Neural Network Estimation of Packet Arrival Rate in Self-Similar Queuing Systems

Homayoun Yousefi'zadeh
Department of Electrical Engineering and Computer Science
University of California, Irvine
hyousefi@uci.edu

***Abstract*—Estimating average latency of queuing systems is one of the most challenging tasks in the analysis and design of traffic control algorithms. In this paper a new approach for predicting packet arrival rate in multiple source queuing systems is introduced. The approach relies on the modeling power of neural networks in predicting self-similar traffic patterns to determine the packet arrival rate of low loss, moderately loaded queuing systems accommodating such traffic patterns. Characterizing the average latency by the average queuing delay, Little's law can be appropriately utilized to describe the average packet latency of such systems.**

***Index Terms*— Bursty Traffic, Self-Similarity, Intelligent Traffic Modeling, Neural Network, Packet Arrival, Average Queuing Latency.**

I. Introduction

Analysis of traffic data from networks and services such as Ethernet LANs [11], Variable Bit Rate (VBR) video [2], ISDN traffic [9], and Common Channel Signaling Network (CCNS) [3] have all convincingly demonstrated the presence of features such as long range dependence, slowly decaying variances, and heavy-tailed distributions. These features are best described within the context of second-order self-similarity and fractal theory. Self-similar phenomena show structural similarities across a wide range of time scales in which traffic spikes ride on the longer term ripples, that in turn ride on longer term swells, so on and so forth.

Chaos is a phenomenon observed in nonlinear dynamical systems. It may be used to explain why a low order system is capable of exhibiting very complicated behavior. Since the trajectories of chaotic systems are mostly fractals, they may be used as suitable generators of fractals and traffic patterns with fractal nature. From the modeling point of view, the challenge is to capture the complexity of a bursty traffic pattern with the small number of parameters of a chaotic map. Chaotic maps have been used in the literature for modeling and forecasting of teletraffic. Erramilli et al. [4] used a number of simple nonlinear maps in order to capture some of the real traffic patterns characteristics. Giovanardi et al. [7] used self-similar chaos-based traffic patterns in an analytical study of queuing systems. Alkhatib et al. [1] used chaos theory in modeling and forecasting VBR video patterns.

Neural networks are a class of nonlinear systems capable of learning and performing tasks accomplished by other systems. Some of the applications of neural networks are speech and signal processing, pattern recognition, and system modeling. Systems with neural network building blocks are robust in the sense that occurrence of small errors does not interfere with proper operation of the system. This characteristic of neural networks makes them quite suitable for traffic modeling.

Predicting packet arrival and estimating packet latency is a major design issue in computer communication networks. There are a number of factors that introduce delay in network services. Different delay types may be classified under processing, propagation, multiplexing, and queuing categories. The main objective of packet scheduling methods is then to come up with solutions for predicting and reducing delay while efficiently utilizing network resources.

In [16], we made use of the modeling power of neural networks introduced in [15] to provide a fair dynamic buffer management scheme improving the loss performance of a class of queuing systems with self-similar characteristics. In this study, we utilize the modeling power of neural networks in predicting self-similar traffic patterns to determine the arrival rate and the packet latency of queuing systems accommodating such patterns. In doing so, we assume the service rates are given and there is no significant loss impact. Our packet arrival estimation technique might be employed as a part of a packet scheduling algorithm.

An outline of the paper follows. Section II briefly reviews the characteristics of aggregated self-similar traffic patterns. Section III provides an overview of the neural network modeling of such traffic patterns. Section IV describes a multiple source system used for the application task and discusses the packet arrival estimation application. Section V evaluates the performance of an average latency estimation technique based on our proposed method. In our evaluation, we compare the results of our estimation technique with measured average latency results in the presence of typical buffer management and server scheduling schemes. The paper concludes in Section VI.

