The *synTraff* Suite of Traffic Modeling Toolkits

Ram Balakrishnan Carey Williamson
Department of Computer Science
University of Saskatchewan
E-mail: `carey@cs.usask.ca`

Abstract

This paper describes three visually interactive tools for the analysis, modeling, and generation of long-range dependent (LRD) network traffic. The synTraff *toolkit uses a three-step modeling approach based on F-ARIMA processes to generate* monofractal *traffic; the* WsynTraff *toolkit implements the Wavelet-domain Independent Gaussian (WIG) model from the literature for representing* multifractal *traffic; and the* MsynTraff *toolkit models multifractal traffic using the Multifractal Wavelet Model (MWM) from the literature. All three tools can be used to generate synthetic LRD traffic streams for use in network simulations.*

1. Introduction

Network traffic modeling is an essential part of any simulation-based network performance study, such as a capacity planning study. Accurate representative workloads are crucial for these studies, for successful realization of cost-effective, high-performance network designs.

Recent research in network traffic measurement and analysis has shown that network traffic is long-range dependent (LRD) [8]. In simple terms, the LRD property means that the traffic has a non-degenerative autocorrelation structure, and that there is similar-looking burstiness in the traffic structure across many time scales (e.g., four or more orders of magnitude, from milliseconds to minutes or more).

Significant early work on LRD network traffic was done by Leland *et al.* [8]. Their paper showed that Ethernet LAN traffic exhibited the mathematical property of *self-similarity* (a special type of LRD behaviour). The property of network traffic self-similarity is quite ubiquitous: it has since been confirmed as present in WAN TCP/IP traffic [11], compressed video traffic [5], signalling traffic, Web traffic, frame relay traffic, and ATM network traffic.

More recent research work has identified a *multifractal* structure in Internet traffic [4]. In multifractal traffic, the scaling behaviours in traffic are different at different time scales, as opposed to following a global time-scaling relationship as in the special case of *monofractal* (i.e., self-similar) traffic. These differences are due to the combination of hardware-level, protocol-level, and network-level effects in the traffic, each operating at different time scales [3]. The qualitative differences between monofractal and multifractal traffic are summarized in Table 1.

The foregoing network measurement research sparked the development of many mathematical models for correlated and long-range dependent traffic, in an effort to understand impacts on network performance. This work includes TES (Transform-Expand-Sample) processes [6, 7], Fractional Brownian Motion (FBM) [10], cascades [4], wavelets [9, 12], and many others.

The purpose of this paper is to describe a visually interactive traffic modeling front-end that has been built to work with these types of traffic models. In particular, this paper describes the design and operation of three toolkits for the generation and analysis of LRD traffic: *synTraff*, *WsynTraff*, and *MsynTraff*. Each is called a "toolkit" since it provides a common Tcl/Tk interface to a set of related C/C++ programs for traffic generation and analysis. Figure 1 shows the graphical user interface (GUI) for each of these toolkits.

The toolkits can be used to model monofractal and multifractal traffic [1]. The *synTraff* toolkit uses a three-step modeling approach [13] based on F-ARIMA processes to generate *monofractal* traffic, which can be described using the three Norros traffic parameters (m, a, and H) [10]. The other toolkits are for modeling multifractal traffic: *WsynTraff* implements the Wavelet-domain Independent Gaussian (WIG) model [9], and *MsynTraff* implements the Multifractal Wavelet Model (MWM) [12]. All three tools can be used to generate synthetic LRD traffic streams for use in network simulations.

The remainder of this paper is organized as follows. Section 2 provides an overview of the toolkits (e.g., modeling process, inputs, outputs, and system requirements). Section 3 describes the *synTraff* toolkit, Section 4 describes *WsynTraff*, and Section 5 describes *MsynTraff*. Section 6 summarizes the paper and the availability of our toolkits.

Table 1. Key differences between monofractal and multifractal traffic

Monofractal Traffic	Multifractal Traffic
The traffic scaling characteristics are the same across many or all time scales	The traffic scaling characteristics change across time scales
A single parameter H (called the Hurst parameter) is adequate to capture the global scaling characteristics	More parameters are required to capture the scaling characteristics of the traffic across many time scales
The Hurst parameter H is fixed and well-defined for a given traffic time series	The Hurst parameter H can vary with time, even within a given time series
The variance-time plot has a predominantly straight line shape with a slope noticeably flatter than -1	The variance-time plot is piece-wise linear (different slopes at different time scales), or even non-linear

