TECHNICAL REPORT

Traffic modeling in a multi-media environment *

Selvakumaran N. Subramanian and Tho Le-Ngoc

March 1996

Center for Signal Processing and Communications
Department of Electrical and Computer Engineering
Concordia University
1455, de Maisonneuve Blvd. West
Montréal, Québec
Canada H3G 1M8
E-mail: selva@ECE.Concordia.ca and tho@ECE.Concordia.ca

*This work was supported by a grant from the CRC and SPAR Aerospace Limited

SUMMARY:

This final report presents the results obtained in the 2-year CRC/SPAR RESEARCH PROJECT: STATISTICAL TRAFFIC MODELLING IN WIDEBAND MULTIMEDIA SATELLITE COMMUNICATIONS. The overall project has six following steps:

- Step 1: Survey of voice traffic models,
- Step 2: Survey of constant bit rate video traffic models,
- Step 3: Survey of variable bit rate video traffic models,
- Step 4: Study of data traffic models,
- Step 5: Study and development of multimedia traffic models,
- Step 6: Performance evaluation of queueing and multiaccess schemes using the developed model

Interim reports have been delivered at the end of each step. This final report integrates these interim reports and includes new results. It is organized as follows.

Main technical results obtained in Steps 1-6 are presented in the Technical Report *Traffic Modeling in a Multimedia Environment.* After a short introduction in Chapter 1, we present a survey of voice, video and data traffic models in Chapters 2. In Chapter 3, we discuss the modeling of aggregate multimedia traffic, present our proposed Pareto Modulated Poisson Process (PMPP) Model for long-range dependent video or data traffic sources, and the proposed model for aggregate multimedia traffic. Chapter 4 provides the conclusions and suggested further research. The derivations of the Index of Dispersion for Counts (IDC) of the MMPP and PMPP are included in the Appendix.

Attachment #1 is the *User Manual of the Traffic Generator*. It describes the structure and use of the developed traffic generator for a multimedia environment on OPNET. This traffic generator can be used to evaluate the performance (e.g., loss probability, blocking probability, delay,...) of queueing and communications networks by simulation using OPNET.

Attachment #2 is the Technical Report *Performance of CFDAMA in a Multimedia SATCOM System using MF-TDMA.* It presents the simulation results on the performance of the Combined Free/Demand Assignment Multiple-Access scheme in a multimedia SATCOM environment by using the developed traffic generator and OPNET.

CRC/SPAR RESEARCH PROJECT:

STATISTICAL TRAFFIC MODELLING IN WIDEBAND MULTIMEDIA SATELLITE COMMUNICATIONS

FINAL REPORT

Prepared by
Tho Le-Ngoc, Principal Investigator
Concordia University

Submitted to
Chun Loo, Technical Representative, CRC
and
M.R. Soleymani, Technical Representative,
Spar Aerospace Limited

April 15,1996

Abstract

With the advent of B-ISDN (Broadband ISDN), significant effort has been devoted to supporting real time traffic such as voice and video along with jitter tolerant traffic such as data traffic, in a packet switched environment. The wide spectrum of traffic sources exhibitsa diverse mixture of traffic characteristics. Hence it is imperative to develop a model that aptly characterizes the variability and statistical correlations of the packet arrival process. In this project we survey the existing traffic models available in the literature and also propose a new model, consisting of doubly stochastic Poisson processes (the PMPP and MMPP) for aggregate traffic. A traffic generator comprising the standard traffic model is also built. The performance of a G/D/1 queue and performance of the CFDAMA (Combined free/demand assignment multiple access) protocol are evaluated using the proposed model, by simulation.

Contents

Chapter 1

Introduction

The evolving BISDN (Broadband ISDN) networks provide bearer service supporting real time traffic such as voice and video traffic along with jitter tolerant traffic such as data traffic. These wide spectrum of traffic sources (such as computer data, VBR video and voice, etc.,) exhibit a diverse mixture of traffic characteristics. Also, through statistical multiplexing, several of these individual sources may share a high transmission rate link capacity. Designing and managing these evolving networks require prediction of network performance. Analytical techniques, computer simulation, projections from existing data are methods that are used to evaluate and design networks.

Traditionally the Poisson model has been used as the model for characterizing packet traffic. However, recent studies indicate that these models are no longer applicable to the diverse mixture of traffic present in the broadband networks. Hence it is imperative to develop a model that aptly characterizes the variability and statistical correlations in the aggregate packet arrival process. This model may then be used to evaluate the network performance (QOS, utility, etc.) or to evaluate the connection admission control and source policing algorithms. Also, recent studies in LAN data traffic indicate that such data traffic exhibits long-range dependence and self-similar (or fractal) characteristics, i.e., the traffic exhibits "burstiness" across a wide range of time scales ranging from milliseconds to hours. Hence in a multi-media environment fractal traffic co-exists with non-fractal traffic. Characterizing such a mix of traffic by an unique model poses a great challenge to the modeler. The model proposed should be versatile in the sense that it should be able to capture the long term and short term correlations . In this project we propose an traffic model to characterize the traffic in a multi-media environment.

The key objectives of this project are as follows:

- To study the various models proposed in the literature for the various constituents of the aggregate traffic (voice, video and data traffic).
- To develop a traffic generator in the OPNET environment, comprising of - standard traffic models (such as on/off model, MMPP, etc.).
- To study and propose a new traffic model that accurately characterizes the aggregate traffic.
- To investigate the queueing performance of the aggregate traffic model through simulation.
- To evaluate the performance of the CFDAMA (Combined free/demand assignment multiple access) protocol using the traffic model proposed.

This report summarizes the results of the study conducted in this project. This report is organized as follows. The following chapter surveys and classifies the various models proposed for voice, video and data traffic. In chapter 3, we propose a new model for aggregate data traffic and present the simulation results for this model. The queueing performance of this model is also presented in this chapter. Chapter 4 concludes the results of the study conducted in this project. Attachment #1, "User Manual of traffic generator", explains the usage of the traffic generator, built in the OPNET environment and Attachment #2, "Performance of CFDAMA in multimedia SATCOM system using MF-TDMA", presents the results of the analysis of the CFDAMA protocol using the proposed model.

Chapter 2

Survey of traffic models

2.1 Survey of voice traffic models

With the advent of B-ISDN significant effort has been devoted to supporting real time communication application such as real time voice and video in a packet switched environment. In such a multiplexing environment, the packets from many sources are statistically multiplexed on to a single high speed link in order to exploit the bursty nature of the sources. Such a statistical multiplexing introduces different delays to packets. Real time traffic (like voice and video) are delay sensitive (loss insensitive) while data traffic is loss sensitive (delay insensitive). Hence in packet networks supporting real time traffic delay is bounded at the expense of some loss. However, in order to meet a required grade of service the loss of packets have to be kept within a certain limit. This necessitates that the buffer used to queue the packets in the statistical multiplexer be engineered to keep the delay and packet loss within specified limits. In order to do so, a thorough understanding of the packet arrival process to the statistical multiplexer and simple but accurate models to analyze such a system are required. Traditionally a Poisson approximation has always been adopted to characterize the packet arrival process. But recent studies indicate that the packet arrival process to the multiplexer is highly correlated and that the Poisson approximation for the arrival process results in erroneous results since it fails to account for these correlations.

The queueing behaviour when voice is coded with silence detection is different from that when voice is coded without silence detection. Many models have been proposed to characterize the superposition arrival process of statistically multiplexed voice (with silence detection). The arrival process in such a superposition is found to have a strong positive correlation and a Poisson approximation results in serious underestimation of delay.

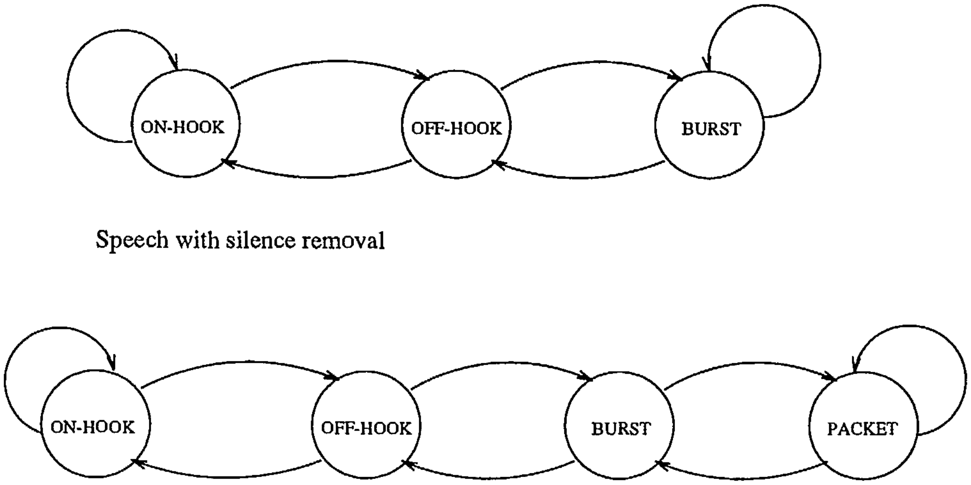
The superposition arrival process in continuous bit rate (CBR) traffic (such as voice without silence detection and constant bit rate video) is found to have negative correlations and a Poisson approximation here, results in overestimation of delay.

