TECHNICAL RESEARCH REPORT

Traffic Models for the OPNET Simulator of the ALAX

by S. Rao

CSHCN T.R. 95-20
(ISR T.R. 95-117)

The Center for Satellite and Hybrid Communication Networks is a NASA-sponsored Commercial Space Center also supported by the Department of Defense (DOD), industry, the State of Maryland, the University of Maryland and the Institute for Systems Research. This document is a technical report in the CSHCN series originating at the University of Maryland.

Web site http://www.isr.umd.edu/CSHCN/

Traffic Models For the OPNET Simulator of the ALAX

Sandeep Rao
Electrical Engineering Department
and Institute for Systems Research,
Laboratory for Advanced Switching Technologies,
University of Maryland, College Park, MD 20742

1 Introduction

Traffic modeling is an important component of the design of any communication network. This is even more crucial for emerging networks which are expected to operate in high speed and high bandwidth environments. As the design of a network depends to a great extent on the type of traffic it is expected to carry, it is essential to characterize the traffic that the network is expected to carry. In the absence of traffic models the only way to validate and refine a network design would be to simulate the network using real life traffic sources. For any meaningful conclusion to be reached the simulation must be repeated for a lot of such real traffic sources, and in the end we still cannot be sure whether we have "covered all the cases". This is were traffic models come in very handy for they allow a parametrization of the essential characteristics of the network loads. So by generating the traffic under these traffic models for a range of their parameters and then running simulations for each of these generated sources we can say with greater confidence that the network has been tested under all the different traffic variations possible.

A good traffic model is one which is able to capture the characteristics of the traffic, as accurately as possible, with a minimum number of parameters. The most important characteristics of a traffic model that have a bearing on network performance are (i) **correlation** and (ii) **burstiness**. Correlation is that between bit rates at different time instants, and burstiness refers to the fact that time instants during which the number of bits generated is high tend to occur in clusters. High

burstiness in the source means that we can expect rather large intervals of time during which the source will have a high rate. These will be interspersed with time intervals during which the source has a low bit rate. Burstiness can be often be considered as a manifestation of positive correlation.

A high degree of burstiness and/or correlation in the input traffic can manifest itself through (i) **large packet delays** and (ii) **buffer overflow** at the switch nodes (and hence **cell loss**). It is essential that the traffic model used in simulations accurately reflects these traffic characteristics. Only then will a network design based on this model be able to take care of all the eventualities mentioned above.

A good survey of many of these issues is contained in [**?**]. Here we concentrate on traffic models which are planning to use in conjunction with the OPNET model of ALAX.

2 Types of Traffic models

Traffic models basically fall into two categories: (i) **short range** dependent models and (ii) **long range** dependent models. Both short and long range dependence refer to properties of wide–sense stationary stochastic processes (i.e., time series which have a constant mean and a covariance function which depends only on the time difference.) Specifically, let $X = \{X_t,\ t = 0, 1, 2, \ldots\}$ be a scalar process. We define its mean m and covariance function r by

$$m(t) \equiv \mathbf{E}\left[X_t\right] \quad \text{and} \quad r(s,t) \equiv \mathbf{E}\left[X_s X_t\right] - m(t)m(s), \quad s,t = 0, 1, \ldots \tag{2.1}$$

The scalar process $X = \{X_t,\ t = 0, 1, 2, \ldots\}$ is said to be **wide–sense stationary** process if

$$m(t) = m(0) \quad \text{and} \quad r(s,t) = r(|t-s|), \quad s,t = 0, 1, \ldots \tag{2.2}$$

In other words, the mean function is constant and the covariance function depends on the arguments s and t only through the difference $|t-s|$.