2. Preliminaries

2.1. Overview of toolkits

Our toolkits provide three main functions: *traffic analysis*, *traffic modeling*, and *traffic generation*. The generic term "traffic" here means any set of time series data (i.e., a sequence of observations, ordered in time). In our context, the time series is a network traffic trace representing a count of the number of bits (or bytes or packets) transmitted on a network per fixed interval of time. We assume such values are non-negative, and have been obtained by preprocessing a collected network traffic trace to compute and extract the time series data of interest. We use the term "empirical trace" to refer to such a time series extracted from measurements of an operational network (e.g., the LBL-TCP traces of TCP/IP traffic from Lawrence Berkeley Laboratories [11], or the Bellcore Ethernet LAN traffic traces [8]), and the term "synthetic trace" to refer to an artificially generated trace (i.e., generated using a statistical traffic model). All three traffic generation kits are intended for the modeling and analysis of *stationary* time series (i.e., those that do not have significant upward, downward, periodic, or cyclic trends in the data).

By "traffic analysis", we refer to the process of taking an input time series data set, and analyzing it using graphical and/or statistical methods. For this purpose, the toolkits offer five capabilities: traffic profile plots, marginal distribution (frequency histogram) plots, autocorrelation function plots, R/S pox plots, and variance-time plots (see Figures 2 and 3). The latter three analyses are particularly useful for characterizing long-range dependent traffic [8].

By "traffic modeling", we refer to the process of extracting the salient characteristics of a trace (e.g., mean, variance, correlation structure) and internalizing it into an approximate statistical model. For *synTraff*, these characteristics are reflected in three traffic parameters [10]: the mean bit rate m (expressed in bits per second), the Norros variance coefficient (the variance-to-mean ratio of the traffic at the 1.0-second time scale granularity, expressed in bit-sec), and the Hurst parameter H (a unitless parameter in the range $0.5 \leq H < 1$). For *WsynTraff* and *MsynTraff*, the traffic characteristics are captured using wavelet coefficients and scaling coefficients in the wavelet-based models.

By "traffic generation", we refer to the process of taking the internalized statistical traffic model and generating a synthetic time series (i.e., a sample path) that has similar statistical behaviours as the original time series data. Multiple time series can be generated (one at a time), each with a different random number seed. Each run produces an independent realization of the statistical traffic model (i.e., a time series), which can then be saved to a file and used (for example) in simulation-based network performance studies.

All three toolkits support similar traffic analysis functionality; the toolkits differ primarily in how they do the internal traffic modeling and generation.

