infinite variance. Heavytailed behavior of the parameters is a consequence of the

high workload, and the heavytailed distribution of the file sizes on the server. Presence of robots also contributes towards the heavy-tailed distribution of these parameters, especially for the lightly loaded servers.

Our last contribution is in terms of methods for estimating the Hurst exponent, H and the index of heavy-tailed distribution, α. We point out problems in the existing work on estimating H and α accurately. We used a combination of multiple techniques to generate more accurate estimates of these parameters.

6.1 Summary of Results

In this section we summarize in details the results of this thesis.

1. Based on the time series analysis of requests per unit time and sessions per unit time parameter we conclude that web-traffic is indeed self-similar in nature and the degree of self-similarity depends on the workload. Heavily loaded servers like (WVU, Clarknet and CSEE) are more self-similar with higher value of Hurst exponent H as compared to the lightly loaded NASA-Pub1 and NASA-Pub3 servers.

2. Of the intra-session parameters that we analyzed, the distribution of bytes transferred per session is the most heavy-tailed. This is because of the heavy-tailed distribution of file sizes as pointed in [23].

3. We saw that the degree of heavy-tailedness for bytes transferred per session for a particular web-site is independent of the amount of workload and the duration under consideration as shown by almost same value of α for LOW, MED, HIGH 4 hour periods and one weeks period. This conclusion holds true for other intra-session parameters: number of requests per session and session length.

4. For the parameters: number of requests per session and bytes transferred per session, we saw that the distribution is more heavy tailed for lightly loaded NASA servers as compared to the heavily loaded WVU and Clarknet servers. This might be because for the heavily loaded servers the relative probability of high bytes transferred and high number of requests per session (which is the main reason for heavy-tailed distribution) is low. In other words,

for the heavily loaded servers there are very high number of sessions that have low bytes transferred and low number of requests per session that contribute in reducing the degree of heavy-tailedness.

5. For session length we see the exactly opposite behavior as compared to bytes transferred per session and number of requests per session i.e. the heavily loaded web servers show higher degree of heavy-tailedness. In fact, for lightly loaded NASA servers, the distribution of session length is not heavy-tailed (but pareto) as shown by $\alpha > 2$.

6. Based on our analysis of self-similarity in terms of requests per unit time and sessions initiated per unit time we confirm the results published in [13] that Abry-Veitch method for estimation of self-similarity estimates a consistently higher value of Hurst exponent H as compared to the Whittle estimator for all the data-sets. For sessions initiated per unit time the Periodogram method estimates almost the same value as estimated by Whittle estimator.

7. For estimating α the index of heavy-tailed distribution we performed a rigorous analysis based on Hills plot, Smooth Hills plot, Alternate Hills plot and Alternate Smooth Hills plot. We showed that estimating α based on Hills plot is not trivial. We found that zooming into the upper tail of the Hills plot and looking at Alternate Hills and Alternate Smooth Hills helps in more accurate estimation of α. We also found that looking at the LLCD plots and the corresponding R^2 values helps in increasing the confidence in our estimates of α; this especially holds true when the Hills plot shows two plateaus.

8. Finally, based on a preliminary analysis of robots we saw that for lightly loaded NASA web servers robots tend to make the intra-session parameters more heavy-tailed. For the heavily loaded WVU server robots did not impact degree of heavy-tailedness because of the presence of significant number of non-robot sessions that contribute towards the heavy-tail of the distribution. A caveat in this analysis is that we have identified and removed only well-behaving robots. Further work needs to be done to characterize and identify the robot sessions that do not access the robots.txt file.

6.2 Future Work

As future research, we suggest characterizing workload in terms of more parameters such as request inter-arrival time within a session, number of active sessions per unit time and inter-arrival time between sessions from the same user. Some of these parameters can help in characterizing robot sessions and differentiating them from user sessions.

References

[1] Leland W, Taqqu M, Willinger W, Wilson D, "On the Self-Similar Nature of Ethernet Traffic," *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, Feb 1994.

[2] Karagiannis Thomas, Faloutsos Michalis, Molle Mart , "A User-Friendly Self-Similarity Analysis Tool," *Special Section on Tools and Technologies for Networking Research and Education, ACM SIGCOMM Computer Communication Review*, 2003.

[3] Goseva-Popstojanova Katerina, Li Fengbin, Wang Xuan, Sangle Amit, Datla Venu, "Performability of Web Based Applications," Tech. Rep., NASA IV&V, Apr 2005.

[4] Menasce, Almeida, Riedi, Ribeiro, Fonseca, Meira Jr, "In Search of Invariants for E-business Workloads," in *"Proceeding of the Second ACM Conference on Electronic Commerce"*, "2000", pp. "56 – 65".

[5] Arlitt M, Williamson C, "Internet Web Servers: Workload Characterization and Performance Implications," *IEEE/ACM Transactions on Networking*, vol. 5, no. 5, pp. 631 – 645, Oct 1997.