II. Second-Order Self-Similarity

This section includes a brief description of self-similarity. Suppose $X = (X_t : t = 0, 1, 2, ...)$ is a covariance stationary stochastic process with mean μ, variance σ^2, and autocorrelation function $R(n), \quad n \geq 0$. Particularly, assume the autocorrelation function of X has the form

$$R(n) \sim k_1 n^{-\beta}, \quad \text{as} \qquad n \rightarrow \infty \tag{1}$$

where $0 < \beta < 1$ and constants $k_1, k_2, ...$ are finite positive integers. For each $m = 1, 2, 3, ...$ let $X^{(m)} = (X_n^{(m)} : n = 1, 2, 3, ...)$ be the covariance stationary time series with corresponding autocorrelation function $R^{(m)}$ obtained from averaging the original series X over the non-overlapping time periods of size m, i.e., for each $m = 1, 2, 3, ...$, $X^{(m)}$ is given by

$$X_n^{(m)} = \frac{1}{m}(X_{nm-m+1} + ... + X_{nm}), \quad n \geq 1 \qquad (2)$$

The process X is called exactly second-order self-similar with the self-similarity parameter $H = 1 - \beta/2$ if the corresponding $X^{(m)}$ has the same correlation function as X, i.e., $R^{(m)}(n) = R(n)$ for all $m = 1, 2, 3, ...$ and $n = 1, 2, 3, ...$. X is called asymptotically second-order self-similar with self-similarity parameter $H = 1 - \beta/2$ if $R^{(m)}(n)$ asymptotically approaches to $R(n)$ given by (1), for large m and n. Hence, if the correlation functions of the aggregated processes $X^{(m)}$ are the same as the correlation functions of X or approach asymptotically to the correlation functions of X, then X is called exactly or asymptotically second-order self-similar.

Fractal Gaussian Noise (FGN) is a good example of an exactly self-similar process with self-similarity parameter $H, \quad 1/2 < H < 1$. Fractional Arima processes with the parameters (p, d, q) such that $0 < d < 1/2$ are examples of asymptotically second-order self-similar processes with self-similarity parameter $d + 1/2$.

Mathematically, self-similarity manifests itself in a number of ways.

- The variance of sample mean decreases more slowly than the reciprocal of the sample size. This is called slowly decaying variance property which means $var(X^{(m)}) \sim k_2 m^{(-\beta)}$ as $m \to \infty$ with $0 < \beta < 1$.
- The autocorrelations decay hyperbolically rather than exponentially fast implying a non-summable autocorrelation function $\sum_n R(n) = \infty$. This is called long range dependence property.
- The spectral density $f(.)$ obeys a power-law near the origin. This is the concept of $1/f$ noise with the meaning $f(\lambda) = k_3 \lambda^{-\gamma}$ as $\lambda \to \infty$ with $0 < \gamma < 1$ and $\gamma = 1 - \beta$.

It appears that the most important feature of self-similar processes is that their aggregated process $X^{(m)}$ possess a non-degenerate correlation function as $m \to \infty$. This is completely different from typical packet traffic models previously considered in the literature, all of which have the property that their aggregated processes $X^{(m)}$ tend to second order pure noise, i.e., $R^{(m)} \to 0$ as $m \to \infty$.

The concept of self-similar processes provides an elegant explanation for the original Hurst effect phenomenon. In order to describe the Hurst effect, we should first describe the rescaled adjusted range. For a given set of observations $(X_n : n = 1, 2, ..., N)$ with sample mean $\overline{X}(N)$ and sample variance $S^2(N)$, the rescaled adjusted range denoted by the R/S statistic is given by

$$\frac{R(N)}{S(N)} = \frac{1}{S(N)}[max(W_i) - min(W_i)] \qquad (3)$$

where $i = 0, ..., N$, $W_0 = 0$, and

$$W_n = (X_1 + ... + X_n) - n\overline{X}(N), \quad n \geq 1 \qquad (4)$$

While many time series appear to be well represented by the relation $E[R(N)/S(N)] \sim k_4 N^H$, as $N \to \infty$, with Hurst parameter H typically about 0.73, observations X_n from a short-range dependent models are known to satisfy $E[R(N)/S(N)] \sim k_5 N^{0.5}$, as $N \to \infty$. This is usually referred to as the Hurst effect.