2.2. System requirements

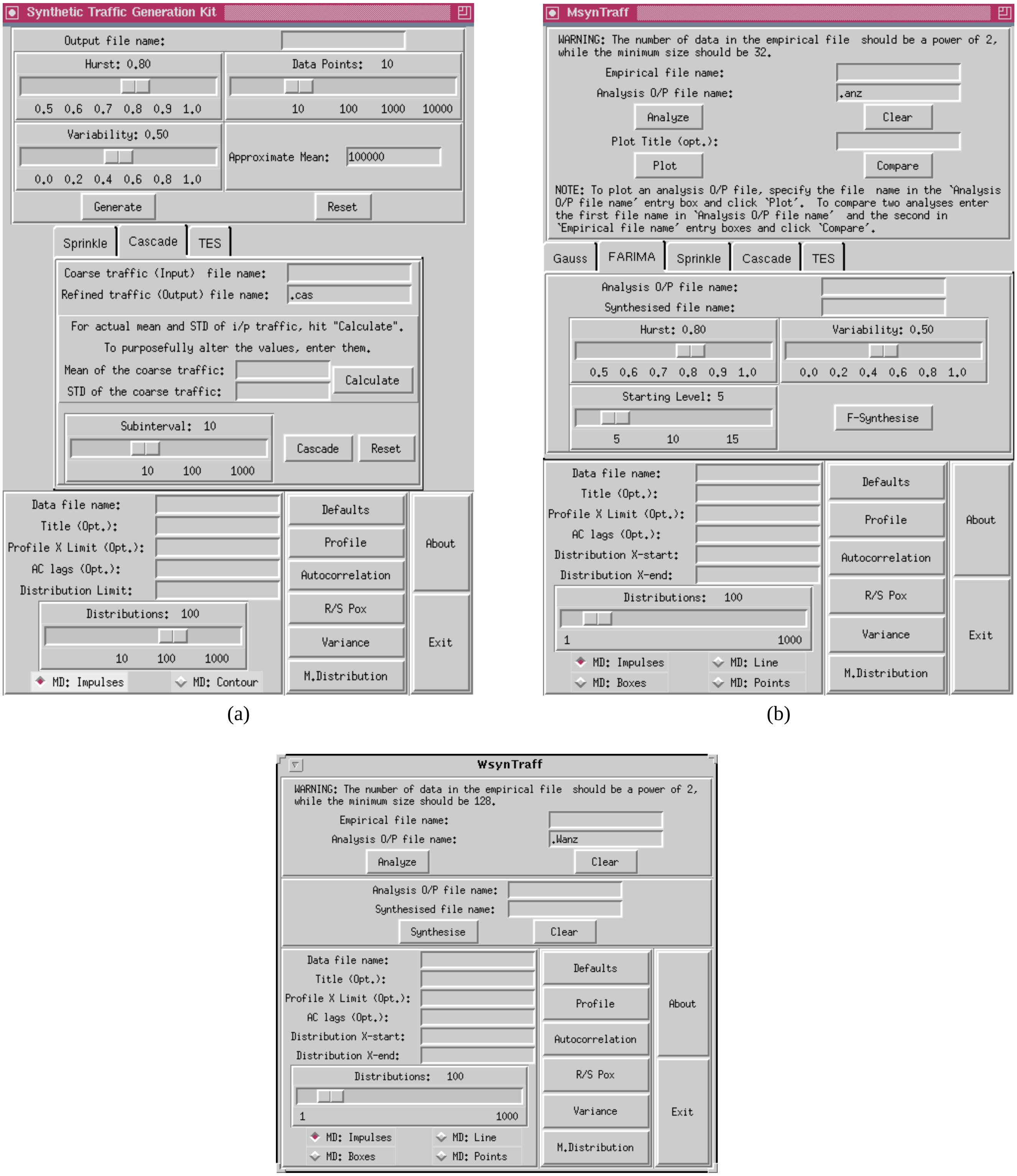
Our traffic modeling toolkits have been developed in and for a Unix-based environment running X windows. The traffic analysis, modeling, and generation tools are written in C/C++. The user interface is written in Tcl/Tk.

Installing and running our toolkits requires the following software: cc, gcc, g++, tcl (version 8.0 or newer), tk (version 8.0 or newer), mktclapp, gnuplot (a graph plotting tool), and gs (a PostScript previewer, also known as ghostscript, gsview, or gsview32).

For hardware requirements, at least 32 MB RAM is desirable, so that reasonably large data sets can be analyzed (e.g., 2^{20} data points). Adequate disk storage capacity (e.g., 10-100 Megabytes) is required for storing empirical and/or synthetic traces, which tend to be large.

3. The synTraff toolkit

Figure 1(a) shows the main interface of the *synTraff* kit. This interface has three main blocks from top to bottom, corresponding to traffic generation, refinement, and analysis, respectively.



Hurst: 0.80

Data Points: 10

Variability: 0.50

Approximate Mean: 100000

Refined traffic (Output) file name: .cas

For actual mean and STD of i/p traffic, hit "Calculate".

To purposefully alter the values, enter them.

Subinterval: 10

Distributions: 100

◆ MD: Impulses ◇ MD: Contour

(a)

WARNING: The number of data in the empirical file should be a power of 2, while the minimum size should be 32.

Analysis O/P file name: .anz

NOTE: To plot an analysis O/P file, specify the file name in the `Analysis O/P file name' entry box and click `Plot'. To compare two analyses enter the first file name in `Analysis O/P file name' and the second in `Empirical file name' entry boxes and click `Compare'.

Hurst: 0.80

Variability: 0.50

Starting Level: 5

Distributions: 100

◆ MD: Impulses ◇ MD: Line
◇ MD: Boxes ◇ MD: Points

(b)

WARNING: The number of data in the empirical file should be a power of 2, while the minimum size should be 128.

Analysis O/P file name: .Wanz

Distributions: 100

◆ MD: Impulses ◇ MD: Line
◇ MD: Boxes ◇ MD: Points

(c)

Figure 1. Illustration of the graphical user interfaces for: (a) *synTraff*; **(b)** *MsynTraff*; **and (c)** *WsynTraff*.

The top block is for traffic generation. It has an entry box to enter the name of the file for saving the traffic generated. It has three scale widgets and an entry box for specifying the three traffic parameters–Hurst, Variability, and Mean–and the number of data points to be generated. The block has two buttons, one for generating traffic according to the specifications provided (the output contains a series of positive real values, representing traffic volumes) and the other to reset the default parameter values. Scale widgets constrain users to provide valid specifications. The LRD time series is generated using an F-ARIMA process (Fractional Auto-Regressive Integrated Moving Average) [13]. This process has $O(N^2)$ running time for N data points.

