Modeling Correlated Proxy Web Traffic Using Fourier Analysis

Ghaleb Abdulla, Ali H. Nayfeh, and Edward A. Fox
Phone: (540)231-6931, Fax: (540)231-6075
email: abdulla,nayfeh,fox@vt.edu

Abstract

We analyze the arrival rate of accesses to Web proxy caching servers. The results show that the data display strong periodic autocorrelation. The examined data sets show a consistent behavior in terms of having periods corresponding to daily and weekly cycles that can be explained in terms of daily and weekly cyclic behavior of Web users. While these results confirm the correlation in the network traffic noticed by other researchers, we emphasize that this correlation is periodic. A new approach is introduced to model data that exhibit such characteristics by a combination of Fourier and statistical analysis techniques. The source of high correlation in the data is shown to come from the periodic and hence the deterministic part. Synthesized data that results from this modeling approach is shown to have a long-range dependent and self-similar behavior.

1 Introduction

Proper and correct models of network and WWW traffic are important for simulation studies and for understanding the nature of interactions that appear over the Internet and the Web [1, 2, 3]. Recent studies to understand and model network traffic challenge the traditional Poisson distribution assumption [4, 5, 2, 3]. Leland and Wilson [6] show that simulations using real network traffic produce different results from simulations of synthetic traffic generated using traditional approaches (that ignore long-range dependency in network traffic). Another important features of models created for Web traffic simulation studies is that they should have a period of several weeks or months in order to capture the cache behavior and predict user accesses for studies such as forward caching or prefetching.

Web traffic, which is a major component of Internet traffic, displays a significant variance or burstiness over a wide range of time scales. This is an important distinguishing property of self-similar traffic. It also displays strong *auto-correlation* and *long-range dependence*. The number of accesses per unit time at a certain time t is correlated with values at time $t + \tau$, where τ is an arbitrary time delay. As a result, modeling methods that assume independent and identically distributed data are not suitable for Web traffic data. New methods and approaches that capture the characteristics of Web traffic should be developed. There are several efforts to define methods to generate self-similar traffic for network simulation purposes. Leland et al. [5] describe a methodology for generating bursty and hence self-similar traffic by multiplexing a large number of ON/OFF sources with heavy tailed period lengths. Paxson [3] introduces "a fast Fourier transform method for synthesizing approximate self-similar sample paths for one type of self-similar process". However he warns that "One must use caution in assuming that traffic sources are well modeled using self-similar processes". In Section 6 of the same article he questions "even if network traffic is long-range dependent, are self-similar models sufficient for capturing the long-range dependence?"

There are three main contributions in this paper. First, we introduce a new approach to model traffic that has oscillatory or periodic autocorrelation and is self-similar. Using our suggested approach, we show how to generate synthetic traffic that displays similar burstiness characteristics. Our objective is not only to generate self-similar traffic, but also to identify and use traffic characteristics that can generate traffic that is long-range dependent and self-similar. The methodology is explained by applying it step by step to a specific proxy workload. Note, however, that the steps to model the deterministic part also have been tested successfully on other collected proxy workloads.

Second, by examining the auto-correlation plots for Web traffic, we uncover and explain the long-range dependency noticed in the literature [2, 7]. The auto-correlation function reveals periodic long-range dependency in the examined data. Tests over several workloads show that this periodicity is not arbitrary; it is common to all tested workloads and can be explained in terms of daily and weekly cyclic behavior of Web users. Although, the daily and weekly cycles seem to be intuitive, however the strong and sinusoidal nature of the correlation makes it interesting and useful for modeling. In addition these cycles are the source for Web traffic burstiness and hence they can be the source for web self-similarity. In this paper we go beyond identifying those cycles; we quantify and generate traffic using the user cyclic behavior.

Third, we show that the synthesized traffic has long-range dependent and self-similarity characteristics that are very close to the long-range dependent and self-similarity characteristics observed in the original data set.

We use proxy workloads because proxies are the most attractive solutions for scaling information distribution and copyright management [8]. We encourage caching using proxy servers because it captures locality of reference within the user community. Studies show that proxies have good potential for reducing network loads and thus are important for building scalable architectures, see for example [9].

The rest of this paper is organized as follows. In Section 2 we describe the related work in the literature. In Section 3 we describe the data used in this study, and in Section 4 we describe methods for visualizing and discovering periodicity in the data. In Section 5 we introduce our approach to modeling data with periodic components. In Section 6 we measure the periodic signal in the data, and in section 7 we present the conclusions and outline future work.

2 Background and Related Work

2.1 Network traffic and long-range dependency

Long-range dependency in network traffic was noticed in the ethernet measurement for a local area network (LAN) at Bellcore and for Web proxy and server traffic [7]. Leland et al. [5] show how to model this data with a self-similar process. Beran [10] lists common features that distinguish a long-range dependent process, including both qualitative and quantitative features. The qualitative features include:

a) burstiness or long periods of high level activity are followed by periods of low level activity; and

b) the time series appears to be random when observed using a long period of time, however by examining shorter times one can see trends or cycles.

The quantitative characteristics of a long-range dependent series include:

a) the variance of the sample mean decays to zero at a rate slower than n^{-1}, where n is the number of data points, and

b) the series correlations $\rho(t)$ decay to zero at rate that is proportional to $K^{-\alpha}$, where $0 < \alpha < 1$, and t is a large number.