III. Neural Network Modeling of Self-Similar Traffic

In [15], we describe how a fixed structure feed forward perceptron neural network with back propagation learning algorithm can be used to model aggregated self-similar traffic patterns as an alternative to stochastic and chaotic systems approaches proposed in [10] and [4]. We note that although the emphasis of our work is on self-similar traffic modeling, our proposed neural network modeling approach can nevertheless be used for any traffic pattern independent of self-similarity. In what follows we briefly review the neural network modeling technique of [15] in which an elegant approach capable of coping with the fractal properties of the aggregated traffic is introduced. The approach provides an attractive solution for traffic modeling and has the advantage of simplicity compared to the previously proposed approaches namely stochastic and deterministic chaotic map modeling. The promise of neural network modeling approach is to replace the analytical difficulties encountered in the other modeling approaches with a straight forward computational algorithm. As oppose to the other modeling approaches, neural network modeling does not introduce a parameter (or a set of parameters) describing the fractal nature of traffic neither does it investigate identification of appropriate maps. It, hence, need not cope with the complexity of estimating multifractal Hurst parameters [10], [6] and/or fractal dimensions [4]. The approach simply takes advantage of using a fixed structure nonlinear system with a well defined analytical model that is able to predict a traffic pattern after learning the pattern dynamics through the use of information available in a number of traffic samples. Interestingly and as proposed by Gomes et al. [8], neural networks can also be utilized as appropriate estimators of the Hurst parameter.

The fixed structure, fully connected, feed forward perceptron neural network utilized for the task of modeling consists of an input layer with eight neurons, three hidden layers with twenty neurons in each layer, and an output layer with one neuron. Fig. 1 illustrates the structure of the neural network. The sigmoid transfer function defined below

$$f(z) = (1 + e^{-z})^{-1} \qquad (5)$$

is utilized to generate the output of each neuron from its compound input. The output of each neuron is connected to the input of all of the neurons in the layer above after being multiplied by a weighting function. The specific neural network used for the task of modeling relies on the so-called back propagation learning algorithm described in [5], [15], and the references therein. In a nutshell, the back propagation learning algorithm changes the weighting functions of the underlying neural network in the opposite direction of the gradient vector and its momentums in order to minimize the absolute error function defined proportional to the square of the difference between the neural network output and the real output.

In a typical iteration of the learning phase, the neural network is provided with samples $x[k-8]$ through $x[k-1]$ of the real traffic pattern and the difference between sample $x[k]$ of the real traffic pattern and the neural network output is used to adjust the weighting functions of the network accordingly. In the next iteration, sample $x[k-8]$ of the real traffic pattern is discarded, samples $x[k-7]$ through $x[k]$ of the real traffic pattern are used as the new input sample set, and sample $x[k+1]$ is used as the new real traffic sample. The neural network continues processing more information in consecutive iterations of the learning phase until the absolute error is less than a specified error bound, δ. The learning phase of the perceptron neural network is directly followed by the recalling phase when the network output is able to follow the real traffic within the acceptable error bound, δ. In each iteration of the recalling phase, the neural network independently generates the samples by discarding the oldest input sample, shifting the input samples by one, and using its output as the most recent input sample. The same sequence of following a learning phase by a recalling phase is repeated when and if the neural network output difference exceeds the acceptable error bound, δ. The number of samples required for the training of the neural network depends on the complexity of the traffic pattern dynamics. The time complexity and the space complexity of the back propagation algorithm are respectively $\mathcal{O}(IN)$ and $\mathcal{O}(N)$ where N is the number of weighting functions in the network and I is the number of iterations. Although the complexity is typically better than the complexity of implementing statistical approaches such as fractional ARIMA processes or the complexity of calculating fractal dimensions such as correlation dimension, wide variations of I prevent us from making a strong claim about complexity advantage of the algorithm compared to other algorithms. Nonetheless combining the straight forward way of implementation with the analysis of complexity, we claim that the neural network modeling approach provides an elegant approach for the task of traffic modeling.