The middle block is for refining coarse-grain (higher) time scale traffic down to finer-grain (lower) time scale traffic. For example, a user can initially generate the time series data at one time scale (e.g., 1.0 second traffic samples), and then refine it to a lower time scale (e.g., 0.01 second traffic samples) by randomly distributing the "mass" of each coarse-grain observation across several subintervals (e.g., 100). The block has three overlapping frames corresponding to three different refining techniques, namely Sprinkle, Cascade, and TES; clicking on the tabs provided will bring the corresponding frame into view. While Cascade and TES are network traffic modeling techniques in themselves, here they can be used to refine traffic generated by F-ARIMA or to generate their own traffic. To generate pure Cascade or TES traffic, the mean of the top block is used to specify the characteristics of the desired synthetic traffic, prior to proceeding with refining.

Regardless of the initial generation technique chosen, the (optional) refinement step requires the name of the source file that has coarse-grain (synthetic or empirical) traffic, a file name to store the refined traffic, and the refining subinterval. The latter can be input using the "Subinterval" scale widget. The other parameters required for refinement vary depending on the refinement technique. For the Sprinkle technique, the user has to specify a threshold value using the scale widget provided. By varying the threshold value, the user can control the smoothness or variability of the refined traffic. For the Cascade technique, the required parameters are the mean and the standard deviation (STD) of the input traffic. By altering these values, the user can modify the histogram and variability of the traffic. However, if the user does not want to change these values he/she can simply press the "Calculate" button to have synTraff calculate and display these values automatically. Finally, for the TES technique the only additional refining parameter required is the Hurst value, which affects the magnitude of the short-range correlations induced by the TES model. The Hurst value is specified by the scale widget provided. After entering the refining parameters for the chosen technique, the user can click on the "Refine" button or can reset the parameters by clicking the "Reset" button.

The bottom block is for viewing the statistical properties of a trace (e.g., profile, marginal distribution, autocorrelation, variance-time, and R/S pox plots). Before generating the mentioned plots by clicking corresponding buttons in the block, the user has to specify the name of the traffic file to be analyzed. The user can change the viewing properties of the plots (e.g., scales on axes, line types) by entering plot properties in the entry boxes provided. Before changing a plot property the user can view the default values by clicking on the "Defaults" button.

An example of the monofractal traffic generated (and analyzed) by the *synTraff* tool is shown in the leftmost column of Figure 2. The empirical traffic trace (a Bellcore Ethernet LAN traffic trace [8]) on which it is based is shown in the middle column of Figure 2. The rightmost column shows a multifractal traffic trace generated (and analyzed) by the *MsynTraff* tool (see Section 5). Both synthetic traces show that they capture the behavioural characteristics of the empirical trace reasonably well.