2.1.1 Modeling of a single voice source

Traditionally the voice source (telephone) in a circuit switched environment has been modeled as a 2 state process. The source is either in the *OFF* state (on-hook) or *ON* state (off-hook). The length of an off-hook period corresponds to the duration of the *call* and is called the *call holding time*. It should however be noted that during each call, the user is not always talking and there are periods of *silence* between successive *bursts* or *talkspurts.*

With the digitization of speech and introduction of Digital Speech Interpolation (DSI) techniques [1], the voice source transmits only when there is speech activity. Such a system has been modeled by 3 states: on-hook, off-hook and burst. If each burst is packetized for transmission, a fourth state is needed which represents the state of a packet transmission during the burst state, as shown in Figure 2.1.

The above model characterizes the single source at a higher level (i.e., at a call level). Different approach has been followed to characterize the single source at the packet level. In this model the voice source is *active* when there is speech activity (i.e., the talker is actually speaking) and during these times the voice source periodically generates fixed length packets. A voice source is *inactive* when the speaker is silent (during the course of the call) and during these times the voice source does not



Speech with silence removal

Packetized voice with silence removal

Figure 2.1: Call level models for a single voice source

generate packet, Figure2.2. Experimental results have proved that the duration of the active periods fits the exponential distribution very well, while the duration of the inactive period is not as well approximated by the exponential distribution [2, 3]. However for analytical simplicity the silence periods have always been modeled as exponentially distributed.

Single voice source - Model 1

If T ms is the packetization time then, the packet stream from a single voice source is characterized by arrivals at fixed intervals of T ms during talkspurts and no arrivals during silences. The talkspurts are assumed to be exponentially distributed with mean α^{-1} generating a geometrically distributed number of packets of mean α^{-1}/T. The silent periods are assumed to be exponentially distributed with mean β^{-1}. Under these assumptions the packet arrival process can either be treated as a renewal process (since the talkspurt and silence periods are independent and identically distributed) or as a 2 state (ON/OFF) discrete time (or continuous time) Markov chain with the transition rates from ON to OFF state equal to α and from OFF to ON state equal to β, Figure 2.3.

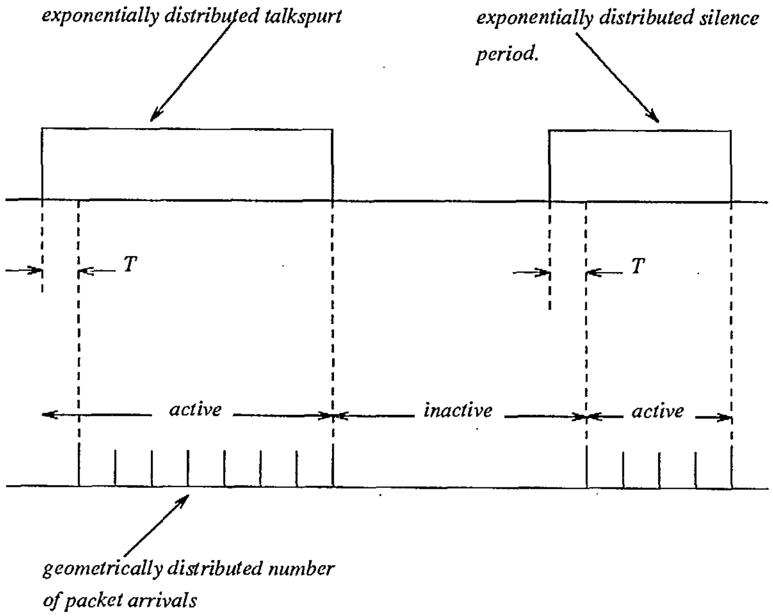

exponentially distributed talkspurt
exponentially distributed silence period.
T
T
active
inactive
active
geometrically distributed number of packet arrivals

Figure 2.2: The packet arrival process from a single voice source

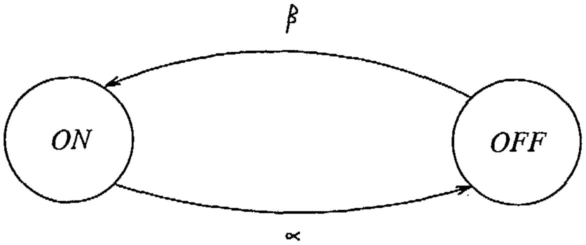


Figure 2.3: Two state continuous time Markov chain model

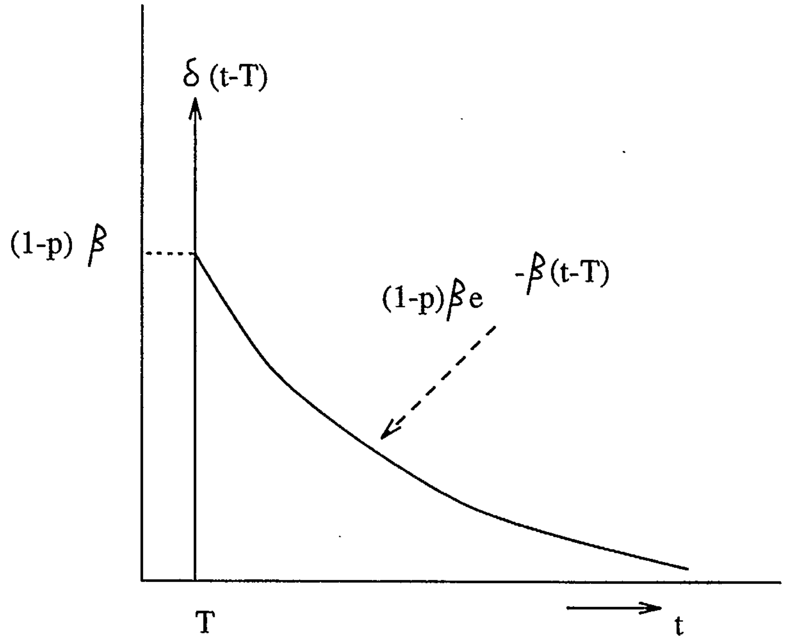


Figure 2.4: Probability density function for packet interarrival time from a single voice source.

For a packet of 64 bytes, coded with 32 Kbps ADPCM T = 16 ms. Typical values of $\alpha^{-1} = 352$ ms (with a mean of 352/16 = 22 packets) and $\beta^{-1} = 650$ ms [4]. The interarrival period for such a source is T ms for most of the packets and ocassionally greater than T ms, when there is a silence period in between. Hence the probability density function of the interarrival period, as shown in Figure 2.4 ([4])is as given below,

$$f(t) = p.\delta(t-T) + (1-p).\beta \exp^{-\beta(t-T)}$$

where p is the probability that a packet is followed by another packet after T ms and is given by p = $\exp^{-\alpha T} \approx 1 - \alpha T$. Therefore,

$$f(t) = (1-\alpha T)\delta(t-T) + \alpha T \beta \exp^{-\beta(t-T)} \tag{2.1}$$

The cumulative distribution function for the interarrival time F(t) is obtained by integrating f(t) and is given by

$$F(t) = [(1-\alpha T) + \alpha T(1 - \exp^{-\beta(t-T)})]U(t-T) \tag{2.2}$$

where U(t) is the unit step function.

The number of packets per talkspurt is geometrically distributed with mean equal to $1/\alpha T$, and the distribution is given by

$$P_i = (1-\alpha t)^{i-1}\alpha T \qquad i = 1,2,3,\ldots$$

The squared coefficient of variation (variance divided by the square of the mean) of an interarrival time is given by

$$c_1^2 = (1-p^2)/[T\beta + (1-p)]^2 \tag{2.3}$$

$c_1^2 = 18.1$ (with typical values). Hence the packet arrival process from a single voice source is highly bursty as is reflected by the high value of c_1^2 compared to that of a Poisson process which has a $c_1^2 = 1$.

Single voice source - Model 2

Another approach followed in characterizing an individual voice source is by approximating it as an Interrupted Poisson Process (IPP). Here again the talkspurt and silence period are assumed to be exponentially distributed, but the arrivals during the talkspurt are Poisson with a rate λ, rather than deterministic [5]. This process can be visualized as a Poisson Process which is alternately turned ON for an exponential period of time and then turned OFF for another independent exponential period of time - hence the name Interrupted Poisson Process.

2.1.2 Statistical Multiplexing of Voice

The schematic of a statistical multiplexer is as shown in Figure 2.5 [6]. The speech

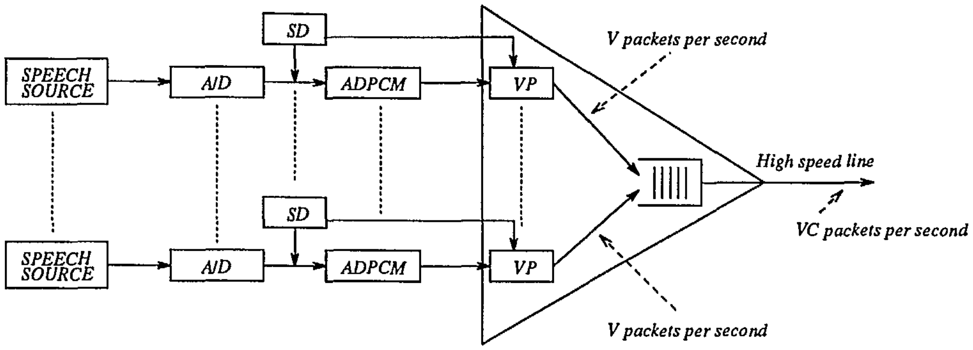


Figure 2.5: Schematic of a statistical voice multiplexer

signals generated by each source are digitized by an A/D coder generating a bit stream of 64 Kbps (8 bits per sample * 8000 samples per second). This may further be compressed by an ADPCM coder to produce a bit stream of 32 Kbps. Further if DSI [1] is used, then a speech detector (SD) is employed to monitor the output of the A/D converter continuously. It allows the voice packetizer (VP) to form voice packets only if there is speech activity on the input source.