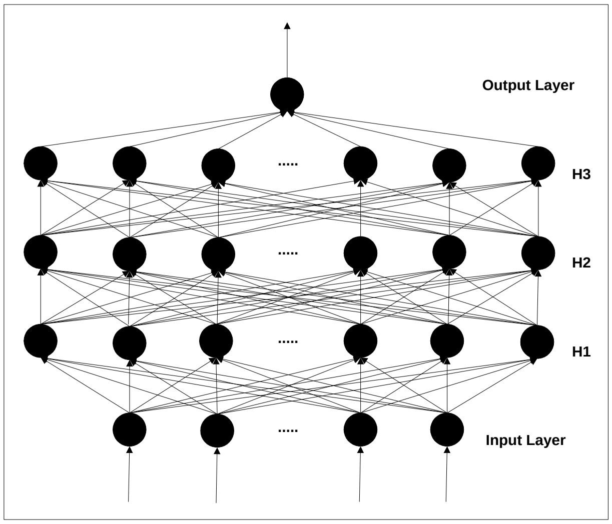

Fig. 1. Fixed structure neural network used for the task of modeling.

In the following section, we apply the proposed neural network modeling technique to predict the packet generation patterns of a number of ON-OFF traffic sources and utilize the results in predicting arrival rates and estimating average latencies in queuing systems accommodating such patterns.

IV. Packet Arrival and Queuing Latency in Self-Similar Queuing Systems

Our application test bed relies on a multiple source queuing system. A multiple source queuing system consists of a number of sources sharing a total available buffer space. Traffic pattern of each source includes the packets generated by a number of ON-OFF chaotic maps. An ON-OFF source model is generating traffic at a peak rate when it is active and becomes active as soon as the state variable of the describing chaotic map goes beyond a threshold value, d. The source becomes passive as soon as the state variable goes below the threshold value. We utilize double intermittency map in our packet generation process as it generates a self-similar traffic pattern according to what is described in [4]. The describing equation of double intermittency map is

$$x_{n+1} = \begin{cases} \epsilon_1 + x_n + c_1 x_n^m & : \quad 0 \leq x_n \leq d \\ -\epsilon_2 + x_n + c_2(1-x_n)^m & : \quad d \leq x_n \leq 1 \end{cases} \tag{6}$$

where x_n represents the discrete variable and the rest of the symbols represent various parameters with the property $c_1 = \frac{1-\epsilon_1-d}{d^m}$. Fig. 2 illustrates a sample drawing of double intermittency map. As observed in the figure, the iterative map requires multiple samples to move from one segment to another.

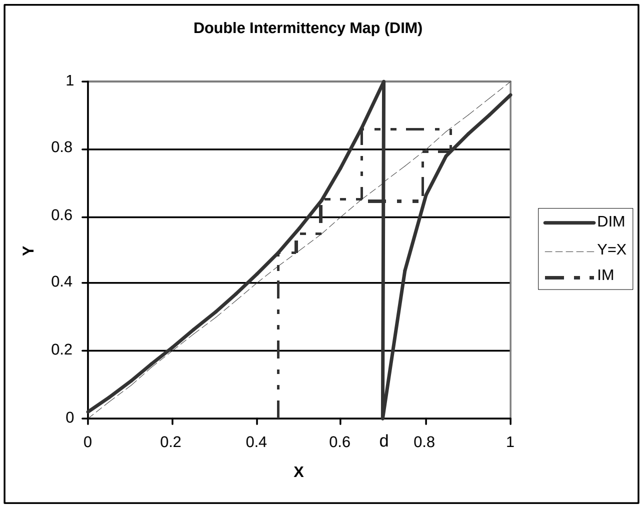

Fig. 2. A sample drawing of the double intermittency map. The legend IM indicates the path traversed by the map through consecutive iterations.

We propose using different initial conditions and a fixed threshold value to obtain different traffic patterns for different sources. As an alternative, one may use different threshold values with fixed or variable initial conditions to achieve varying traffic patterns for different sources. We select initial conditions in the range of $x_0 \in [0.1, 0.3]$ along with a fixed threshold value of $d = 0.7$ and parameters $\epsilon_1 = 0.01$, $\epsilon_2 = 0.05$, $m = 5$, $c_1 = 1.73$, $c_2 = 267.49$.