Short range dependent models (SRD): The defining characteristic of this class of wide–sense stationary models is the **summability** of the covariance function, i.e.,

$$\sum_{h=0}^{\infty} r(h) < \infty \tag{2.3}$$

However a lot of short range dependent models which are often cited in the literature satisfy a much stricter property, namely they have an **exponentially** decaying

covariance function . This exponentially decaying covariance function implies that the lengths of packets generated at two time instants very far apart will not be correlated. The rate of the exponential decay can often be expressed as a function of the parameters defining the model. Classical models such as **autoregressive** and **Markov** models are short range dependent; in fact they all have an exponentially decaying covariance function.

Long range dependent models (LRD) : A wide–sense stationary stochastic process is said to be LRD if the covariance function is **not** summable, i.e.,

$$r(h) = \sum_{h=0}^{\infty} \quad \text{diverges.} \tag{2.4}$$

This non-summability of the correlations captures the intuition behind long-range dependence, namely that though the high lag correlations might be individually small, their cumulative effect counts and gives rise to features which are drastically different from SRD processes. Any process with a **hyperbolically** decaying covariance function, namely

$$r(k) \sim k^{-D} \ (k \to \infty) \quad \text{with } 0 < \mathrm{D} < 1 \tag{2.5}$$

satisfies this criterion. In fact this much stricter condition is often cited in the literature (albeit erroneously) as the definition of a LRD process. This hyperbolic decay, being much slower than an exponential decay, also intuitively emphasizes the notion of long range dependence: Two packets generated at two time instants very far apart may still have a considerable amount of correlation. Examples of long range dependent models are the **Fractional Gaussian Noise** model [**?**, **?**, **?**] and the model based on the $M/G/\infty$ queue (to be discussed later).

Self–similar processes are a class of processes which are often used to generate LRD series. Formally, a scalar stochastic process $Y = \{Y_t,\ t \geq 0\}$ is said to be a self–similar process with self–similar parameter H if $\{Y_{at},\ t \geq 0\}$ and $\{a^H Y_t,\ t \geq 0\}$ have identical finite dimensional probability distributions. Mathematics apart, a self–similar process basically looks the same on any time scale with the absolute time scale playing no distinguishing role. A self–similar process can be made to display long range dependence by suitable choice of the parameters defining the particular self–similar model.

In recent years increasing evidence has accumulated that points to the (asymptotically) self–similar nature of aggregate packet streams in a wide range of currently working packet networks, e.g., Ethernet LANs [**?**, **?**, **?**], VBR traffic [**?**], WAN traffic

[?, ?]. This self–similarity manifests itself most crisply through a long–range dependence effect [?, ?] which is characterized by the autocorrelation of the traffic process obeying a power law (in the lag time). Long–range dependent processes are inherently non–Markovian, and have the property that while long–term correlations are individually small, they nevertheless accumulate in the long run to create scenarios which are drastically different from those produced by more traditional, typically Markovian in nature, short–range dependent models.

This established presence of long–range dependence over a wide range of time scales in packet traffic processes is expected to have a significant impact on queueing performance when such processes are offered to a muliplexer. In fact, if LRD traffic is passed through a network designed only for SRD traffic, it can have a drastic effect in terms of cell loss and cell delay. Networks to be engineered for LRD traffic in general will have to be designed very differently from networks required to carry only SRD traffic; for instance they might need larger buffers at the switching nodes, faster processing for a given delay, etc.

3 Models considered for the ALAX simulation

For the purpose of designing the ALAX, we have developped the following models in the OPNET environment: (i) **Markov–modulated Poisson processes** (MMPP); (ii) **Autoregressive (AR)–Markov** hybrid processes [?]; and (iii) **LRD** processes based on the $M/G/\infty$ queue. Models (i)–(ii) are SRD while (iii) is LRD. We are using these source models in simulations which are currently being run in conjunction with the OPNET model for the ALAX in order to evaluate performance.