For more details on this subject, we refer the reader to Beran [10].

2.2 Fourier analysis of time series

The main objective of this analysis is to study the structure embedded within the data in order to develop accurate models of it. The structure can be captured by representing the set of data in terms of sinusoidal functions. Sinusoidal functions are attractive because of their simple behavior under change of time scale.

Periodic and quasiperiodic data When a series can be expressed in terms of a sinusoidal function with one fundamental frequency ω and its harmonics, it is called periodic. However, if the data contains more than one incommensurate frequency and we need more than one periodic component to represent the signal, it is called quasiperiodic. The analysis done here show that the data is quasiperiodic; however, for simplicity we always use the term periodic instead of quasiperiodic.

The rest of this paper focuses on the steps to perform this analysis.

2.3 Web traffic characterization and self-similarity

The main objective of the study reported by Arlitt [11] and Arlitt and Williamson [12] was to identify workload invariants across a set of log files collected from different locations. The authors identified 10 invariants in all collected workloads. Self-similarity was not reported as an invariant across all workloads, however some of the tested workloads appeared to be self-similar.

Crovella et al. [4] used traces collected from Web clients to test for the existence of self-similarity in the client-based Web traffic and to explain the causes for this phenomenon. They conclude that traffic due to WWW transfers shows characteristics that are approximately consistent with self-similarity. They trace the causes of the existence of self-similarity to the basic characteristics of information organization and distribution. Accordingly, they claim that changes in the protocols or machine architecture will not affect self-similarity in Web traffic.

The authors in [7] identified the self-similarity property as an invariant for a set of proxy workloads. However, to our knowledge, no one have done a complete study to model proxy Web traffic in a way that the models can be used in simulation studies.

3 Data Collection and Transformation

We examine four log files collected from different communities. The files were analyzed using a script written in Perl. The script extracts the times of access and then creates a time series of arrival rates per hour for each log file.

Table 1 describes the workloads used in this study, showing dates of collection, duration, and number of accesses. An access is defined as a “GET” or “PUT” HTTP command and is represented in one line in the access-log file of a proxy or HTTP server.

The Korea proxy server log file contains accesses from all over Korea to servers inside and outside the country. It contains accesses from clients in industry, education, and other communities. All accesses to the external and Korea internal internet that use the proxy server are logged.

Workload	Collection date	Duration in days	Number of (K) accesses
Korea	9/2/95	24	1680
Library	1/14/97	62	241
CS	1/29/97	38	87
Engineering	7/12/96	131	440

Table 1: Workloads used in this study

The workload from the school Library contains accesses from clients inside the library to servers outside the school. Accesses are from students who use machines in the library hall or laboratory areas or from library employees. Clients in this workload represent undergraduate and graduate students, faculty, and staff accesses from the library.

Accesses in the CS workload are mainly from graduate students in the department; in addition there are some accesses from the faculty.

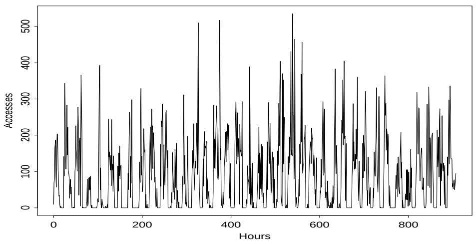
Accesses in the Engineering building workload are from faculty and students working in their offices and laboratories. The building hosts the multimedia and visualization laboratory.

4 Visualizing Periodicity

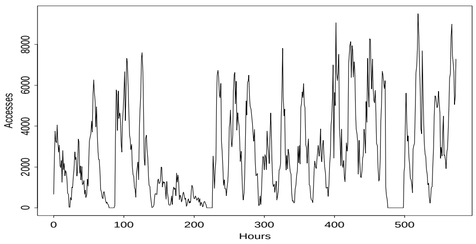
In this section we describe methods to identify what periodicities if any are present in the data. These tools can be used to test for periodic behavior on other data sets and with different time resolutions. We start by examining the time traces for the collected data.

4.1 Time traces

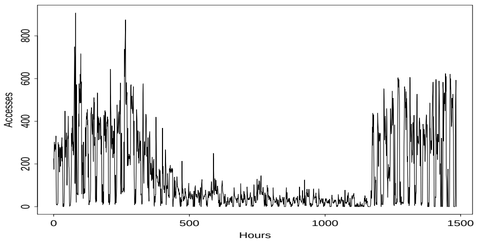
In Figure 1, we show the time traces for four samples of WWW access logs from our set of collected workloads. The graphs look different from each other and some of them suggest that the data is completely random. Figure 1(b) illustrates the shortest workload period: three weeks. The data in this workload seem to have a periodic signal. To explore this further we zoom-in to each of the data sets.