There are N such sources each of which generates a packet every 1/V seconds. The packets so generated are fed to a common queue from which a server removes them for transmission over a communication link at the rate of VC packets per second. Hence C is the link capacity and is equal to the number of voice sources that will just saturate the link. Since on the average, less than half the lines will be active at any one time, the channel capacity C can in principle, be less than N. Hence during periods in which the number of active sources generating packets are more than C (i.e., more than C packets arrive in a frame consisting of C time slots, where a timeslot = τ ms is the transmission period of a packet), packets accumulate in the queue and this backlog of packets is eliminated during periods in which the number of active sources fall below C. Hence the packet arrival process is highly correlated and bursty[1] as it is dictated by the number of speech sources in talkspurt. It may also be noted that queueing of packets may also result due to the stochastics of arrival process, i.e.,

[1] the term bursty is used when referring to processes whose interarrival time distribution shows greater variability than Poisson process

2 or more packets arriving simultaneously. (However this queueing does not seem to cause significant amount of delay.)

The statistical multiplexing of voice has been studied extensively and treated analytically in the literature ([6, 4, 7, 8, 9, 10, 5, 11, 12, 13, 14, 15, 16]). All the studies concur that the modeling of aggregation of voice sources as a Poisson process gives erroneous results. The Poisson approximation works well only at low to moderate traffic intensities. (Traffic intensity or utilization of such a system is given by $\rho = (N/C)\,(1/\alpha/(1/\beta+1/\alpha) = (N/C)\,(\beta/(\alpha+\beta))$ where the quantities N,C,α^{-1} and β^{-1} are as defined previously.)

An excellent treatment of the superposition arrival process giving an intuitive explanation of the deviation of the process from that of a Poisson process is found in [4]. In [4, 7] the superposition non renewal point process is approximated by a renewal process with an inflated coefficient of variation, characterized by the indices of dispersion of counts (IDC) and intervals (IDI). It principally focuses on the dependence among successive interarrival times in the aggregate packet arrival process. Let $\{X_k, k \geq 1\ \}$ represent the sequence of packet interarrival times from the superposition process of N voice sources. Then, the index of dispersion for intervals (IDI), also called the k interval squared coefficient of variation sequence, is the sequence $\{c_k^2, k \geq 1\}$ defined by

$$c_k^2 = \frac{k\,Var\{X_1 + X_2 + \ldots + X_k\}}{E[\{X_1 + X_2 + \ldots + X_k\}]^2}$$

Assuming that $X_k, k \geq 1$ is stationary we note that the sum $X_1 + X_2 + \ldots + X_k = X_{i+1} + X_{i+2} + \ldots + X_{i+k}$. Denoting this sum by S_k we have

$$\begin{aligned} c_k^2 &= \frac{k\,Var(S_k)}{[E(S_k)]^2} \\ &= \frac{Var(S_k)}{k[E(X_1)]^2} \\ &= \frac{k\,Var(X_1) + \sum_{i,j=1;i\neq j}^{k} Cov(X_i, X_j)}{k\,[E(X_1)]^2} \end{aligned}$$

$$= \quad c_1^2 + \frac{2\sum_{j=1}^{k-1}(k-j)\, Cov(X_1 + X_{1+j})}{k\,[E(X_1)]^2} \qquad k \geq 1 \qquad (2.4)$$

For $k = 1, c_k^2 = c_1^2$ is the squared coefficient of variance (variance divided by square of the mean) of a single interarrival time. For $k > 1, c_k^2$ measures the cumulative covariance (normalized by the square of the mean) among k consecutive interarrival times.

The IDI has the following properties for the various processes A

a) For a Poisson process $c_k^2 = 1$ for all k.

b) For a renewal process $c_k^2 = c_1^2$ for all k.

c) For a stationary point process with positive correlation c_k^2

increases monotonically and the asymptote in the limit depends upon the sum of all correlation coefficients.

To define the Index of dispersion for counts (IDC) let A(t) denote the number of arrivals in an interval of length t, then

$$IDC = I(t) = \frac{Var[A(t)]}{E[N(t)]}, \qquad t > 0 \qquad (2.5)$$

The Poisson process has $I(t) = 1$ for all t.

It has been noted in [4, 8] that variability in the variance of the sum of consecutive interarrivals (or equivalently the variance of the arrival counts) is the major cause of packet queueing delays. The variability in the packet arrival process is strikingly revealed if we add together groups of n successive interarrival times and compare the variance of the resulting series with that of the original interarrival time series. It would be interesting to note that the variance of the times between n arrivals is much larger than n times the variance of the original series.

[4, 8] also point out that IDI and IDC can best characterize the variability. The figures 2.3, 2.4 of [4] show that for the voice superposition arrival process

a) IDC increases nonlinearly with t (indicating deviation from Poisson process)

b) IDI, c_{kn}^2 (where n denotes the number of processes superposed) tends to 1 for all k as $n \to \infty$.

c) IDI,$c_{kn}^2 \to c_{11}^2$ as $k \to \infty$ for all n.

These results imply the following:

i) Although the single interval in the superposition process tends to be an exponentially distributed variable, as the number of voice sources increase, positive correlations over many consecutive intervals exist (as indicated by $c_{kn}^2 > 1$ for $k > 1$) and this causes the process to be substantially deviated from a Poisson distribution. Hence the notion of relevant time scale is important while analyzing the superposition arrival process.

ii) Approximation of the arrival process thus depends upon identifying the relevant time scale and the relevant time scale in turn depends upon the traffic intensity in the queue (since the traffic intensity relates the arrival rate with the service rate).

iii) The correlation and traffic variability effects of the arrival process become significant only at higher loading (corresponding to an utilization ¿ 0.73 as noted in [4]). Under these conditions a Poisson approximation leads to underestimation of delays. This is due to the fact that as $\rho \to 1$ the traffic interaction in the queue spans over many intervals.

iv) The Poisson approximation holds good for low to moderate loading.

v) In terms of the buffer sizes the Poisson approximation holds good for the voice multiplexers with small buffer sizes. For multiplexers with large buffer sizes the Poisson approximation does not hold due to the effect of correlations in the successive interarrival times of the queued packets.

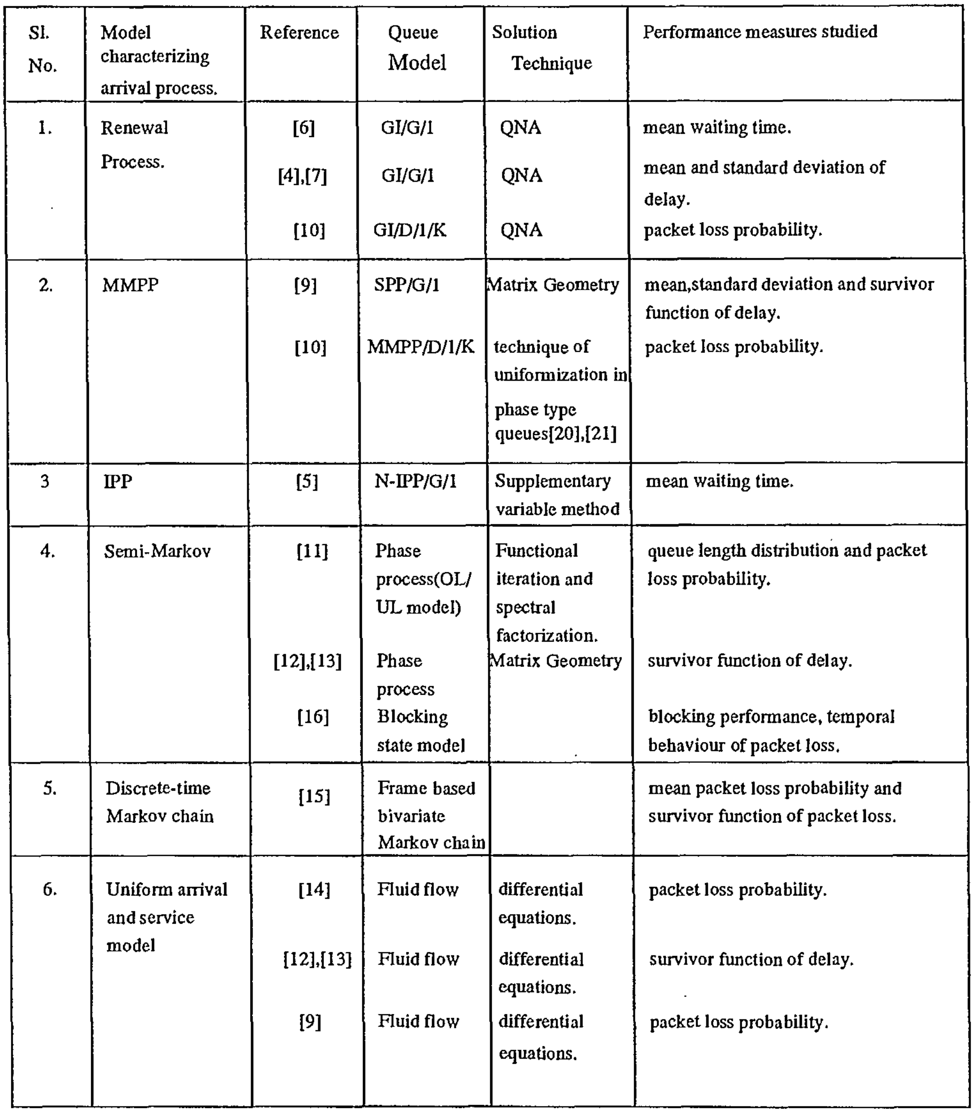
2.1.3 Modeling of Statistically multiplexed voice

Several studies ([6, 4, 7, 8, 9, 10, 5, 11, 12, 13, 14, 15, 16]) have dealt with the issue of characterizing the superposition of voice sources and analyzing the behaviour of the resulting queue. All of them concur that superposition process is not Poisson but they differ in their approaches to modeling the process or in the choice of the performance parameter evaluated using their model. While most of the models are used to evaluate the mean and standard deviation of delay as performance measures of the queueing system under consideration, few analysis like [10], [14], [15], [16] argue that for a multiplexer with a finite buffer, the average statistics like mean queue length and mean packet delay are no longer suitable performance measures. [10],[14],[15] choose the packet loss probability and maximum tolerable packet delay as the performance measures, while [16] also evaluates the temporal behaviour of packet loss. Table 1 gives an overview of the various models proposed in the literature to characterize the superposition arrival process of voice, and the respective performance measure the models were used to evaluate.