[6] Fowler Henry, Leland Will, "Local Area Network Traffic Characteristics, With Implications for Broadband Network Congestion Management," *IEEE Journal of Selected Areas in Communications*, 1991.

[7] Arlitt M, Jin T, "Workload Characterization of the 1998 World Cup Web Site," Tech. Rep., HP Labs, Oct 1999.

[8] Menascé, Virgilio, Almeida, Fonseca, Mendes, "A Methodology for Workload Characterization of E-commerce Sites," in *EC '99: Proceedings of the 1st ACM conference on Electronic commerce*, New York, NY, USA, 1999, pp. 119–128, ACM Press.

[9] Downey Allen, "Evidence for Long-tailed Distributions in the Internet," in *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, Nov 2001, pp. 229–241.

[10] Barford P, Crovella M, "Generating Representative Web Workloads for Network and Server Performance Evaluation," in *"In Proceedings of the ACM SIGMETRICS"*, ACM, Ed., Nov 1998, pp. 151–160.

[11] Danzig Peter, Jamin Sugih, "Tcplib: A Library of TCP/IP Traffic Characteristics," *USC Networking and Distributed Systems Laboratory TR CS-SYS-91-01*, Oct, 1991.

[12] Karagiannis Thomas, Faloutsos Michalis, Riedi Rudolf, "Long-range Dependence: Now You See It, Now You Don't!," in *In Proc. GLOBECOM '02, Taipei, Taiwan*, November, 2002, pp. 2165–2169.

[13] Karagiannis Thomas, Molle Mart, Faloutsos Michalis, "Long-range Dependence: Ten Years of Internet Traffic Modeling," *IEEE Internet Computing*, vol. 8, no. 5, pp. 57–64, 2004.

[14] Mandelbrot B, *The Fractal Geometry of Nature*, W.H. Freeman and Company, New York, 1982.

[15] Mandelbrot B, "http://www.fortunecity.com/emachines/e11/86/mandel.html," .

[16] Crovella Mark, Bestavros Azer , "Self-similarity in World Wide Web Traffic: Evidence and Possible Causes," *In Proceedings of SIGMETRICS*, May 1996.

[17] Taqqu Murad, Teverovsky Vadim, "On Estimating the Intensity of Long-range Dependence in Finite and Infinite Variance Time Series," pp. 177–217, 1998.

[18] Abry P, Veitch D, "Wavelet Analysis of Long-range-dependent Traffic," *IEEE Trans. on Information Theory*, vol. 44, no. 1, pp. 2–15, Jan 1998.

[19] Resnick Sidney, "Heavy Tail Modeling of Teletraffic Data," *Tha Annals of Statistics*, vol. Vol.25, no. No.5, pp. 1805–1849, Oct 1997.

[20] Willinger Walter, Taqqu Murad, Sherman Robert, Wilson Daniel, "Self-Similarity Through High Variability: Statistical Analysis of Ethernet LAN traffic at Source Level," Apr 1997.

[21] Garrett, Willinger, "Analysis, Modeling and Generation of Self-Similar VBR Video Traffic," *In Proc. ACM SIGCOMM*, pp. 269–280, 1994.

[22] Paxson Vern, Floyd Sally, "Wide-Area Traffic: The Failure of Poisson Modeling," *IEEE/ACM Transactions on Networking*, vol. 3, no. 3, pp. 226–244, Jun 1995.

[23] Park Kihong, Kim Gitae, Crovella Mark , "On Relationship Between File Sizes, Transport Protocol and Self-Similar Network Traffic," Aug 1996.

[24] Menascé, Virgilio, Almeida, Fonseca, Mendes, "Business-oriented Resource Management Policies for E-commerce Servers," *Perform. Eval.*, vol. 42, no. 2-3, pp. 223–239, 2000.

[25] Karagiannis T, Molle M, Faloutsos M, Broido A, "A Nonstationary Poisson View of Internet Traffic," in *"Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies"*, "2004", vol. "3", pp. "1558– 1569".

[26] Goseva-Popstojanova Katerina, Singh Ajaydeep, Mazimdar Sunil, "Empirical Study of Session-based Workload and Reliability for Web Servers," *15th IEEE International Symposium on Software Reliability, Saint-Malo, France*, pp. 403–414, Nov 2004.

[27] Apache Web Server, "http://httpd.apache.org/docs/logs.html," .

[28] DJ Decompiler, "http://www.download.com," .

[29] JExcelApi, "http://www.andykhan.com/jexcelapi/index.html," .

[30] Goseva-Popstojanova Katerina, Singh Ajaydeep, Mazimdar Sunil, Li Fengbin, "Empirical Characterization of Session-based Workload and Reliability for Web Servers," *Empirical Software Engineering Journal, Springer, accepted for publication.*