4. The WsynTraff Toolkit

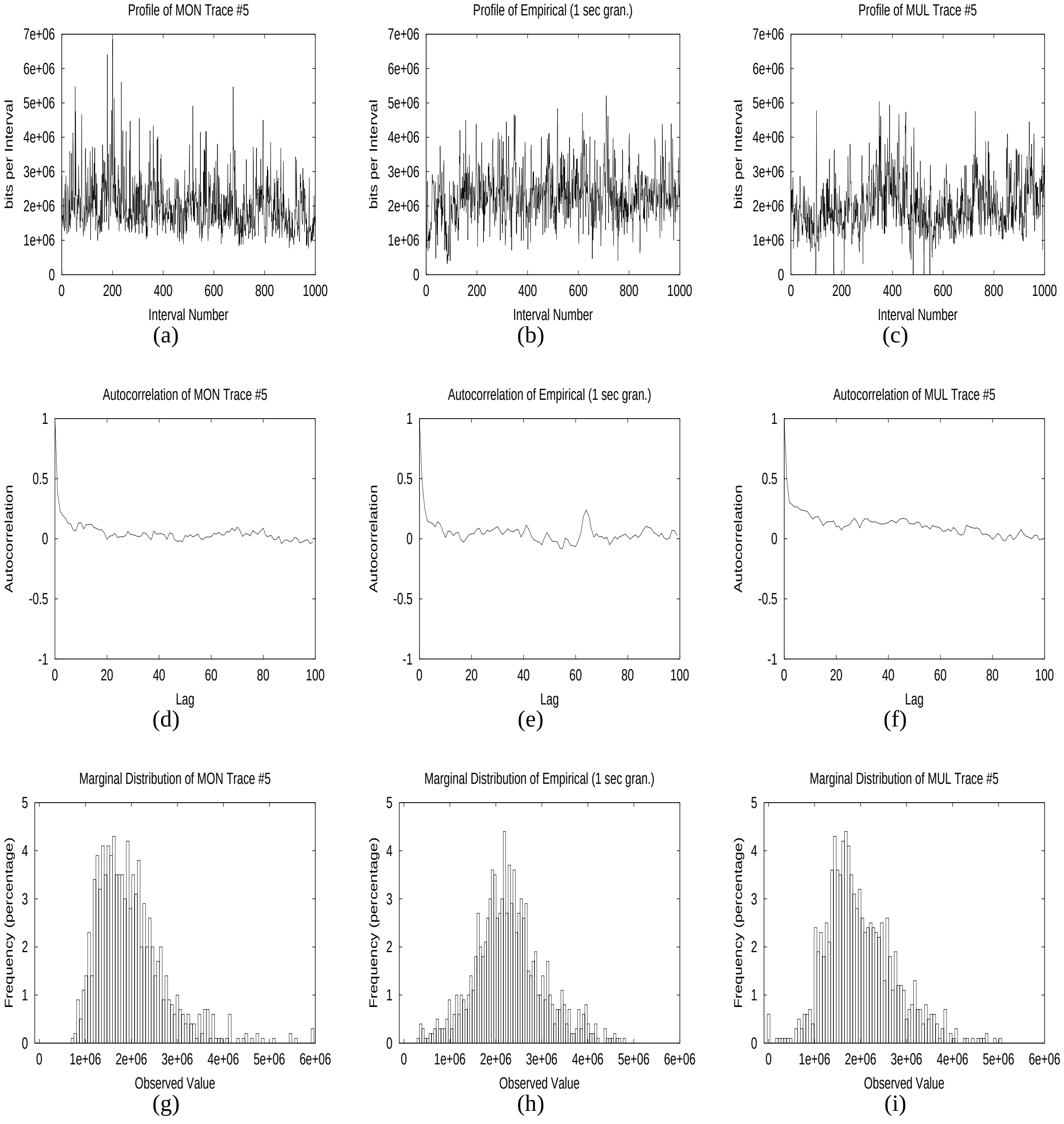
Figure 1(c) shows the main interface of the *WsynTraff* kit. Similar to *synTraff*, *WsynTraff* has three main blocks representing three major functions of the tool. However, the first two blocks (top and middle) totally differ from their counterparts in *synTraff*.

The top block is for constructing a WIG model of an empirical traffic trace. The block has two entry boxes for entering the file name of the empirical trace to be analyzed and the other for the output file name. The block has two push buttons: one to analyze the specified traffic traces, and the other to clear the entries.

The middle block of the kit is for synthesizing multifractal wavelet traffic based on the analysis files generated by the kit. The block has two entry boxes: the first is for the name of a WIG analysis file generated by the kit, and the second is for the name of the file for the generated traffic. The block has two push buttons. The "Synthesize" button is to instruct the kit to synthesize traffic, and the "Clear" button is for clearing the entries in the block.

The bottom block of the kit is for analyzing and plotting the statistical properties of synthetic or empirical traffic traces. This block is similar to that in the *synTraff* tool.

One advantage of the WIG model is that it is fast: $O(N)$ running time to generate N data points. One disadvantage is that it occasionally generates negative values for traffic samples [1, 12]. For this reason, we have not used it as a traffic generation technique for our simulation studies. The interested reader is referred to [1] for further details about the WIG model and the *WsynTraff* toolkit.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 2. Comparison of the synthetic traces generated by *synTraff* (left column) and *MsynTraff* (right column) with an empirical trace (middle column) at 1.0-second time-scale granularity: (a) & (b) & (c) Traffic profiles; (d) & (e) & (f) Autocorrelation functions; and (g) & (h) & (i) Marginal distributions