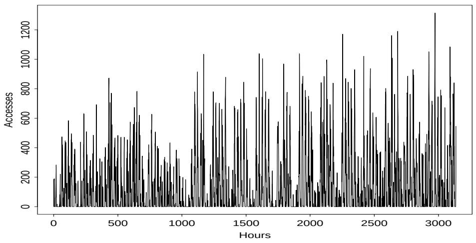
a- CS



b- Korea



c- Library

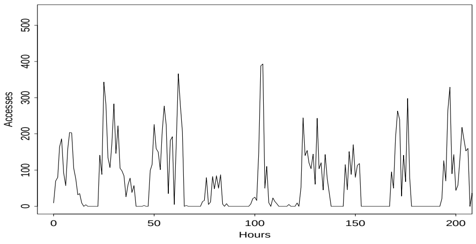


d- Engineering

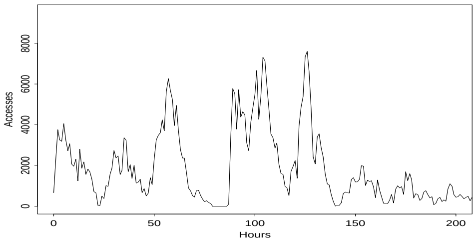
Figure 1: Time traces for the collected data

In Figure 2 we show the first 200 hours from the previous samples. By counting the main peaks in the figures, we note that there are 9 peaks in all of them. This suggests that there is a periodic signal with a period approximately $T = 22.2$, which is very close to 24 hours or a day period. This value is roughly estimated by counting the number of peaks in the figure. We will see in section 5 how to extract the correct periods.

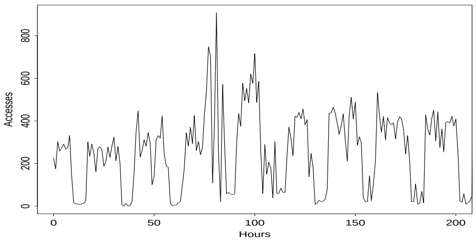
Although inspection of the time series visually can help identify strong periodic signals, there might be some hidden periodicities that are not easily identified by such tests.



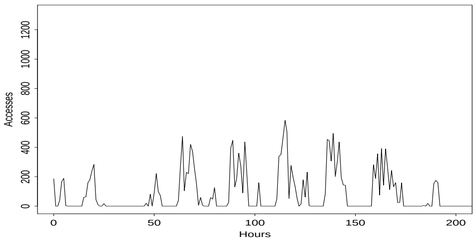
a- CS



b- Korea



c- Library



d- Engineering

Figure 2: First 200 hours of the time traces

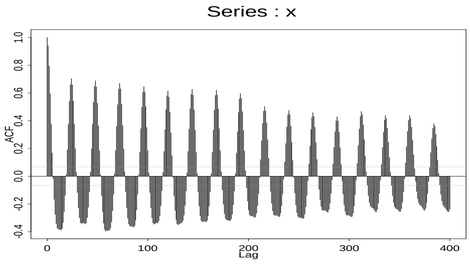
4.2 Auto-correlation function

The autocorrelation function $R_{xx}(\tau)$ is a measure of correlation between $x(t)$ and $x(t+\tau)$, where τ is a time delay [13, 14, 15]. For random signals, $R_{xx}(\tau)$ decays to zero as τ increases. However for a signal with a periodic component, $R_{xx}(\tau)$ does not decay to zero and is oscillatory. Using $R_{xx}(\tau)$ will give a less confusing representation of the data than examining the time trace, especially when there is a high-frequency component in the signal.

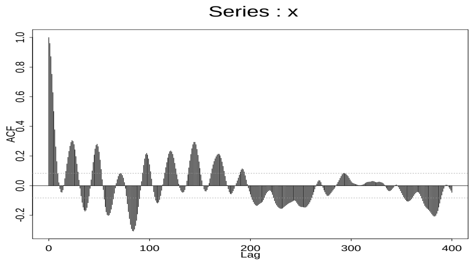
As we showed in the previous section, the data has a periodic component. However, we could not ascertain if more than one such component is present, and if so what are the frequencies involved.

By examining Figure 3 we clearly see that the number of accesses are strongly correlated and it is not clear whether the correlations decay to zero. Actually in some data (a, c, and d) the correlation gets weaker and then becomes stronger. By examining the figures carefully, we note the daily cycle mentioned previously. In addition a weekly cycle appears to be very strong in the data collected from the Engineering building and the Library and less strong in the case of the CS and Korea data. The reason for this is that the Korea data is for the shortest period and the CS data comes in second in terms of duration. When we have longer periods the weekly cycle appears stronger.

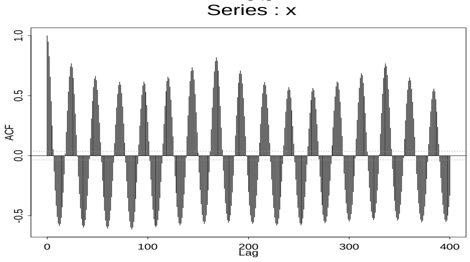
The autocorrelation function in Figure 3 was plotted after characterizing the mean using a moving average program. Plotting the ACF before taking out the mean will give a periodic correlation, however it will be weaker and may be on one side of the x axis especially if the data is not-stationary.