4 Markov–Modulated Poisson Processes

Markov–modulated Poisson processes are driven by an underlying continuous–time finite–state Markov chain with rate matrix $R \equiv (r(i,j))$ which is assumed to be ergodic (i.e., irreducible and positive recurrent). Associated with each state $i = 1, \ldots, m$, there is a Poisson process with parameter $\lambda_i > 0$. The corresponding Markov–modulated Poisson process is defined as follows: As long as the underlying Markov chain is in state i, events (or arrivals) are generated according to a Poisson process with rate λ_i. Each arrival of the Poisson process is considered as a packet arrival (the packets are of fixed size). Basically what happens is the following: (i)

The underlying Markov chain enters state i; (ii) It stays in state i for an amount of time T_i which is exponentially distributed with parameter $r(i,i)^{-1}$. During this period packets (of constant size) arrive according to a Poisson process of rate λ_i; (iii) At the end of the time interval of length T_i, the underlying Markov chain jumps to state j $(i \neq j)$ with probability p_{ij} given by

$$p_{ij} \equiv \frac{r(i,j)}{\sum_{k \neq i} r(i,k)}, \quad i \neq j \tag{4.6}$$

(with the natural convention $p_{ii} = 0$.)

Let $p = (p(1), \ldots, p(m))$ denote the vector of steady–state probabilities for the underlying Markov chain with rate matrix R. The vector p solves the equation

$$pR = 0 \quad \text{and} \quad p(1) + \ldots + p(m) = 1. \tag{4.7}$$

Then, the average rate of this source is given by

$$\lambda_{av} = \sum_{i=1}^{m} \lambda_i p(i) \tag{4.8}$$

We have considered two implementations for simulating this model: One implementation is an exact representation of the model but is computationally intensive. The second implementation trades off accuracy for savings in computation. Assume that the chain is in state i and that it is going to stay there for time T_i. During this period events are generated according to a Poisson process with rate λ_i. The two implementations essentially differ in the way this Poisson process is generated.

Implementation 1: Start with $t = 0$; the process has just entered state i and will stay there for a time interval T_i.

(i) Generate τ, a random variable which is exponentially distributed with parameter $1/\lambda_i$. This is basically the interarrival time of the Poisson process.

(ii) Wait for an amount of time τ, and at the end of this interval generate a packet; this is the Poisson arrival)

(iii) Set $t = t + \tau$. If $t < T_i$ goto (i); else stop.

The problem with this approach is that if the rate of the Poisson process is very high, the interarrival time will be correspondingly small. This will result in a lot of interrupts being generated when the simulation is run in OPNET and hence will slow down the simulation. Also when the Poisson rate is very high it might

be unnecessary to implement the arrival process at such a detailed level such that every arrival takes place exactly at the time it is supposed to occur. An alternative would be choose some quantization time scale δ and make all the arrivals which were supposed to occur in that time interval to arrive in one batch at the end of the interval δ. Clearly, as δ increases we save more and more in terms of computation but lose out on the accuracy of the model.

Implementation 2: Start at $t = 0$; the process has just entered state i and will stay there for a time interval T_i.

(i) Wait for a time δ. At the end of this time interval generate N packets with N distributed according to a Poisson distributed random variable with rate $\lambda \equiv \lambda_i \delta$, i.e.,

$$\mathbf{P}\left[N = n\right] = \exp^{-\lambda} \lambda^n / n! \tag{4.9}$$

(ii) Set $t = t + \delta$. If $t < T_i$, then go back to (i); else stop.

5 An AR–Markov hybrid process

This model, which was proposed in [**?**], is a good SRD model for VBR video traffic. The bit rate creation pattern for a VBR source is basically modeled by a three–state Markov chain (Fig 1). The states 2 and 3 correspond to a scene change, while the time spent in state 1 corresponds to the time spent in a particular scene. As long as a particular scene is being displayed the chain stays in state 1. Whenever there is a scene change (which happens with probability p), the chain goes sequentially through states 2 and 3, before returning to state 1. It then stays in state 1 until the next scene change. This model basically generates a discrete time sequence $\{X(n),\ n = 0, 1, \ldots\}$ where $X(n)$ is interpreted as the number of bits produced by a slotted source in the n^{th} time slot.