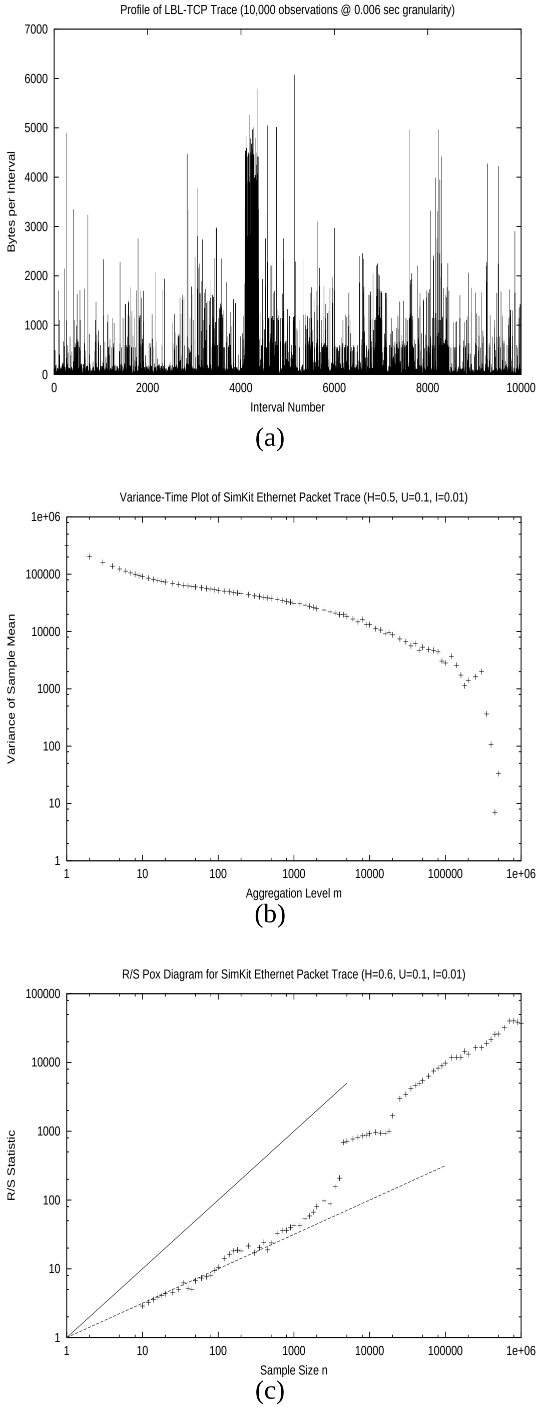


Figure 3. Illustration of traffic analysis for LBL-TCP trace (1,048,576 observations at 0.006 sec granularity): (a) Traffic profile (10,000 observations); (b) Variance-time plot; and (c) R/S pox plot

5. The MsynTraff Toolkit

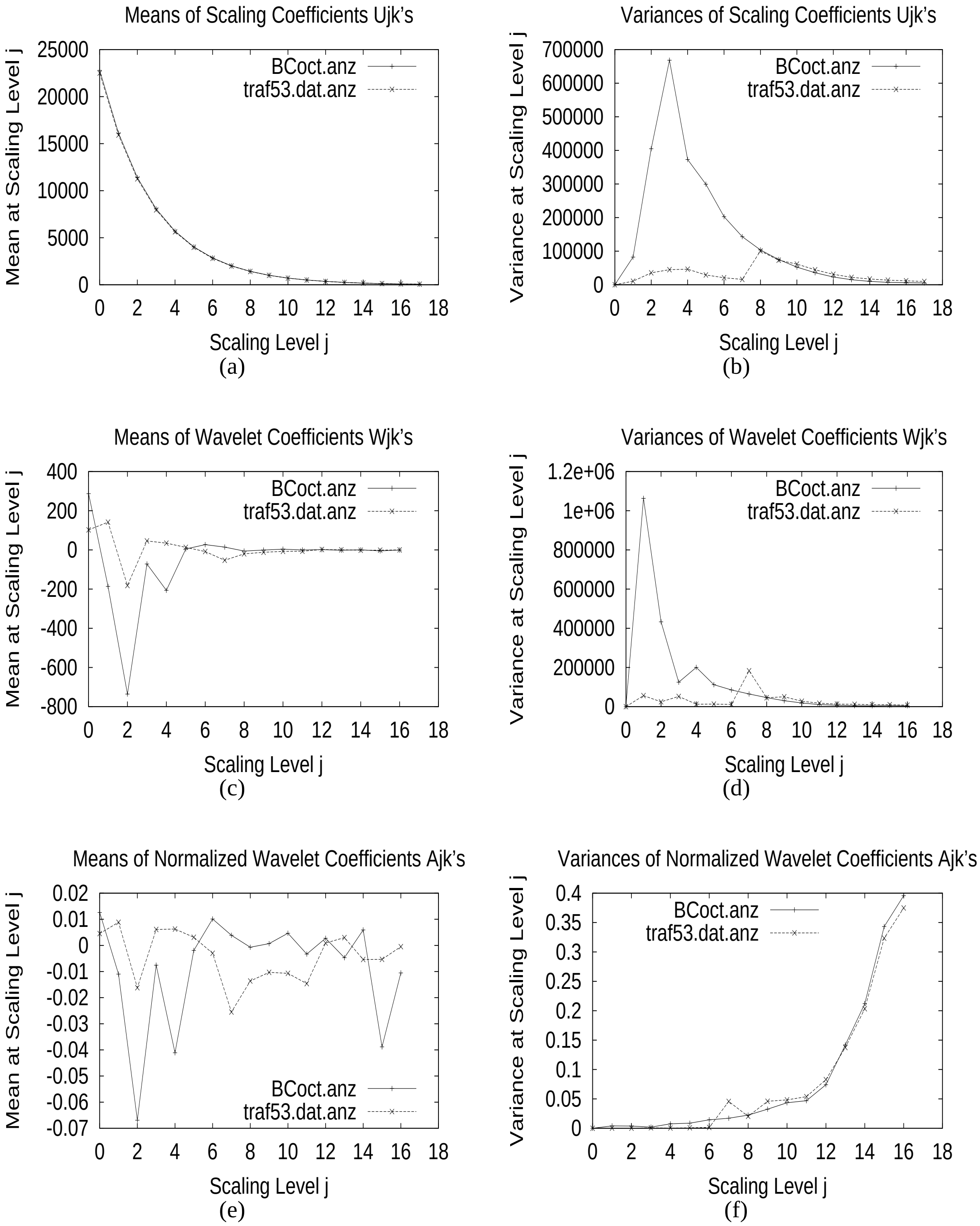
Figure 1(b) shows the main interface of the *MsynTraff* kit. Like the other tools, *MsynTraff* has three main blocks.