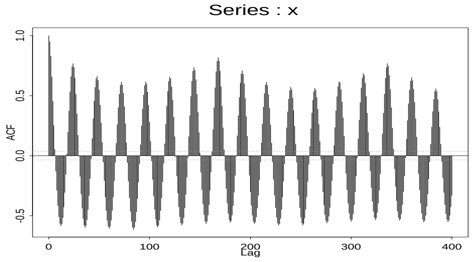
a- CS



b- Korea



c- Library

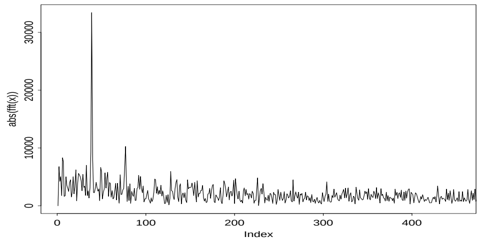


d- Engineering

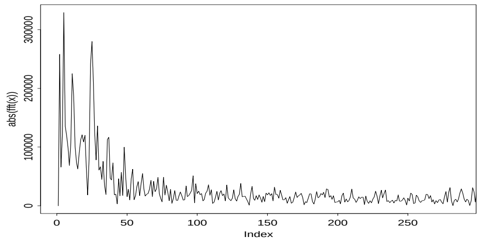
Figure 3: Auto-correlation functions

4.3 Fourier and spectral analysis

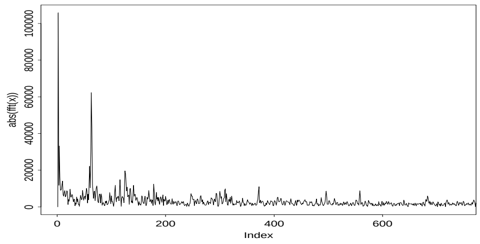
In Figures 4 we show the Fourier spectra for the four sets of data under examination. These plots were made after the data mean was extracted. (We show how to characterize the mean in the next section.) The plot shows half of the data points because the second half is a mirror image of the first half due to *aliasing* [16]. Again we clearly see the peaks that correspond to daily and weekly cycles. We now show how to find the frequencies associated with these peaks. Although the peaks in the Korea data are not isolated and clear as in the other data sets, they exist and can be considered a major component of the data. The reason that the peaks are not very well isolated is that the Korea data set was collected over three weeks only. A longer period will emphasize these periods.



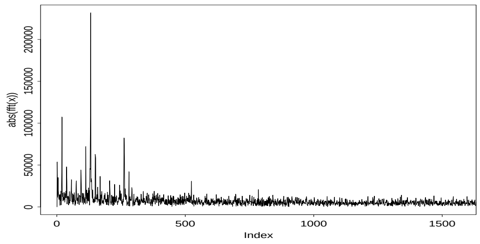
a- CS



b- Korea



c- Library



d- Engineering

Figure 4: The Fourier spectra

We can find the frequencies of these peaks by locating the index for each peak using the FFT command in Splus or any other mathematical software, dividing the index value by the number of points in the data set, and multiplying the result by 2π. To determine the period, we take the reciprocal of the number obtained in the previous step.

Another tool that can be used to visualize periodicity is the autoregressive spectrum but since it gives the same information that is obtained by the Fourier spectra we will not discuss it here. In some data sets the peaks might not be very clear due to *leakage* [16], for example Korea data, in such cases we can use *tapering* [15, 16] to reduce the leakage of power.

5 Modeling Algorithm

The tools described in the previous section can be used to test for the existence of periodic components in the data because they are important in the modeling process.

In Figure 5 we show the steps needed to develop a model for data that contains deterministic components and displays long-range dependency and oscillatory correlation. In some cases we might need to add tapering, or filtering, depending on the characteristics of the data under examination.

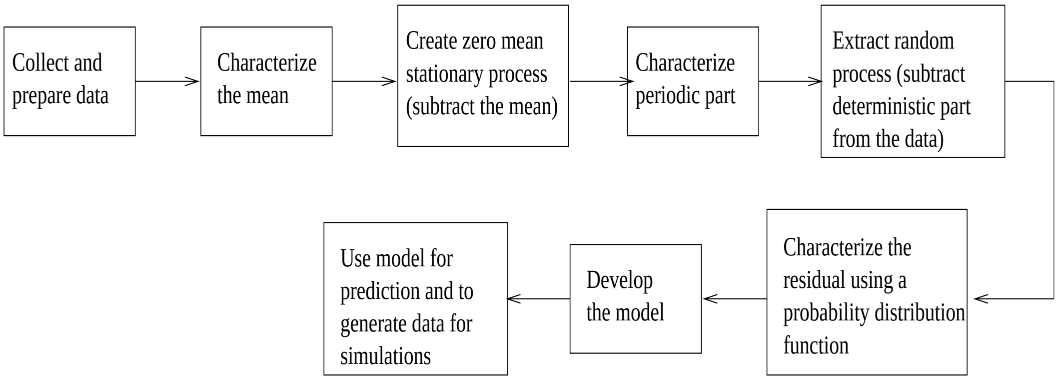


Figure 5: Steps to model the data

5.1 Characterizing the mean

The first step in the modeling process is to characterize the data mean. The mean might be a function of time in a linear or nonlinear fashion. In general, we can identify three main cases for the mean:

- constant mean,
- linear time-varying mean, and
- nonlinear time-varying mean.