A brief description of each of the models used to characterize the superposition arrival process is given in the following sections.

Renewal Process

As observed earlier, the packet arrival process from a single source can be modeled as a renewal process with exponentially distributed talkspurts alternating with exponentially distributed silence periods. [6, 4, 7] approximate the superposition arrival process as a renewal process with inflated coefficient of variation for the interarrival time. A 2 parameter approximation technique as in [17, 18] called the Queueing Network Analyzer (QNA) approach is adopted. In this method the superposition



Sl. No.	Model characterizing arrival process.	Reference	Queue Model	Solution Technique	Performance measures studied
1.	Renewal Process.	[6]	GI/G/1	QNA	mean waiting time.
		[4],[7]	GI/G/1	QNA	mean and standard deviation of delay.
		[10]	GI/D/1/K	QNA	packet loss probability.
2.	MMPP	[9]	SPP/G/1	Matrix Geometry	mean,standard deviation and survivor function of delay.
		[10]	MMPP/D/1/K	technique of uniformization in phase type queues[20],[21]	packet loss probability.
3	IPP	[5]	N-IPP/G/1	Supplementary variable method	mean waiting time.
4.	Semi-Markov	[11]	Phase process(OL/ UL model)	Functional iteration and spectral factorization.	queue length distribution and packet loss probability.
		[12],[13]	Phase process	Matrix Geometry	survivor function of delay.
		[16]	Blocking state model		blocking performance, temporal behaviour of packet loss.
5.	Discrete-time Markov chain	[15]	Frame based bivariate Markov chain		mean packet loss probability and survivor function of packet loss.
6.	Uniform arrival and service model	[14]	Fluid flow	differential equations.	packet loss probability.
		[12],[13]	Fluid flow	differential equations.	survivor function of delay.
		[9]	Fluid flow	differential equations.	packet loss probability.

Table 1: Models for superposition for voice sources

Table 2.1: Models

arrival process is characterized by 2 parameters; one is the average arrival rate (λ) and the other is the squared coefficient of variation of the interarrival time (c_a^2). The squared coefficient of variation of the interarrival time of the renewal process may be approximated from the original superposition process by one of 2 methods:[17]

- *Stationary interval method*:Here the moments of the renewal interval is approximated with the moments of the stationary interval in the superposition arrival process.
- *Asymptotic method*: In this method the moments of the renewal interval is determined by matching the asymptotic behaviour of the moments of the sum of successive intervals.

The formula for the squared coefficient of variation of the interarrival time distribution (c_a^2) in the approximating renewal process for the aggregate packet arrival process is as given below ([4])

$$c_a^2 = w\, c_1^2 + (1 - w) \tag{2.6}$$

where

c_1^2 = squared coefficient of variation of a single voice source

$w = 1/[1 + 4(1 - \rho)^2(N - 1)]$

ρ = traffic intensity

N = number of sources multiplexed

The QNA approximation as given above selects an increasingly higher squared coefficient of variation c_a^2 as N increases (when ρ is kept constant), to directly capture the effect of covariance. (See Figure 5 of [4]).

The other parameter λ of the approximating renewal process can be found as $\lambda = N\,\lambda_1$ where λ_1 is the mean arrival rate of a single source.

Let τ and c_s^2 be the mean and squared coefficient of variation associated with the packet service time. ($c_s^2 = 0$ if the service time is constant)

Now, given the mean and squared coefficient of variation of the interarrival and service times ($\lambda, c_a^2, \tau, c_s^2$), the congestion measures for the queue such as the mean and standard deviation of delay can be obtained by regarding it as a GI/G/1 queue (with renewal arrival process). See [19] for specific formulas. The mean delay calculated by this model in [4] seems to agree well with simulation results especially at high traffic intensities, where the Poisson approximation fails.

[10] also uses this renewal process model with QNA approximations but introduces an additional heuristic needed to handle finite buffers. Given the distribution $P(Q_\infty = i)$ (probability that the queue length is equal to i) for an infinite buffer case, $P(Q_k = K)$ for a multiplexer with K buffers is approximated in [10] by

$$P(Q_k = K) = \frac{P(Q_\infty = K)}{P(Q_\infty \leq K)} \tag{2.7}$$

where $P(Q_\infty = K)$ is obtained as outlined before. An approximate method for solving 2.7 is given in [10].

Markov Modulated Poisson Process

Markov modulated Poisson process (MMPP) is a nonrenewal, doubly stochastic Poisson process where the rate process is determined by the state of a continuous time Markov chain. In other words in state k of the underlying Markov chain arrivals occur according to a Poisson rate λ_k. [9], [10] model the superposition arrival process as a 2 state MMPP.

In [9] the approximating MMPP is chosen in such a way that several of its characteristics identically match with those of the original superposition. There are 4 parameters for the 2 state MMPP chosen, namely, the mean sojourn times in states 1 and 2,r_1^{-1} and the Poisson arrival rates in states 1 and 2 λ_1 and λ_2 respectively. In order to determine these 4 parameters of the model the following 4 characteristics of the model are matched with those of the superposition process:

1. the mean arrival rate

2. the variance to mean ratio of the number of arrivals in an interval $(0, t_1)$

3. the long term variance to mean ratio of the number of arrivals and

4. the third moment of the number of arrivals in $(0, t_2)$

First all the above characteristics are determined for the superposition arrival process as follows. Consider the single voice source as a renewal process (single voice source - model 1 of 2.1). then the interarrival distribution is as given by (2). Taking the Laplace Stieltjes transform (LST) of (2) we have

$$\tilde{f}(s) = \int_0^\infty exp^{-st}\, dF(t) = [1 - \alpha T + \alpha T\beta/(s+\beta)]exp^{-sT} \tag{2.8}$$

Expected interarrival time of a single source $= -\tilde{f}'(0) = T + \alpha T/\beta$.
Equivalently, the mean packet arrival rate λ is given by

$$\lambda = 1/(T + \alpha T/\beta) \tag{2.9}$$

Now let $A(0, t)$ denote the number of arrivals of a stationary renewal process in the interval $(0, t)$ and let

$$M_r(t) = E[A^r(0, t)]$$

be the rth moment of arrivals in $(0, t)$ and let

$$M_r(s) = L[M_r(t)]$$

where L(.) denotes the Laplace transform. Using the results of the renewal process we have

$$M_1(s) = \lambda/s^2 \tag{2.10.1}$$

$$M_2(s) = \tfrac{\lambda}{s^2}\tfrac{1+\tilde{f}(s)}{1-\tilde{f}(s)} \tag{2.10.2}$$

$$M_3(s) = \tfrac{\lambda}{s^2}\tfrac{1+4\tilde{f}(s)+\tilde{f}^2(s)}{(1-\tilde{f}(s))^2} \tag{2.10.3}$$

But $M_1(t) = \lambda t$. Using (9) for λ gives

$$M_1(t) = t/(T + \alpha T/\beta) \tag{2.11.1}$$

The Index of dispersion for counts, I(t), satisfies

$$lim_{t\to\infty} I(t) = lim_{t\to\infty} \frac{Var[A(0,t)]}{M_1(t)} = \frac{Var(X)}{E^2(X)}$$

where X is the interarrival time. Therefore

$$lim_{t\to\infty}\frac{Var[A(0,t)]}{M_1(t)} = \frac{1-(1-\alpha T)^2}{(\alpha T+\beta T)^2} \tag{2.11.2}$$

The values of $M_2(t)$ and $M_3(t)$ can be obtained by numerical transform inversion of (2.10.2) and (2.10.3).

For the superposition process, the number of arrivals is given by

$$A^s(0,t) = \sum_{i=1}^{N} A_i^s(0,t)$$

For the superposition process, the number of arrivals is given by

$$A^s(0,t) = \sum_{i=1}^{N} A_i^s(0,t)$$

where $A_i(0,t)$ is the number of arrivals during the interval from source i.