We now apply our neural network modeling scheme to predict the total number of generated packets and utilize the results in predicting the arrival rate and estimating the queuing delay for the packets generated by a number of traffic sources. Recall that for a given service rate and a known buffer occupancy, the queuing delay of a packet can be measured as the average number of time units it spends in the queue before leaving the buffer. Consider a multiple source queuing system such as the one shown in Fig. 3 with three sources sharing the space available in a central buffer. Assume that the aggregated traffic pattern of each individual source consists of the traffic patterns of 120 sources generating ON-OFF packet traffic according to double intermittency map model. The buffer size is assumed to be fixed, large enough to prevent any loss. In addition, suppose that the system is utilizing complete sharing buffer management, Statistical Time Division Multiplexing (STDM) scheduling, and First Come First Serve (FCFS) service discipline schemes as described in [16]. Under the above circumstances, the queuing system is best described by $G/G/1/\infty$ model. The average latency of the queuing system is, hence, described by Little's law as $N = \lambda T$ where N, λ, and T respectively represent the number of packets in the queue, the queue arrival rate, and the average service time. We make note of the fact analyzing such a system utilizing Lindley's Integral Equation or another comparable analytical technique is a rather complicated task. Instead, we propose utilizing the neural network modeling scheme of [15] to predict the packet arrival rate of the central buffer.

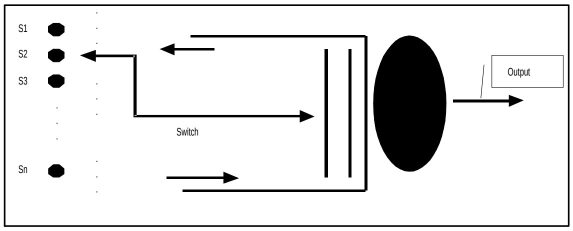

Fig. 3. The structure of a multiple source queuing system.

V. Simulation Results

Fig. 4 displays our simulation results for the system described above. It shows the Measured Average Latency (MAL) and the Estimated Average Latency (EAL) versus service time diagram for the triple source queuing system over the intervals in which the arrival rate predictions are of acceptable accuracy. The average latency has been calculated over the time periods in which the neural network has been able to follow the arrival pattern of the central buffer. For the relative error defined as $\frac{|MAL-EAL|}{MAL}$, Fig. 4 shows that the estimation results are within the 3% relative error range pending the following conditions are held. First, the averaging period is long enough in order for the neural network to be able to follow the traffic pattern for a number of times within the specified error bounds and second, the buffer service rate does not exceed an existing threshold value. Although not shown in the simulation results, we have observed that the average packet latency drops sharply by choosing service rates beyond the threshold value. In the latter case, the neural network latency estimation findings are not acceptable as the result of having high service rates and low average latencies. The service rate threshold generally depends on the dynamics of the system and for the triple source system of our experiment is the normalized value 13.

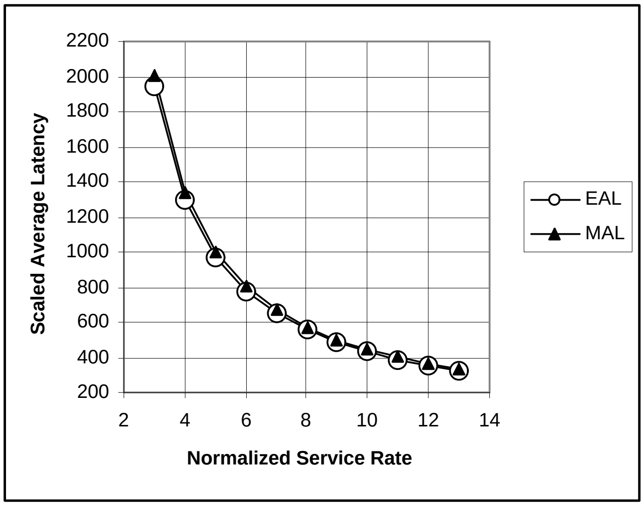

Fig. 4. Estimated average latency (EAL), and measured average latency (MAL) versus normalized service rate for the triple source queuing system.