The top block is for constructing and analyzing the multifractal wavelet model of an empirical traffic trace. Entry boxes are provided in the module to enter the file name of the empirical traffic to be analyzed, and the output file name for the results of the analysis. When constructing an MWM model, a user can choose to view its wavelet characteristics or can choose to compare the wavelet characteristics of two MWM models. For this purpose, the kit generates six graphs, representing the means of the scaling coefficients, the variances of the scaling coefficients, the means of the wavelet coefficients, the variances of the wavelet coefficients, the means of the normalized coefficients, and the variances of the normalized coefficients. In order to generate these graphs for one MWM model, the user enters the file name in the entry box “Analysis O/P file name” and clicks the “Plot” button. To generate comparative graphs for two MWM models, the user enters two file names, one in each of the two entry boxes “Analysis O/P file name” and “Empirical file name”. The user can also choose to enter an optional caption for the graphs. Figure 4 shows an example of the comparative plots generated by the kit, which provide a graphical “goodness of fit” test for a traffic model.

The center block of the kit is for generating multifractal wavelet traffic traces, based on the analysis files generated by the kit. The MWM technique suggests Gaussian random generation of coarse-grain data points (e.g., “seeding” level 5 of the (internal) complete binary tree representation of the wavelet model with 32 independently-chosen random values, prior to traffic generation) [12]. However, the kit allows the user to specify (with the “Starting Level” scale widget) the level of the binary tree at which to seed random values, and also provides the following (new) options to generate coarse-grain data points: Gaussian, F-ARIMA, Sprinkle, Cascade, and TES [1]. As before, clicking the tab in each block brings the corresponding frame into user view. Regardless of the technique chosen, the user has to indicate the name of the MWM analysis file generated by the kit, and the file name for the generated traffic. After specifying the parameters, the user clicks on the “Synthesize” button to generate traffic.

The bottom block is exactly the same as in the *WsynTraff* tool. It is for analyzing and plotting the statistical properties of synthetic or empirical traffic traces.

By design, the MWM model generates only non-negative sample values [12], making it suitable for network traffic modeling. As with the WIG model, the running time for traffic generation (or analysis) is $O(N)$ for N data points, making the approach fast enough to work with large data sets (e.g., 2^{20} data points).



(a) (b)

(c) (d)

(e) (f)

Figure 4. Comparison of MWM models for empirical trace BC-Oct89 and synthetic trace: (a) Mean of scaling coefficients Ujk; (b) Variance of scaling coefficients Ujk; (c) Mean of wavelet coefficients Wjk; (d) Variance of wavelet coefficients Wjk; (e) Mean of normalized wavelet coefficients Ajk; and (f) Variance of normalized wavelet coefficients Ajk

6. Summary

There is ample evidence in the research literature that network traffic is long-range dependent, and furthermore that network traffic is multifractal in nature, rather than monofractal. This correlation structure in traffic is an important property, and is one that needs to be modeled in simulation-based studies of network performance.