Hence ,

$$M_1^s(t) = E[A^s(0,t)] = n\,M_1(t) \tag{2.12.1}$$

$$\frac{var[A^s(0,t)]}{E[A^s(0,t)]} = \frac{Var[A(0,t)]}{E[A(0,t)]} \tag{2.12.2}$$

The third central moment of the superposition process is given by

$$\begin{aligned}\mu_3^s(0,t) &= E\{[A^s(0,t) - E(A^s(0,t))]^3\} \\ &= n[M_3(t) - 3M_2(t)M_1(t) + 2M_1^3(t)] \qquad (2.12.3)\end{aligned}$$

Now for the MMPP, from [9] we have the probability generating function of the number of arrivals in an interval

$$g(z,t) = \pi \exp\{[\mathbf{R} + (z-1)\mathbf{\Lambda}]t\}\mathbf{e}$$

where

$$\pi = \frac{1}{r_1+r_2}(r_2, r_1) \text{ (equilibrium probability vector)}$$

$$\mathbf{e} = (1,1)^T$$

$$\mathbf{R} = \begin{bmatrix} -r_1 & r_1 \\ r_2 & -r_2 \end{bmatrix}$$

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

If A_t is the number of arrivals in the stationary 2 state MMPP over the interval $(0, t)$, then

$$\overline{A_t} = E[A_t] = \tfrac{\lambda_1 r_2 + \lambda_2 r_1}{r_1 + r_2} t \tag{2.14.1}$$

$$\tfrac{Var(A_t)}{\overline{A_t}} = 1 + \tfrac{2(\lambda_1 - \lambda_2)^2 r_1 r_2}{(r_1+r_2)^2(\lambda_1 r_2 + \lambda_2 r_1)} - \tfrac{2(\lambda_1 - \lambda_2)^2 r_1 r_2}{(r_1+r_2)^3(\lambda_1 r_2 + \lambda_2 r_1)} . (1 - \exp^{-(r_1+r_2)t}) \tag{2.14.2}$$

$$lim_{t \to \infty} \tfrac{Var(A_t)}{\overline{A_t}} = 1 + \tfrac{2(\lambda_1 - \lambda_2)^2 r_1 r_2}{(r_1+r_2)^2(\lambda_1 r_2 + \lambda_2 r_1)} \tag{2.14.3}$$

$$\text{3rd moment of number of arrivals in } (0, t_2) = g^{(3)}(1, t_2) \tag{2.14.4}$$

Equation sets of 2.12 and 2.14 are equated to determine the 4 unknowns $\lambda_1, \lambda_2, r_1$ and r_2. Once these are known, Matrix Geometric Techniques [20] can be used to solve the resulting MMPP/G/1 queue as dealt in detail in [9]. In [9] the model was used to evaluate the average delay of an infinite buffer, voice multiplexer with good accuracy. The method however did not work well for the finite buffer case.

In [10] two MMPP models were used to study the performance of fixed buffer multiplexers. The first model was used to evaluate the packet loss of moderate to large buffers while the second model was used for large buffer. Here, the arrival process is considered to consist of an underload and overload period. An overload state occurs when the number of sources in talkspurt exceeds the capacity of the system.

In the first model of moderate to large buffers, the variance in the arrival process during the overload states are considered, since the packet loss in such buffers are expected to occur in overload states only. The parameters that are matched are as follows

1. Value of $E[A_H(0,t)]/t$ at $t = 0$, with that of the superposition process, where $A_H(0,t)$ is the number of arrivals in time t for the MMPP given that the process started in the high arrival rate (H) at $t = 0$.

2. $E[A_H(0,t)]/t$ at $t = \infty$ with that of the superposition process.

3. The derivative of $E[A_H(0,t)]/t$ at $t = 0$ with that of the superposition process.

4. The value of the $Var[A_H(0,t)]$ at $t = t_m$, with that of the superposition arrival process. The value of t_m is chosen so that $Var[A_H(0,t)]$ match well over a period of one second, which is the average ON/OFF period of the voice source.

For the second model discussed in [10], the first 3 parameters matched are as given above and in addition $Var[A_L(0,t)]$ at $t = t_m$ is matched with that of the superposition arrival process. Simulation results of [10] suggest that this model performs better than [9] for finite buffer case.

Interrupted Poisson process

IPP is a special case of a 2 state MMPP, where one state is an ON state with associated positive Poisson rate, and the other state is an OFF state with associated rate zero. As discussed in an earlier section, such models have been used to characterize the packet arrival process from a single source. In a similar fashion the aggregated arrival process can be approximated by the superposition of IPPs, called the N-IPP. The N-IPP is an MMPP. If we denote the state of the N-IPP at time t as $J(t)$ where $J(t) = j$ is the number of IPPs in their ON state, then $J(t)$ is an $(N+1)$ state continuous time Markov chain (a birth and death process). The arrival in state j of the Markov chain is Poisson with rate $j\lambda$ while the birth and death rates are $(N-j)\gamma$ and $j\omega$ respectively. (where γ^{-1} and ω^{-1} are the mean ON time and OFF time of the model). [5] adopts this approach for modeling the superposition arrival process.

In [5], unlike the MMPP approach, the component process (arrival from a single voice source) is characterized rather than the superposition process. Hence, the packet arrival process from a single source is approximated as an IPP with 3 defining parameters, namely, the mean arrival rate in the ON state λ, the mean ON time γ^{-1} and the mean OFF time ω^{-1}. To determine these 3 defining parameters of the IPP from the statistical characteristics of the packet arrival process from a single voice source, 3 methods are considered in [5].

1) *The mean interval method:*The mean ON time γ^{-1}, OFF time ω^{-1} and the mean arrival interval during ON time $1/\lambda$ of the IPP are matched with the mean talkspurt α^{-1}, mean silence period β^{-1} and the packet arrival interval period T, during talkspurts.

2) *3 moments method:*The first 3 moments of interarrival time distributions of the IPP are matched with those of the packet arrival process. If m, c, k are the mean, the coefficient of variation and the third central moment of the packet interarrival time of the packet arrival process respectively, then from [5] we have

$$\lambda = \frac{2(k - 3c^2 + 1)}{(2k - 3c^4 - 1)m} \tag{2.15}$$

$$\omega = \frac{3(c^2 - 1)}{(k - 3c^2 + 1)m} \tag{2.16}$$

$$\gamma = \frac{9(c^2 - 1)^3}{(k - 3c^2 + 1)(2k - 3c^4 - 1)m} \tag{2.17}$$

3) *2 moments and peakedness method:* An important characteristics of the arrival stream is the peakedness. The exponential peakedness function $Z_{exp}(\mu)$ is defined as the variance to mean ratio of the number of busy servers in a fictitious infinite exponential server system with service rate μ, to which the arrival stream is hypothetically offered. For the packet arrival process from a single source, $z_{exp}(\mu)$ is given by (from[5])

$$Z_{exp}(\mu) = \left(1 - [1 - \alpha T + \frac{\alpha T \beta}{\mu + \beta}] \exp^{-\mu T}\right)^{-1} - \frac{\beta}{\mu(\alpha + \beta)T}$$

Hence, in this method as the name suggests, the first 2 moments of the interarrival time distributions and a peakedness are matched to yield ([5])

$$\lambda = \frac{1}{m} + \frac{(c^2 - 1)(z - 1)\mu}{c^2 + 1 - 2z} \tag{2.18}$$

$$\omega = \frac{2(z - 1)\mu}{(z - 1)(c^2 - 1)\mu m + c^2 + 1 - 2z} \tag{2.19}$$

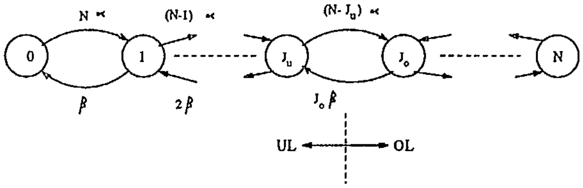


Figure 2.6: Phase process

$$\gamma = \frac{2\mu^2 m(c^2-1)(z-1)^2}{[\mu m(c^2-1)(z-1)+c^2+1-2z](c^2+1-2z)} \tag{2.20}$$

After having approximated the arrival process from a single source with the parameters $(\gamma, \omega, \lambda)$ determined by one of the above methods, the superposition can be analyzed as a N-IPP/G/1 queue as outlined in [5]. Simulation results in [5] show that the 2 moments and peakedness method is the most accurate of the 3 methods discussed.

Semi-Markov Process

In [11], [12], [13] and [16] the superposition arrival process is modeled as a Semi-Markov process or a two dimensional Markov chain.

As discussed before, each of the active sources feed packets to the multiplexer at the rate of V packets per second and these are removed by the multiplexer at the rate of VC packets per second. The number of packets arriving to the multiplexer depends on the number of sources in their active state. Hence, the number of active sources as a function of time $(J(t)$ can be modeled as a continuous time Markov chain as shown in Figure 2.6 ([11]). It is called the phase process in [11]. In [11] an approximate generating function of the probability density function of the queue length is computed by focusing on instants of completion of an overload/underload (OL/UL) cycle, which is defined as follows. Let J_0 be the smallest integer greater than C (the channel capacity) and let $J_u = J_0 - 1$. Then overload starts at the instant the number of active voice sources changes from J_u to J_0 (since, in such a condition more than C packets arrive in a frame of C transmission slots) and ends at the instant when the number of active voice sources change from J_o to J_u. Underload begins at this time and persists till overload starts again. The period between the start of successive overload is called the OL/UL cycle. The number of packets in the queue at the end of the n th OL/UL cycle is denoted by Q_n and the queue at the end of $n+1$th OL/UL cycle by Q_{n+1} (Figure 2.7 [11]). Then,

$$P\{Q_{n+1}|Q_n, Q_{n-1}, \ldots\} = P\{Q_{n+1}|Q_n\}$$

Therefore the sequence Q_n can be viewed as the states of a Semi-Markov chain whose state transition intervals correspond to the OL/UL cycle times, which are random variables. [11] discusses two methods - functional iteration and spectral factorization to determine the probability generating function of the probability density function of the queue length. However [11] does not evaluate the stochastic equilibrium distribution of the queue length.