For data with constant mean, we calculate the mean by summing all of the values and dividing the sum by the number of points. If the mean varies with time in a linear fashion, it can be characterized using linear regression. If the mean varies in a nonlinear fashion, then we can characterize it using a program that calculates the moving average over a specific window. For our data sets we used the later method. The main objective of this step is to create a zero-mean stationary process from the data set. Non-stationarity in our data can result from the change in the amplitude due to the change in number of users or, can result from the main frequencies phase change. Our data showed constant phase difference for all major frequencies identified. However the change was due to the change in the number of accesses. Hence, the effect of this change can be characterized by a moving average mean.

5.2 Periodic part

The data examined so far displays a strong and periodic correlation. Before building a complete model, we need to determine the amplitudes and periods of all of the periodic components. Finding all of the periods can be done by using Fourier analysis, as described in the previous section. Determining the amplitudes can be done using linear regression [16]. The details are given below.

5.2.1 Determining the frequencies

Frequencies, single or multiple, can be determined using Fourier spectra or autoregressive spectra. From the Fourier spectra we can use the index value for each peak to calculate the period T by dividing the number of data points by the index value. The frequency is easily obtained from the period. The autoregressive spectrum can give the frequency directly.

5.2.2 Determining the amplitudes

A data set that has a periodic signal and a constant mean can be expressed by the equation

$$x_t = \mu + R\cos(\omega t + \phi) + \epsilon$$

where μ is the mean, ω is the frequency of the signal, and ϵ is the residual or random part. This function is quadratic so we cannot use linear regression to find an R that minimizes ϵ. To solve this problem, we let $a = R\cos\phi$ and $b = -R\sin\phi$ [16]. Then, we can rewrite the previous equation as

$$x_t = \mu + a\cos\omega t + b\sin\omega t + \epsilon$$

After the previous transformation, the amplitude for each frequency can be obtained by linear regression over the data using the following generalized linear equation:

$$x(t) = \sum_i a_i \cos\omega_i t + b_i \sin\omega_i t$$

where the a_i and b_i are the required amplitudes (assuming that these amplitudes are stationary), and the ω_i are the frequencies associated with the peaks. This equation can be used to model data with multiple frequencies.

For the CS data, the peaks are at $\omega_1 = 1/24$, $\omega_2 = 1/12$, and $\omega_3 = 1/151$. For the day and half day cycles we can see that the numbers are exact for the corresponding periods. The week period is not exact (151) because of the length of the workload. In the longest workload the calculated weekly period from the FFT was 165.2 which is closer to the expected value of 168. This is due to the variation from week to week; when we have longer duration the effect of these variations is minimized. The first period corresponds to the daily cycle, the second corresponds to the periods of silence at night, and the third corresponds to a cycle every 6.3 days which is approximately a week. By using linear regression, we found the following expression to represent the periodic part of the data:

$$x_{per}(t) = 58.98\sin((2\pi t)/24) - 51.61\cos((2\pi t)/24) + 17.41\sin((2\pi t)/12) - \\ 21.18\cos((2\pi t)/12) - 4.51\sin((2\pi t)/151) - 16.75\cos((2\pi t)/151) \quad (1)$$

In Figure 6 we compare the Fourier spectra for the generated data using equation 1 and the collected data. The figure indicates that the suggested model captures the main frequencies and their amplitudes.

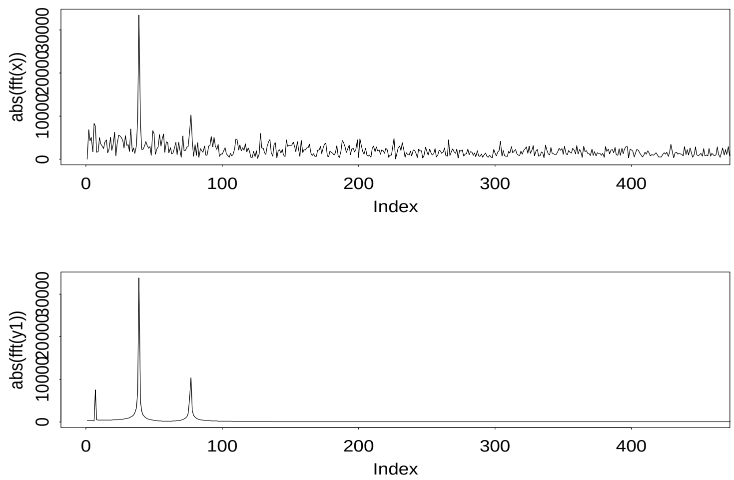


Figure 6: The FFT's of the original data and the generated data set using the periodic model

5.3 Modeling the residual

The main purpose of characterizing the mean and the periodic part is to minimize the residual, which is the non-deterministic or random part of the collected data. Up to this point, we can stop and describe the data in terms of its deterministic component. The significance of characterizing the deterministic components comes from their implications. First, this characterization allows predicting the users' behavior in the future. Second, it provides a nice explanation for the repetitive users' behavior and the time dependent component of the Web data. Third, the same modeling algorithm can be used to characterize document sizes and URL names thus characterizing locality of reference.

Before we perform the statistical characterization of the residual, we test it for the existence of major periods. This test will help identify if the residual still has any periodic components. We generate a data set using the suggested model in equation (1) and subtract the generated data from the original collected data, yielding the residual ϵ. In Figure 7 we show the residual and the associated Fourier spectrum. It appears that the main frequencies have disappeared and that the residual is random.