In state 1 the output is produced as a sum of two first order AR processes (2–AR process). The autocorrelation of a VBR source shows a sharp drop at low lags and a slow decay at higher lags. The exponentially decaying autocorrelation function of a single first order AR process was found to be insufficient to model this behavior. The autocorrelation function of a 2–AR process is basically the sum of 2 exponentials. It was found that with two exponentials there was sufficient freedom to choose the parameters of these exponentials so as to approximate the autocorrelation of the

VBR video source to a reasonably good extent.

To see this, recall that a first order AR process is represented as

$$X(n) = aX(n-1) + bw(n), \quad n = 0, 1, \ldots \tag{5.10}$$

where $\{w(n),\ n = 0, 1, \ldots\}$ is a Gaussian noise process and a, b are fixed parameters. Therefore a 2–AR process can be represented as

$$X(n) = \sum_{i=1}^{2} X_i(n), \quad n = 0, 1, \ldots \tag{5.11}$$

where $\{X_i(n),\ n = 0, 1, \ldots\}$, $i = 1, 2$, are first order AR processes, say

$$X(n) = a_i X(n-1) + b_i w_i(n),\ i = 1, 2; \quad n = 0, 1, \ldots \tag{5.12}$$

The autocorrelation of a 2–AR process is given by

$$C_r(h) = \sum_{i=1}^{2} \frac{a_i^h b_i^2}{(1 - a_i^2)}, \quad h = 0, 1, \ldots \tag{5.13}$$

while the mean and variance are given by

$$\mathbf{E}[X] = \sum_{i=1}^{2} \frac{a_i b_i}{(1 - a_i)} \mu(i) \tag{5.14}$$

and

$$\sigma^2 = \sum_{i=1}^{2} \frac{b_i^2}{(1 - a_i^2)} \tag{5.15}$$

where $\mu(i)$ denotes the mean of the noise process $\{w_i(n),\ n = 0, 1, \ldots\}$, $i = 1, 2$.

Knowing the values of the mean and variance of the orginal source and its autocorrelation for two different lags (h), we can solve the above equations to get the values of a_i and b_i, $i = 1, 2$.

The states 2 and 3 were introduced to model the bit rate increment effects during scene changes. These are modeled as a Gaussian processes with a fixed mean and variance.

The average rate for this model is given by the formula

$$Rate_{av} = p(1)\mathbf{E}[X] + p(2)\mu_2 + p(3)\mu_3 \tag{5.16}$$

where

$$p(1) = (1 + 2p)^{-1} \quad \text{and} \quad p(2) = p(3) = p(1 + 2p)^{-1} \tag{5.17}$$

are the steady–state probabilities for the underlying Markov chain. Also, $\mathbf{E}[X]$ is the mean rate of the AR–2 process, while μ_2 and μ_3 are the means of the Gaussian processes in states 2 and 3, respectively.

6 A LRD process based on a $M|G|\infty$ model

The $M|G|\infty$ model provides a method for generating a self-similar process. Indeed, by suitable choice of one of its defining parameters, viz. α, it can be made to display long range dependence, a fact which appeared to have been mentioned first by Cox in [**?**]. Additional details are available in [**?**].

This process is a discrete–time process, in the sense that every δT units of time the process generates a number which is interpreted as the number of bits generated by the discrete–time source at that particular time instant. So basically this source generates a sequence $\{N(n\delta T),\ n = 0, 1, \ldots\}$.

More precisely, consider the following queueing system (Fig.2): Time is slotted with δT being the length of a time slot, and customers arrive to the system according to a Poisson process with rate λ. Upon arrival customers are offered to an infinite server group, and the required service times are i.i.d. finite mean random variables – let σ denote the generic service time random variable (expressed in number of time slots) and let G denote its probability distribution.