[12] and [13] discuss a method to determine the stochastic equilibrium distribution of the multiplexer queue by approximating the superposition arrival process by a semi-Markov chain. The semi-Markov process approximated is as described below. Consider the phase process as shown in Figure 2.6. The following approximations are made

a) when $J(t) < C$ (corresponding to the underload state UL), the length of the queue (when it is non-empty) decreases at the rate of $V(C - J(t))$ packets per second. If the queue is empty it remains so as long as $J(t) < C$. No queue increment is allowed .

b) when $J(t) = C$ (this is possible only if C is an integer), the rate of change of the queue length is zero.

c) when $J(t) > C$ (corresponding to the overload state OL), the length of the queue increases at the rate of $V(J(t) - C)$ packets per second. No queue decrement is allowed.

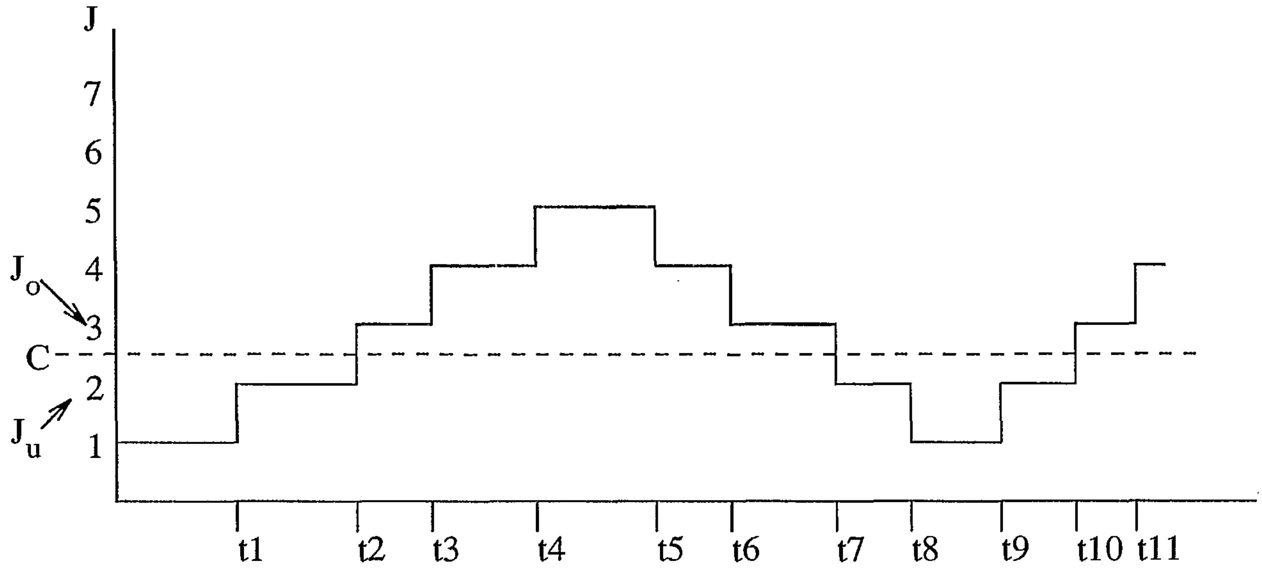
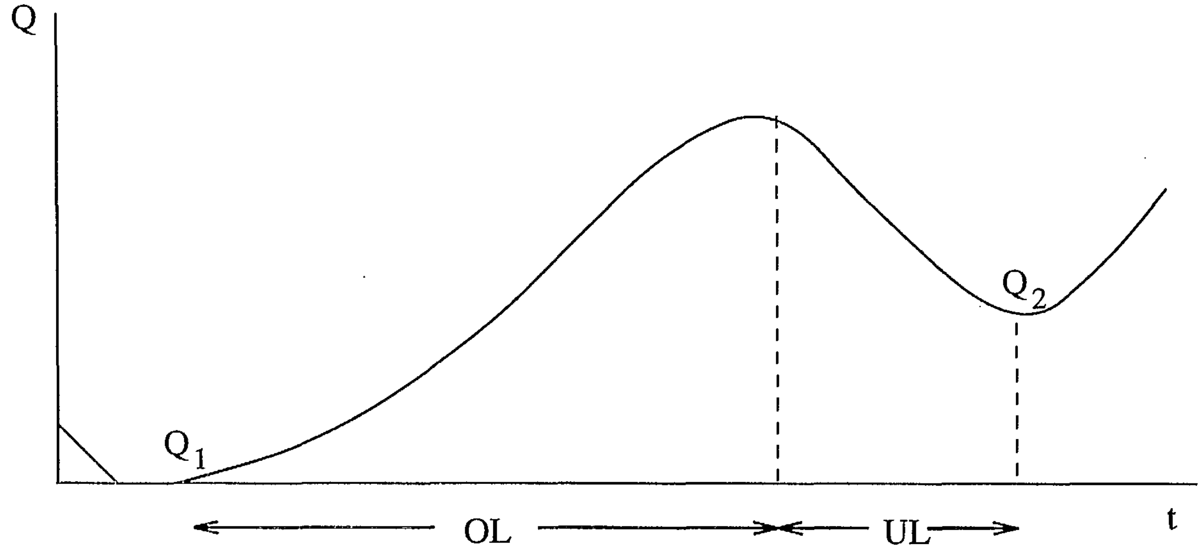


Figure2.6: No. of active voice sources as a function of time (J(t))



The multiplexer queue as a function of time

Figure 2.7: No. of active voice sources and multiplexer queue length as a function of time

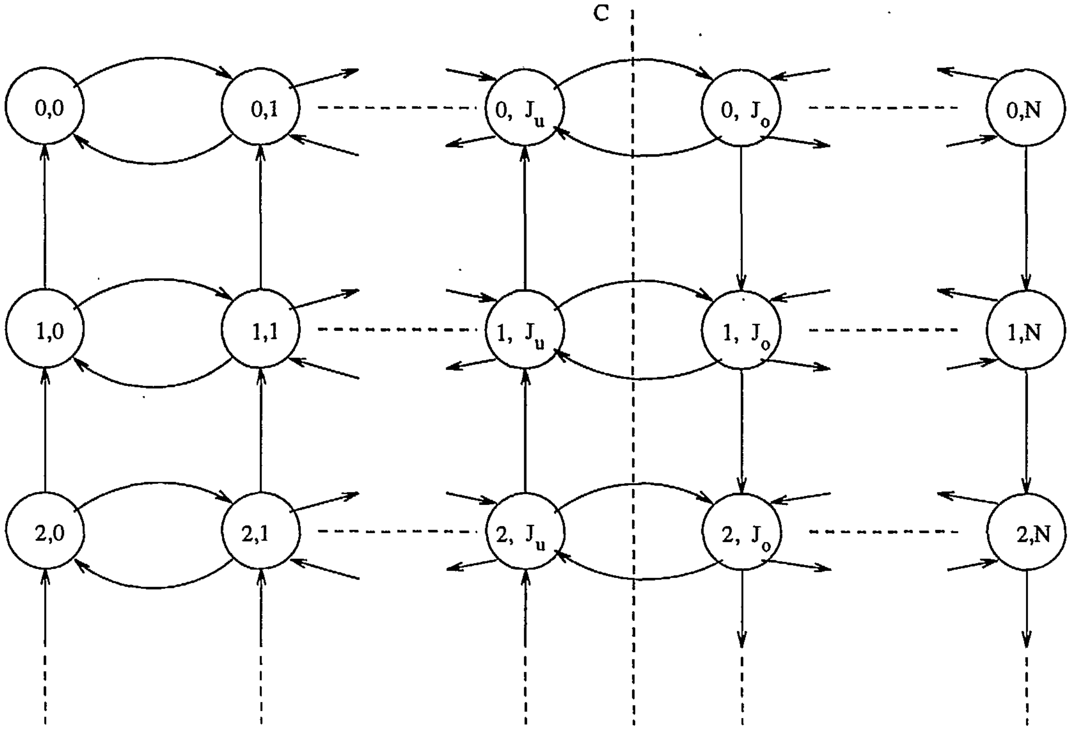


Figure 2.8: Semi-Markov process

Let the states of the process be denoted by (q_t, v_t) where $v_t = J(t)$, the number of sources in talkspurt at time t and q_t is the number of packets in the queue. Transitions from $9i, j)$ to $(i, j-1)$ or $(i, j+1)$ are called phase transitions, since the queue length does not change. The transitions from (i, j) to $(i+1, j)$ is a queue increment and to $(i-1, j)$ is a queue decrement. The process is shown in Figure 2.8. It can be observed that the transition probabilities for the process shown depend on the current state of the process. Hence there exists a Markov chain embedded at the instants of phase state changes, queue increments and queue decrements. Also the expected sojourn time in any state depends only upon the state. Therefore the process is a semi-Markov process. The parameters of this semi-Markov process are the packet generation rate ,the mean talkspurt and silence periods, the communication link capacity and the total number of voice sources.

To compute the equilibrium probability $p_{i,j}$ that $q_t = i$ and $v_t = j$ the following

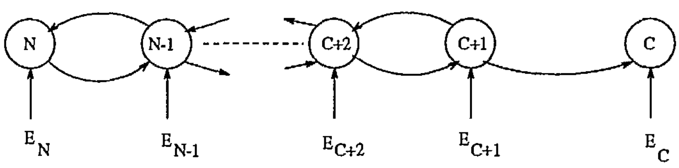

N
N-1
C+2
C+1
C
E_N
E_{N-1}
E_{C+2}
E_{C+1}
E_C
E_i = the probability that a blocking period starts in state (i,K)

Figure 2.9: Blocking state diagram

equation from renewal theory is used in [12] and [13]

$$p_{i,j} = \frac{q_{ij}\, m_{ij}}{\sum_{k=0}^{\infty}\sum_{l=0}^{N} q_{kl} m_{kl}} \tag{2.21}$$

where

q_{ij} = equilibrium probability for the embedded Markov chain

m_{ij} = expected sojourn time in state (i, j)

Matrix Geometric method is used to solve equation 2.21 in [12] and [13]. Comparison of the results obtained by this approach with the simulation shows that the model overestimates the probability that the queue is empty. This is due to the approximations underlying the mode.