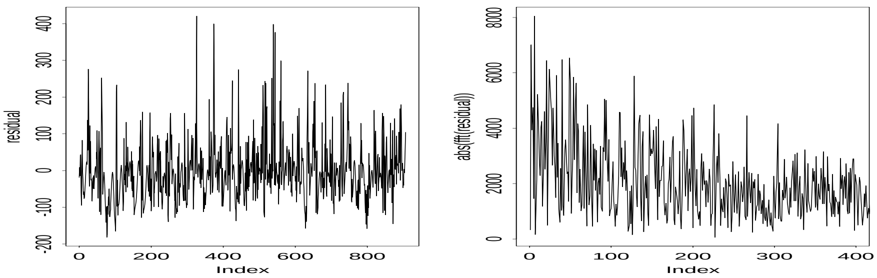


Figure 7: The time trace and the FFT of the residual

5.3.1 Autocorrelation and the residual part

One other test that can serve as a validation step and explain the sources of the strong and periodic correlation in the Web traffic is the autocorrelation function. In Figure 8 we show the autocorrelation function of the residual. The horizontal band about the zero line represents the approximate 95% confidence limits for the null hypothesis that the autocorrelation has zero value. Comparing this figure with Figure 3(a), which contains the autocorrelation function for the data including the periodic component, we find that extracting the periodic part removed the periodic correlation in the data. This means that the main source for the correlation is the periodic behavior of the Web users.

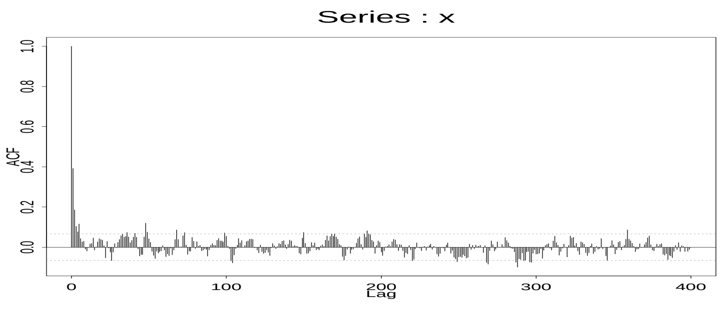


Figure 8: The autocorrelation function for the residual

In the rest of this section, we attempt to model the third component of the data, namely the random part using techniques described in [17].

5.3.2 Fitting the residual

We use quantile summaries and histogram plots to study the distribution for the residual. In Figure 9 we show the histogram and the normal quantile-quantile plots for the residual of the data under examination. The histogram shows that the distribution is asymmetric and the right-hand side of the distribution contains more values than the left-hand side. This implies that the distribution might be a long-tail distribution. The quantile-quantile plot [18] confirms that the distribution is a long-tail distribution from the right-hand side.

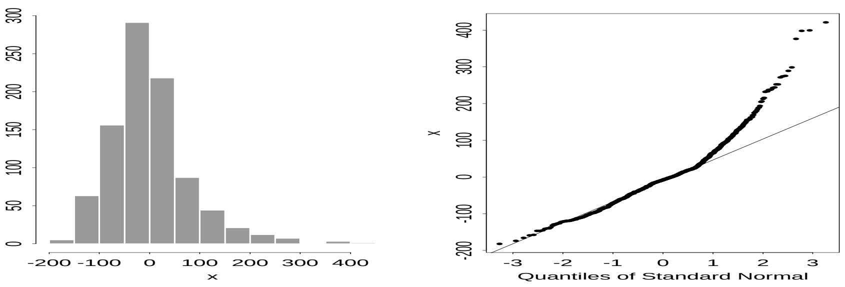


Figure 9: Histogram and Q-Q plot of the residual of the CS data

We use a Weibull distribution to model the residual. A Weibull with $\alpha = 2$ is a good candidate. To generate a synthetic set of data, we used the UnifitII software [19]. The parameters for the distribution were estimated from the data. We compared the probability distribution functions of both the Weibull distribution and the residual and they were very close.

In Figure 10 we show the original data, its Fourier spectrum, the newly generated or synthetic data, and its Fourier spectrum. The main three frequencies and the general behavior of the data are captured in the model. The amplitudes of the negative peaks in the model are clearly higher than the ones in the original data. The reason for this is that the periodic part is symmetric around the zero value while the original collected data does not have the symmetry property.

The final step in constructing the model is to get rid of the excess values from the negative peaks. A crude solution for this problem is to write a filter that will replace all negative values with zeroes.