As there is an unlimited number of servers, any incoming packet is immediately assigned a server (the packets do not have to wait for service), and begins service in the next time slot following its arrival. Such a queueing construct is represented by $M|G|\infty$, where M refers to the exponential interarrival times (of the Poisson arrivals), G refers to a general service distribution and ∞ refers to the number of servers (unlimited in this case).

Here, we assume that the generic service time σ (when expressed in terms of number of slots) has a a Pareto distribution with parameter $\alpha > 0$, say

$$P(\sigma > r) = (r + 1)^{-\alpha}, \quad r = 0, 1, \ldots \tag{6.18}$$

We take $1 < \alpha < 2$ to ensure that σ has a **finite** first moment.

The number of busy servers at any instant $t \geq 0$ is denoted by $N(t)$, and we write

$$N(n) \equiv N(n\delta T), \quad n = 0, 1, \ldots \tag{6.19}$$

so that $N(n)$ is simply the number of busy servers at times $n\delta T$, $n = 0, 1, 2, \ldots$. It can be shown that in **steady state** the random variables $\{N(n),\ n = 0, 1, \ldots\}$ constitute the LRD sequence we set out to generate for it has the following properties:

(i) For each $n = 0, 1, 2, \ldots$, the random variable $N(n)$ is a Poisson random variable with parameter $\lambda_d \mathbf{E}[\sigma]$ where λ_d is defined by $\lambda\delta T$.

(ii) Its covariance structure is given by

$$r(h) \equiv r(t, t+h) = \lambda_d \mathbf{E}\left[(\sigma - h)^+\right], \quad t, h = 0, 1, \ldots \tag{6.20}$$

where the quantities are finite under the finite moment assumption made earlier for σ. The above equation can be written as

$$r(h) = \lambda_d \sum_{n=h+1} n^{-\alpha}, \quad h = 0, 1, \ldots \tag{6.21}$$

Using this (**??**) it can be shown that

$$\lim_{h \to \infty} \frac{r(h)}{h^{\alpha-1}} = \lambda_d (\alpha - 1)^{-1}. \tag{6.22}$$

In particular, this limit implies the self–similar nature of the process $\{N(n),\ n = 0, 1, \ldots\}$. A similar limit for a process with an exponentially decaying correlation function would be zero. The parameter α controls the burstiness of the source. It is related to another parameter called the Hurst parameter H (which is commonly used to define burstiness in LRD sources) through the relation

$$H = 1 - \frac{1}{2}(\alpha - 1) = \frac{1}{2}(3 - \alpha). \tag{6.23}$$

Note that $0.5 < H < 1$ if $1 < \alpha < 2$, whence the process $\{N(n),\ n = 0, 1, \ldots\}$ exhibits long range dependence. In [**?**] the $M|G|\infty$ model was shown to occur in a very natural manner: An aggregate traffic model was constructed by superposing a large number of on–off sources with Pareto distributed activity periods, and shown in the limit to coincide with the $M|G|\infty$ model discussed here.

The average number of bits per unit time is given by $\mathbf{E}[N(n)]$ (which is independent of n in steady state). Therefore, from (i) we immediately get that the average rate is given by

$$Rate_{av} = \lambda \delta T . \mathbf{E}[\sigma] \tag{6.24}$$

It has been found that low activity scenes like video conferencing or videophone have a lower value of H than high activity scenes such as TV shows or motion pictures. Typically low activity scenes can be characterized by a value of H in the range of $(0.5, 0.75)$, while values of H from 0.75 to 1.0 correspond to high activity scenes.