[16] also approximates the superposition as a semi-Markov chain. However, the system considered is a finite buffer one and the emphasis is placed on the placed on the packet loss which is incurred only when the buffer is full. If K is the total buffer capacity in packets and $\pi_{i,K}$, the equilibrium probability of i voice calls in talkspurt when the buffer is fullthen we have $\pi_{i,K} = 0$ for $i \leq C$, since the buffer will not be full when the service rate is greater than the arrival rate. Hence the packets would be lost only for states greater than C. [16] considers these states alone and calls it the blocking states (Figure 2.9 [16]). Focusing on the blocking states analytical expressions are derived in [16] for the temporal behaviour of packet loss. Results show that the packet loss rate changes slowly and has large fluctuations. Increasing the buffer size merely extends the non-blocking periods and thereby reduces the overall aaverage packet loss rate. However, once a blocking period occurs, the length of the period as well as the packet loss within this period becomes irrelevant to the buffer

size.

Discrete time Markov model

Here again the state of the process is the tuple consisting of the number of sources in talkspurt and the number of packets in the queue. But the sampling is done after every frame and hence the process is in discrete time domain. The state of the system at the beginning of the nth frame is given by (t_n, b_n) where t_n is the number of users in talkspurt and b_n is the queue length.

Such a system is studied in [15] for a finite buffered voice multiplexer. Two schemes for discarding the packets are considered. In the first scheme a buffer of size K is properly selected so that all the packets within the buffer can be transmitted within their time (delay) constraint. All the packets arriving after the buffer is full are discarded. In the second scheme, all the arriving packets are stored in the buffer and at the end of a frame, the system randomly selects a packet to drop from the arrivals in the frame. This process is repeated until the all the remaining packets meet their delay constraint. This scheme balances the packet loss for each user.

For both the schemes the transition probability,

$$p_{ij,kl} = Pr\{t_{n+1} = k, b_{n+1} = l | t_n = i, b_n = j\} \qquad 0 \leq i, \quad k \leq N \quad 0 \leq j \quad l \leq K$$

and the equilibrium state probability

$$\pi_{nm} = Pr\{t = n,\, b = m\} \qquad 0 \leq n \leq N, \quad ;\, 0 \leq m \leq K$$

in both the schemes are determined by considering the queue length transitions from b_n to b_{n+1} for the following four cases

- *case 1:* $b_n \geq C$ and $t_n \leq K - b_n + 1$; enough packets in the queue to keep the server busy and not too many arrivals to cause overflow.

- *case 2:* $b_n < C$ and $t_n \leq K - b_n + 1$; not enough packets in the system to keep the server busy and not too many arrivals to cause overflow.

- *case 3:* $b_n \geq C$ and $t_n > K - b_n + 1$; the server keeps busy a nd overflow may occur. Due to overflow some packets will be discarded. Let the number of packets discarded $D_1 = d$. Then D_1 is a random variable with probability density function $\Psi_{D_1}(d)$.

- *case 4:* $b_n < C$ and $t_n > K - b_n + 1$; the server may go idle and overflow may occur. Let D_1 be the number of packets discarded and R the number of packets served in the frame (then the server is free for $C - R$ timeslots during the frame). Then R and D_1 are random variables with a joint probability density function $\Theta_{R,D_1}(r, d)$.

[[15] discusses the computation of the pdfs $\Psi_{D_1}(d)$ and $\Theta_{R,D_1}(r, d)$ for both the schemes. Results show that scheme 2 performs better than scheme 1 as it spreads the packet loss across the users.

Uniform Arrival and Service model

The Uniform Arrival and Service (UAS) model, which assumes that the information flow in and out of the buffer is uniform rather than in discrete packets was used by [23] for modeling data traffic. In the UAS model the source generates information to the transmitter at a rate of one unit of information per unit time and the server removes information from the buffer at a uniform rate not to exceed C units of information per unit of time. As in the semi-Markov process of [12], while the system is in state $J(t) = j > C$, the buffer content increases at the rate of $j - C$ units of information per unit of time (if the queue reaches its limits it will stay on its limit) and when the system is in state $J(t) = j < C$, the buffer content reduces at the rate of $C - j$ units of information per unit of time as long as the buffer is nonempty(if the buffer becomes empty it will stay empty)[14],[12] and [13] approximate the superposition arrival process by this model.

In [14] the UAS model is used to model a finite buffer multiplexer. The equilibrium distribution is described by a set of differential equations, which together with a set of boundary conditions can be solved to yield the equilibrium distribution of delay and packet loss. The method is briefly outlined below.

If $P_i(t, b)$ be the probability that at time t there are b packets in the queue and i lines are in their talkspurt, where $0 \leq i \leq N, t \geq 0$ and $0 \leq b \leq K$. If δt be a small time interval, then from Figure 2.6 we have

$$\begin{aligned} P_i(t+\delta t, b) &= P_{i-1}\{t, b-(i-C)\delta t\}p(i-1,i)\delta t \\ &\quad +P_{i+1}\{t, b-(i-C)\delta t\}p(i+1,i)\delta t \\ &\quad +P_i\{t, b-(i-C)\delta t\}(1-p^*(i)\delta t)+O(\delta t) \qquad (2.22) \end{aligned}$$

where

$$\begin{aligned} &p^*(i) = p(i,i+1)+p(i,i-1) \\ &p(i,i+1) = (N-i)\beta \qquad i \neq N \\ &p(i,i-1) = i\alpha \qquad i \neq 0 \end{aligned}$$

Dividing equation 2.22 by δt and letting $\delta t \rightarrow 0$ we get

$$\begin{aligned} \frac{\partial P_i(t,b)}{\partial t}+(i-C)\frac{\partial P_i(t,b)}{\partial b} &= p(i-1,i)P_{i-1}(t,b) \\ &\quad +p(i+1,i)P_{i+1}(t,b) \\ &\quad -p^*(i)P_i(t,b) \qquad 0 < b < K \qquad (2.23) \end{aligned}$$

To find time independent equilibrium probability $lim_{t\rightarrow\infty}P_i(t,b)$ define $F_i(b) = lim_{t\rightarrow\infty}P_i(t,b)$, then equation 2.23 becomes

$$(i-C)\frac{dF_i}{db} = p(i-1,i)F_{i-1}(b)+p(i+1,i)F_{i+1}(b)-p^*(i)F_i(b) \; ; \qquad 0 < b < K \qquad (2.24)$$

Equation 2.24 can be written in matrix form as

$$\mathbf{D}dF(b)/db = \mathbf{M}F(b) \qquad 0 < b < K \qquad (2.25)$$

where

$$\mathbf{D} = diag\{-C, 1-C, 2-C, \cdots\cdots N-C\}$$

$$\mathbf{M} = \begin{bmatrix} -p^*(0) & p(1,0) & & & & & \\ p(0,1) & -p^*(1) & p(2,1) & & & & \\ & p(1,2) & -p^*(2) & p(3,2) & & & \\ & & \cdot & & & & \\ & & & \cdot & & & \\ & & & & \cdot & & \\ & & & & \cdot & & \\ & & & & p(N-2,N-1) & -p^*(N-1) & p(N,N-1) \\ & & & & & p(N-1,N) & -p*(N) \end{bmatrix}$$

The solution to the differential equation 2.25 is

$$F(b) = \sum_{k=0}^{N} \exp(z_k b) a_k \phi_k \qquad 0 < b < m \qquad (2.26)$$

$$z_k = \text{eigen value of } D^{-1}M$$

$$\phi_k = \text{right eigen vector of } D^{-1}M$$

The a_k are coefficients got by solving boundary conditions. For an infinite buffer case, closed form expressions exist for z_k, ϕ_k and a_k, as given in [23]. In the case of finite buffers [14] discusses a method of formulating the boundary equations to solve for ϕ_k, z_k and a_k.

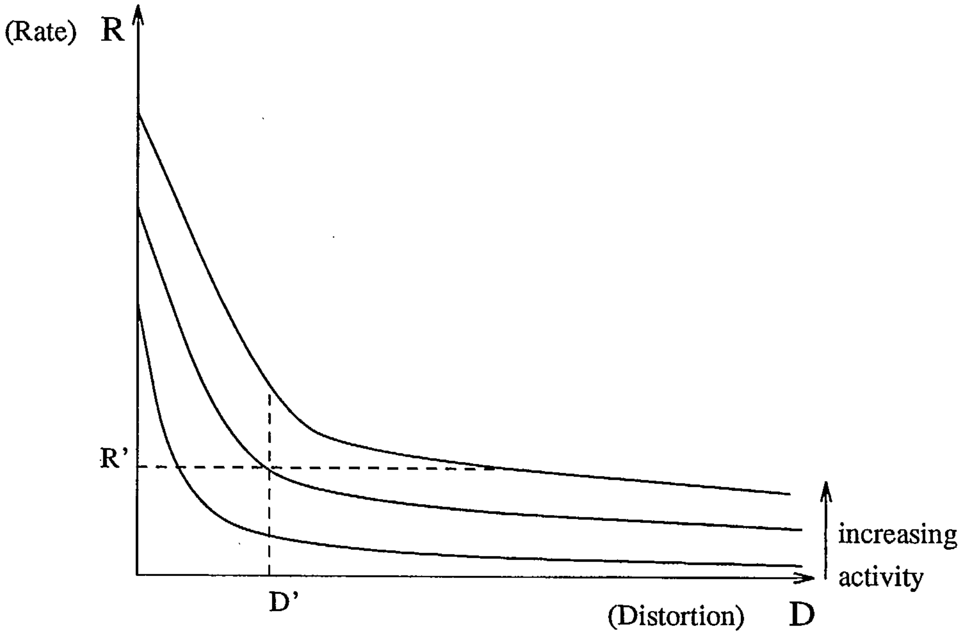


Figure 2.10: Rate Distortion Curves

2.2 Survey of video traffic models

The introduction of BISDN/ATM technologies to broadband networks and the advancements in source coding algorithms for video, have made feasible the use of variable bit rate coding for video transmission. This would engender a flexible communication network with a high efficiency, as network resources can be shared dynamically by numerous users.