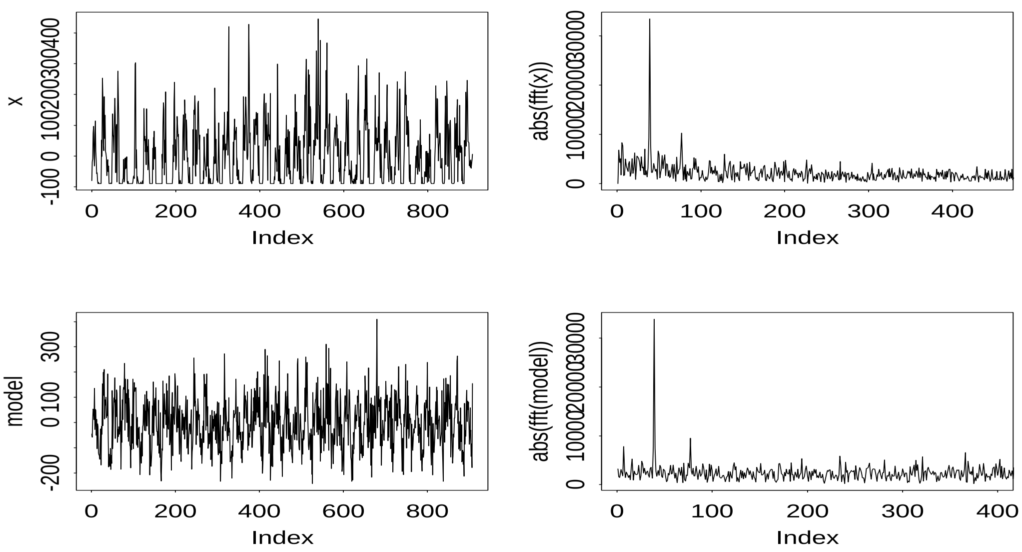


Figure 10: The original data its FFT and the synthetic data and its FFT

We apply this filter after we add the mean to the generated data.

5.4 Model validation

The time series of the synthesized data after applying the filter looks very similar to the time series of the original data. However, we will validate the model using other techniques such as Q-Q plot rather than eyeballing the time series.

In Figure 11 we show the Q-Q plot for the original data versus the modeled data. The relation is almost linear which means that the two sets have close distributions.

In Figure 12 we show the autocorrelation function for the modeled data. Clearly the model displays a periodic autocorrelation and captures the long-range dependent behavior that was observed in the original data. The amplitudes are shorter for the model; this can be because of the applied filter at the end of the modeling process.

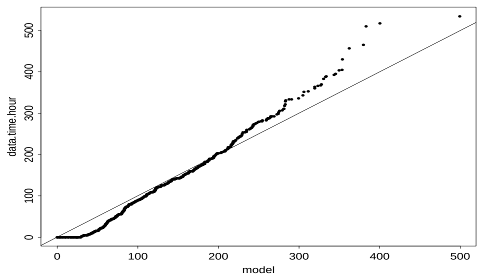


Figure 11: Q-Q plot of the model Vs. original data

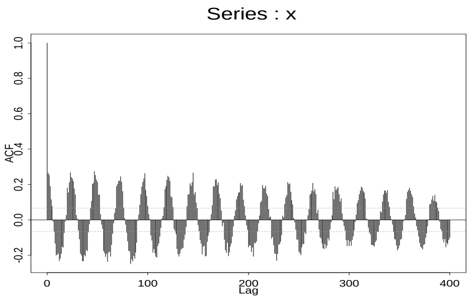


Figure 12: ACF for the synthesized data

5.5 Self-similarity in the generated model

To measure if the synthesized data displays a self-similar behavior we user the R/S statistical test to obtain a value for the Hurst Parameter (H). We run the test on both the original and the synthesized data. The H value for both data sets was 0.6 and this means that they have very close self-similar characteristics. The nice thing about our modeling approach is that by focusing on the network traffic characteristics we created a synthetic traffic that is self-similar and long-range dependent.

We have tested the modeling algorithm on the data sets that we have collected and we came up with the following general equation that can capture the changing mean, the periodic part, and the random part:

$$X(t) = \mu(t) + \sum_i (a_i \cos \omega_i t + b_i \sin \omega_i t) + \epsilon$$

6 Measuring Periodicity in the Data

To find out the strength of our periodic signal with respect to the random part, we use a measure that is called *signal to noise ratio* (SNR) [16] defined as,

$$SNR = \frac{mean\ square\ value\ (MSV)\ of\ the\ signal}{mean\ square\ value\ (MSV)\ of\ the\ random\ part}$$

For the periodic signal

$$MSV = \frac{\sum(y_s - \overline{y_s})^2}{n-2}$$

Because $\overline{y_d}$ is zero (since we extracted the mean), the previous equation reduces to

$$MSV = \frac{\sum(y_s)^2}{n-2}$$

With a similar argument we can show that for the random signal

$$MSV = \frac{\sum(y_r)^2}{n-2}$$

Finally we can write $SNR = \frac{\frac{\sum(y_s)^2}{n-2}}{\frac{\sum(y_r)^2}{n-2}}$, which reduces to

$$SNR = \frac{\sum(y_s)^2}{\sum(y_r)^2}$$

For the tested data we found that SNR=0.5639. This is a significant ratio, since the periodic signal is approximately 30% of the full signal.

7 Conclusions and Future Work

Accesses to Web proxies are highly correlated from day to day and from one week to another. A key source for this correlation is the daily and weekly schedule of Web users. Users who share the same time zone should have similar schedules and hence part of their accesses should reflect it. This result is important for commercial proxies since they will act similarly to a local *broadcasting* station or local cable company that distributes information and other kinds of media. Proxies will be installed to support a group of users in certain locality. Peak and minimum usage hours can be identified for upgrades, price commercials, and other time dependent jobs.

This result is also important when characterizing accesses to busy Web servers. The traffic to such servers can be split based on the time zone of the clients accessing the server. The characteristics of sub groups can be modeled using the suggested approach.