This paper has presented a set of traffic modeling toolkits capable of generating and analyzing monofractal and multifractal traffic. The traffic modeling toolkits are based on traffic modeling techniques from the published literature [4, 6, 9, 12, 13], and have an easy-to-use graphical interface written in Tcl/Tk. We have found the toolkits useful for generating independent monofractal and multifractal streams for network performance studies [1, 2].

Our traffic modeling toolkits, and accompanying documentation, are available to the MASCOTS community. At this time, only UNIX-based versions of our tools are available [14]. Porting our toolkits to the PC Windows environment remains as future work.

Acknowledgements

Financial support for this research was provided by an NSERC Collaborative Research and Development (CRD) grant to the ATM-TN TeleSim project (CRD183839), and by NSERC Research Grant OGP0121969. The authors thank the TeleSim project team for building and maintaining the ATM-TN simulator (on which much of our simulation work depends), and for their constructive feedback on our traffic modeling work. We also value the ongoing interactions with our industrial sponsors, including Newbridge Networks (now Alcatel CID), Nortel Networks, Siemens, and Telus.

References

[1] R. Balakrishnan, *Wavelet-Based Network Traffic Modeling*, M.Sc. Thesis, Department of Computer Science, University of Saskatchewan, March 2000.

[2] R. Balakrishnan and C. Williamson, “A Performance Comparison of Monofractal and Multifractal Traffic Streams”, *Proceedings of the Eighth International Symposium on the Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS’2000),* San Francisco, CA, August 2000.

[3] A. Feldmann, A. Gilbert, P. Huang, and W. Willinger, “Dynamics of IP traffic: A Study of the Role of Variability and the Impact of Control”, *Proceedings of the 1999 ACM SIGCOMM Conference*, Cambridge, MA, pp. 301-313, August 1999.

[4] A. Feldmann, A. Gilbert, and W. Willinger, “Data Networks as Cascades: Investigating the Multifractal Nature of Internet WAN Traffic”, *Proceedings of the 1998 ACM SIGCOMM Conference*, Vancouver, Canada, pp. 42-55, August 1998.

[5] M. Garrett and W. Willinger, “Analysis, Modeling, and Generation of Self-Similar VBR Video Traffic”, *Proceedings of the 1994 ACM SIGCOMM Conference*, London, UK, pp. 269-280, August 1994.

[6] D. Jagerman and B. Melamed, “The Transition and Autocorrelation Structure of TES Processes Part I: General Theory”, *Stochastic Models,* Vol. 8, No. 2, pp. 193-219, 1992.

[7] D. Jagerman and B. Melamed, “The Transition and Autocorrelation Structure of TES Processes Part II: Special Cases”, *Stochastic Models*, Vol. 8, No. 3, pp. 499-527, 1992.

[8] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, “On the Self-Similar Nature of Ethernet Traffic (Extended Version)”, *IEEE/ACM Transactions on Networking*, Vol. 2, No. 1, pp. 1-15, February 1994.

[9] S. Ma and C. Ji, “Modeling Video Traffic in the Wavelet Domain”, *Proceedings of 17th Annual IEEE Conference on Computer Communications, INFOCOMM,* San Francisco, CA, pp. 201-208, March 1998.

[10] I. Norros, “On the Use of Fractional Brownian Motion in the Theory of Connectionless Networks”, *IEEE Journal on Selected Areas in Communications,* Vol. 13, No. 6, pp. 953-962, August 1995.

[11] V. Paxson and S. Floyd, “Wide Area Traffic: The Failure of Poisson Modeling”, *Proceedings of the 1994 ACM SIGCOMM Conference*, London, UK, pp. 257-268, August 1994.

[12] V. Ribeiro, R. Riedi, M. Crouse and R. Baraniuk, “Simulation of Non-Gaussian Long-Range-Dependent Traffic Using Wavelets”, *Proceedings of the 1999 ACM SIGMETRICS Conference*, Atlanta, GA, pp. 1-12, May 1999.

[13] C. Williamson, “Synthetic Traffic Generation Techniques for ATM Network Simulations”, *Simulation Journal,* Vol. 72, No. 5, pp. 305-312, May 1999.

[14] C. Williamson, *The synTraff Suite of Traffic Modeling Toolkits*, software and documentation available from URL http://www.cs.usask.ca/faculty/carey/software/