Since computer simulations require everything to be discrete, the continuous–time arrival process will have to be somehow discretized. However in our case it

turns out that this discretization can be done without any loss of accuracy. This is because since we are interested only in time instants $n\delta T$, the continuous arrival process can be equivalently taken to be a discrete process with the arrivals occuring at only discrete time instants. The number of arrivals at these time instants is then Poisson distributed with parameter $\lambda\delta T$. In the simulations we keep track of the number of busy servers through an array. The i^{th} entry in this array stores the residual time that has to elapse before that particular server completes service (this is updated every δT time units). A free server has an entry of zero in this array. Though theoretically the number of servers can be unlimited, the array size can in practice only be finite. The only way to get around this problem is to define an array size that is sufficiently large, so that for the range of parameters that we wish to simulate the number of busy servers never goes beyond the declared array size. A value of 10^6 was found to be sufficient in our case, were the LRD source has a rate of the order of 10^4. The simulation model has a provision for a scaling factor which multiplies the number of bits generated at each time instant by a prespecified factor. With this provision the model can be scaled to any desired bit rate.

This source has been used to model a VBR source in our simulations. A VBR source generates bits every 1/30th of a second, so that $\delta T = 1/30$ and the rate of the source will be in the Megabits range.

References

[1] J. Beran, *Statistics for Long-Memory Processes*, Chapman and Hall, New York (NY), 1994.

[2] J. Beran, R. Sherman, M. S. Taqqu and W. Willinger "Long-range dependence in variable bit–rate video traffic," *IEEE Transactions on Communications* **COM–43** (1995), pp. 1566–1579.

[3] D. R. Cox, "Long–Range Dependence: A Review," *Statistics: An Appraisal*, H. A. David and H. T. David, Eds., The Iowa State University Press, Ames (IA), 1984, pp 55–74.

[4] D. R. Cox and V. Isham, *Point Processes*, Chapman and Hall, New York (NY), 1980.

[5] A. Erramilli, O. Narayan and W. Willinger, "Experimental queueing analysis with long–range dependent packet traffic," Preprint 1994.

[6] S. Frost and B. Melamed, "Traffic modeling for telecommunication networks," IEEE Communications Magazine, March 1994.

[7] H. J. Fowler and W. E. Leland, "Local area network traffic characteristics, with implications for broadband network congestion management," *IEEE Journal on Selected Areas in Communications* **JSAC–9** (1991), pp. 1139–1149.

[8] M. Garrett and W. Willinger, "Analysis, modeling and generation of self–similar VBR video traffic," *Proceedings of SIGCOMM '94*, September 1994, pp. 269–280.

[9] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the self–similar nature of ethernet traffic (extended version)," *IEEE/ACM Transactions on Networking* **2** (1994), pp. 1–15.

[10] N. Likhanov, B. Tsybakov and N.D. Georganas, "Analysis of an ATM buffer with self–similar ("fractal") input traffic," in *Proceedings of Infocom '95*, Boston (MA), April 1995, pp. 985–992.

[11] I. Norros, "A storage model with self–similar input," *Queueing Systems – Theory & Applications* **16** (1994), pp. 387-396.

[12] M. Parulekar, *Buffer Engineering for Self–Similar Traffic*, Ph.D. Thesis, Electrical Engineering Department, University of Maryland, College Park (MD). Expected December 1996.

[13] M. Parulekar and A.M. Makowski, "Buffer overflow probabilities for a multiplexer with self–similar traffic," Infocom'96, San Francisco (CA), March 1996. In press.

[14] V. Paxson and S. Floyd, "Wide area traffic: The failure of Poisson modeling," *IEEE/ACM Transactions on Networking* **3** (1993), pp. 226–244.

[15] C. Shim, J. Lee and S.B. Lee, "Modeling and call admission control of VBR video," *IEEE Journal on Selected Areas in Communications* **JSAC–12** (1994), pp. 332–344.

[16] W. Willinger, M. S. Taqqu, W. E. Leland and D. V. Wilson, "Self–similarity in high–speed packet traffic: Analysis and modeling of ethernet traffic measurements," *Statistical Science*, to appear.