The discovered periodic correlation indicates that Queuing models are not are not the appropriate techniques to model such systems. Simulations need more accurate models that represent the real characteristics of the Web traffic. We introduce a new modeling approach that captures the real characteristics of the Web traffic by using a combination of Fourier analysis and the traditional techniques of statistical analysis. The new approach characterizes the random and the deterministic parts of the data. The generated data displays long-range dependent and self-similar characteristics comparable to the ones observed in the original data. We demonstrate the modeling approach with an example and we validate each step by comparing the generated data to the original one.

By modeling the deterministic component, we generated a synthetic deterministic component and subtracted it from the original data. We tested the residual from the previous step for any relevant frequencies or autocorrelation. We found that the periodic behavior and the autocorrelation are not significant in the residual. It appears that the source of the strong correlation in our data sets is repeated user behavior. In an initial work [7] we suggested that the sources for Web accesses self-similarity might be this correlation. We think now that it is the source of self-similarity in the examined data sets.

The issue of transforming back the modeled arrival rate into arrival times is addressed in detail by Paxson [3]. We will apply similar techniques as part of our future work.

A study we have underway that will help in understanding the nature of Web traffic is to split accesses to a server based on the time zone of the clients and try to see if subsets of traffic have similar characteristics.

We are studying the impact of this behavior on caching in the proxy side. We hope that this study will give some clues about methods to cache documents in the future. To make the study relevant to caching we will test for the existence of correlations between names of accessed URLs, which is categorical data, using a technique developed in [20]. We also will use the same approach to model Web document sizes.

8 Acknowledgments

NSF grants CDA-9312611 and NCR-9627922 partially supported this work. IBM donated equipment to be used for data collection and analysis. Special thanks to Dr. Marc Abrams, Dr. Cary Williamson, and Dr. Jim Pitkow for their valuable comments.

References

[1] Peter B. Danzig and Sugih Jamin. tcplib: A library of tcp intenetwork traffic characteristics. Technical Report USC-CS-91-495, Computer Science Department, University of Southern California, 1991.

[2] V. Paxson and S. Floyd. Wide-area traffic: The failure of Poisson modeling. In *Proc. SIGCOMM94*, pages 257–268, London, August 1994. ACM.

[3] V. Paxson. Fast, approximate synthesis of fractional gaussian noise for generating self-similar network traffic. In *CCR, submitted for publication.*

[4] M. E. Crovella and A. Bestavros. Self-similarity in World Wide Web traffic evidence and possible causes. In *Proc. SIGMETRICS*, pages 160–169, Philadelphia, PA, May 1996. ACM.

[5] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, 2(1):1–15, February 1994.

[6] W. E. Leland and D. V. Wilson. High time-resolution measurement and analysis of LAN traffic: Implications for LAN interconnection. In *Proc. of IEEE Infocomm '91*, pages 1360–1366, Bar Harbour, FL, 1991. IEEE.

[7] G. Abdulla, E. A. Fox, M. Abrams, and Stephen Williams. WWW Proxy Traffic Characterization with Application to Caching. Technical Report TR-97-03, Computer Sci. Dept., Virginia Tech, March 1997.

[8] T. Winograd. The proxy is where it's at. Technical report, Computer Sci. Dept., Stanford Univ.

[9] M. Abrams, C. R. Standridge, G. Abdulla, S. Williams, and E. A. Fox. Caching proxies: Limitations and potentials. In *4th International World-wide Web Conference*, pages 119–133, Boston, December 1995. <URL: http://ei.cs.vt.edu/~succeed/WWW4/WWW4.html>.

[10] J. Beran. *Statistics for Long-Memory Process.* Chapman and Hall, New York, 1994.

[11] M. F. Arlitt. A performance study of Internet Web servers. Master's thesis, University of Saskatchewan, Saskatoon, Saskatchewan, May 1996.

[12] M. F. Arlitt and C. L. Williamson. Web server workload characterization: The search for invariants. In *Proc. SIGMETRICS*, pages 126–137, Philadelphia, PA, April 1996. ACM.

[13] A. H. Nayfeh and B. Balachandran. *Applied Nonlinear Dynamics.* John Wiley, New York, 1995.

[14] George E. P. Box and Gwilym M. Jenkins Gregory C. Reinsel. *Time Series Analysis.* Prentice-Hall, London, 1994.

[15] StatSci. *S-PLUS Users's Manual, Version 3.2.* MathSoft Inc., Seattle, Washington, December 1993.

[16] P. Bloomfield. *Fourier Analysis of Time Series: an Introduction.* John Wiley, New York, 1976.

[17] A. M. Law and W. D. Kelton. *Simulation Modeling and Analysis.* McGraw-Hill, New York, 2nd edition, 1991.

[18] R. Jain. *The Art of Computer Systems Performance Analysis.* John Wiley, New York, 1991.

[19] S. G. Vincent and A. M. Law. UniFit II: Total support for simulation input modeling. In *Proc. Winter Simulation Conference*, pages 371–376, Arlington, VA, December 1992.

[20] R. Ribler, A. Mathur, and M. Abrams. Visualizing and modeling categorical time series data. In *Symposium on Visualizing Time-varying Data*, volume NASA Conference Publication 3321, Williamsburg, VA, January 1996. ICASE and NASA/LaRC. <URL: http://www.cs.vt.edu/~chitra/docs/95vtvdRMA/95vtvdRMA.html>.