We finish this section by mentioning that a typical sequence of learning and recalling phases consists of few hundred thousand samples and hundreds of samples respectively. In addition, all of the convergence results are strongly affected by the choice of initial conditions of the weighting functions of the neural network. As a practical finding, setting the initial values of the weighting functions of the neural network at 0.01 typically yields good results. Additionally, one may set the weighting functions randomly in the order of 0.01 if facing biasing and saturation. Our justification for both of the above phenomena is the fact that the proposed neural network is trying to learn complicated dynamics of chaotic maps exhibiting extreme sensitivity to variations of initial conditions.

VI. Conclusion

In this paper, we introduced a novel approach for predicting packet arrival and estimating queuing latency in multiple source queuing systems as an application of neural network modeling of self-similar packet traffic. We relied on the prediction power of neural networks to estimate arrival rates and packet latencies in multiple source queuing systems accommodating self-similar traffic patterns. We evaluated the performance of our estimation technique by comparing estimated average latency with measured average latency and concluded that the scheme is able to provide an acceptable estimate with a less than 3% relative error below a specified service rate threshold for moderately loaded systems with no significant loss.

References

[1] A. Alkhatib, M. Krunz, "Application of Chaos Theory to the Modeling of Compressed Video," Proc. of the IEEE ICC 2000 Conference, Vol. 2, New Orleans, June 2000.

[2] J. Beran, R. Sherman, M. S. Taqqu, W. Willinger, "Variable Bit Rate Video Traffic and Long Range Dependence," IEEE/ACM Trans. on Networking, Vol. 2, NO. 3, Apr. 1994.

[3] D. E. Duffy, W. Willinger, "Statistical Analysis of CCSN/SS7 Traffic Data from Working CCS Subnetworks," IEEE JSAC, 1994.

[4] A. Erramilli, R. P. Singh, P. Pruthi, "Chaotic Maps as Models of Packet Traffic," ITC Vol. 14, PP. 329-338, 1994.

[5] S. E. Fahlman, "An Empirical Study of Learning Speed in Back-Propagation Networks," Technical Report CMU-CS-88-162, Carnegie Mellon University, June 1988

[6] A. Feldmann, A.C. Gilbert, W. Willinger, "Data Networks as Cascades: Investigating the Multifractal Nature of Internet WAN Traffic," In Proc. of ACM SIGCOMM, September 1998.

[7] A. Giovanardi, R. Rovatti, G. Mazzini, "Queue System Analytical Study with Self-Similar Chaos-Based Traffic," Electronics Letters , Volume: 37 Issue: 3 , PP 169-170, Feb. 2001.

[8] G. Gomes, N. L. S. da Fonseca, N. Agoulmine, J. N. de Souza, "Neurocomputation of the Hurst Parameter," In Proc. of IEEE ITS, 2002.

[9] K. M. Hellstern, P. Wirth, "Traffic Models for ISDN Data Users: Office Automation Application," Proc. ITC-13, Denmark, 1991.

[10] W. E. Leland, W. Willinger, M. S. Taqqu, D. V. Willson, "Statistical Analysis and Stochastic Modeling of Self-Similar Datatraffic," ITC Vol. 14, PP. 319-328, 1994.

[11] W. E. Leland, W.Willinger, M. S. Taqqu, D. V. Willson, "On the Self-Similar Nature of Ethernet Traffic," IEEE/ACM Trans. on Networking, Vol. 2, NO. 1, PP. 1-15, Feb. 1994.

[12] M. Minsky, S. A. Papert, "Perceptrons: An Introduction to Computational Geometry," MIT Press, Cambridge, MA, expanded edition, 1988/1969.

[13] V. Ramaswami, "Traffic Performance Modeling for Packet Communication: Whence, Where, and Whither," In Proc. of Australian Teletraffic Seminar, Nov. 1988.

[14] W. Willinger, M. Taqqu, "Self-Similarity through High-Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level," IEEE/ACM Trans. on Networking, Vol. 5, No. 1, Feb. 1997.

[15] H. Yousefi'zadeh, "Neural Network Modeling of Self-Similar Teletraffic Patterns," In Proc. of the First Workshop on Fractals and Self-Similarity, The 8-th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, July 2002.

[16] H. Yousefi'zadeh, E. A. Jonckheere, J. A. Silvester, "Utilizing Neural Networks to Reduce Packet Loss in Self-Similar Teletraffic Patterns," In Proc. of IEEE ICC, May 2003.