A VBR video codec produces a variable bit rate output by adapting the generated bit rate to the the local and temporal image complexity, while maintaining a constant image quality. This can be observed from the rate distortion curves shown in Figure 2.10. These curves depict the variation in the output bit rate as a function of distortion in the output. From the figure it is evident that in order to maintain a low distortion (or high quality) in the output, a higher bit rate codec is required, however a lower bit rate codec produces high distortion in the output. While a constant bit rate coder, produces a constant bit rate output at the expense of quality, a VBR codec maintains a constant quality by varying the bit rate.

The advantages of employing VBR video codecs are many. First of all at low bit rates, use of constant bit rate video codecs, produces a highly varying picture quality which is particularly annoying to the viewer. Use of VBR video codecs helps maintain a constant quality. Secondly, at high bit rates use of VBR video yields high bandwidth gains by using channel sharing among multiple users. In certain cases VBR coding also alleviates the need of sophisticated coding algorithms, as the same effects in picture quality could be achieved by using higher bit rates.

VBR video sources are highly bursty. The burstiness of VBR video sources is a subjective measure [21] that depends on the content of the video (e.g., picturephone, teleconference, broadcast television etc.,) and the encoding scheme used (DCT, Motion compensated DCT, DPCM, MPEG etc.). As the video signals are expected to occupy most of the bandwidth in the future broadband networks, accurate modeling of a VBR video source based on its statistical characteristics is required for the design of such networks. Numerous works in the literature have focussed on modeling VBR video sources. This section surveys the various models that have been proposed in the literature to characterize VBR video sources.

2.2.1 Characteristics of VBR video

The charcteristics of VBR video depend on the information content of the picture and the encoding algorithm used. The bit rate of the coded video is dependent on the motion activity in the scene, namely low, medium and high motion. Due to the continuity of motion within a scene only small portion of the picture changes from frame to frame. Hence variations in bit rate are smaller within a scene. The bit rate of the coder also depends on the changes in the content of the video (like cuts, scene changes, etc.). Highest bit rates arise during scene changes and last only one or two frames depending on the coding algorithm. However, the data rate output of a VBR video encoder does not actually reflect the changes in the information content of original video signal, since the compression of the bit rate achieved by various algorithms are different.

The data buffering scheme used by the encoder also influences the bit rate variations of the encoder. For example in an encoder that uses frame buffering, all the variations arising from the locality of an image within a frame are smoothed, whereas in a multi-frame buffered codec variations in bit rate between frames are also smoothed.

There is a strong correlation among the bit rates of successive frames due to the nature of actual video scenes and interframe coding. Correlations that occur because data on part of an image is highly correlated with data on the same part on the next line are called spatial correlations. Correlations that occur because data on one part of an image is highly correlated with data on the same part of the next image are called temporal correlations. Spatial and temporal correlations together with the encoding scheme greatly influence the bit rate output of VBR video codec.

Table 2.2 adapted from [21] summarizes the bit rate variations that occur in a VBR video codec and the corresponding time scale they occur. Hence modeling a VBR video source is a difficult and complex task as the bit rate process posseses a high degree of variability at different levels. Thus, modeling of a VBR video source may be done at one of 3 levels, namely at a scene level, frame level or intraframe level, as depicted in Figure 2.11.

Various models have been proposed in the literature to characterize VBR sources with scene changes and without scene changes (at a frame level). However the characteristics of VBR video at the intraframe level have not been well understood. The following sections give a brief overview on the various models proposed to characterize interscene and intrascene variations.

Type	Time scale	Causes	Characteristics
Long term variability (multiple scenes)	Several seconds	Scene changes	Discontinuous variation,differing statistical characteristics before and after the change
Short term variability (intrascene)	Between 1 frame period and several seconds.	Subject motion, camera motion, pattern variation.	Smooth variations with temporal correlations,with occasional large variations due to subject and camera motion.
Intraframe variability	Less than 1 frame period.	Spatial variation of the characteristics within an image.	Variations that have a periodicity due to image scanning or block processing.

Table 2.2: Classification of bit rate variations

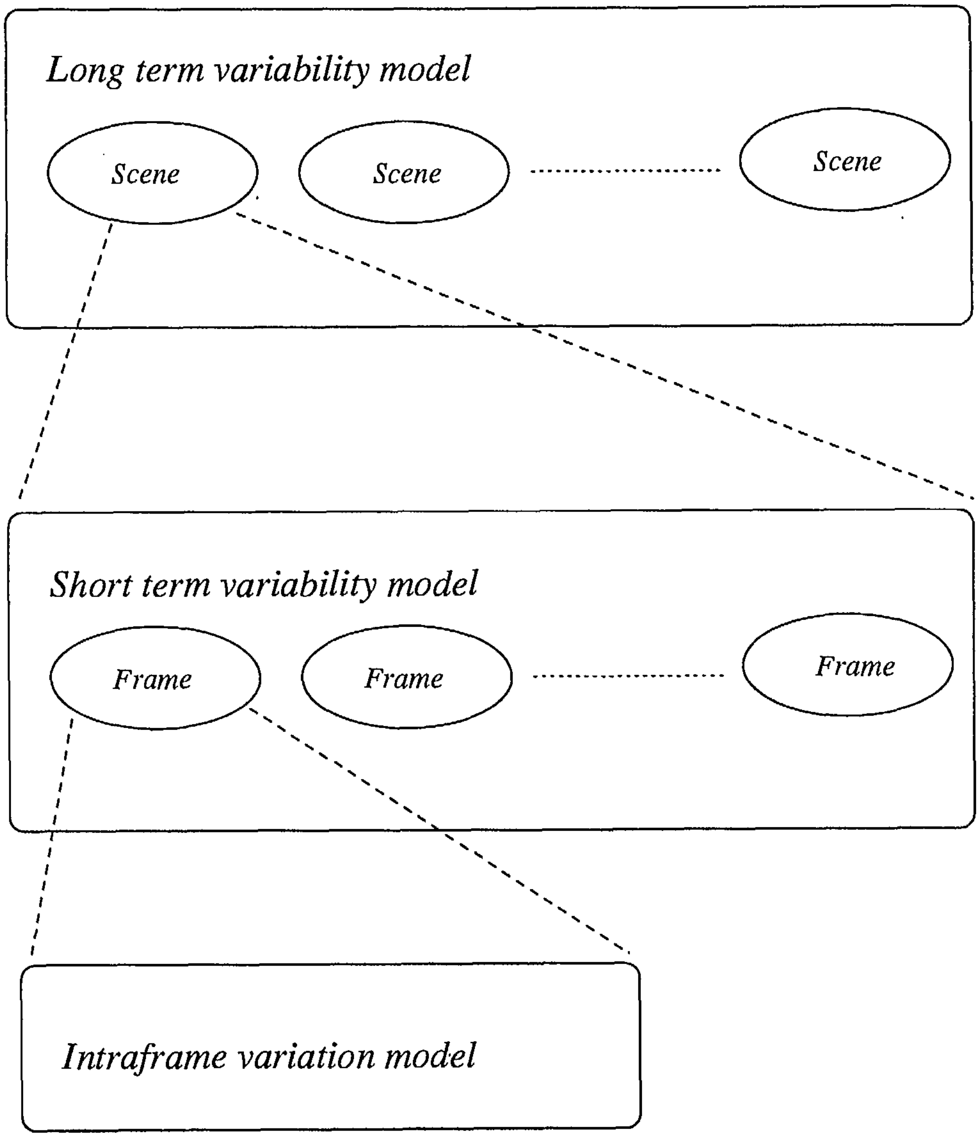

Long term variability model
Scene
Scene
Scene
Short term variability model
Frame
Frame
Frame
Intraframe variation model

Figure 2.11: Modeling of VBR video

2.2.2 Models of Intrascene variations (i.e. without scene changes)

These models are applicable to video scenes with relatively uniform activity levels, with few scene changes like video conference scenes showing a person talking. Under these circumstances the variations in bit rate is small and the bit rate process possesses short term corelations only. Infact the study of such bit rate processes have shown that they possess bell shaped nearly normal distributions [21], [22], [23], [24], [25],[26]. The autocorrelation function of the bit rate process closely resembles a negative exponential (for a frame buffered codec). Based on these a few models have been suggested to characterize intrascene variations.

Autoregressive process model

An autoregressive process model of order M (denoted $AR(M)$) is one which predicts the future values of a time series by regressing on the past M sets of values. Such process models have exponentially decaying autocorrelation and a Gaussian distribution. Based on this, AR process was suggested as a model for VBR video in [23], [27], [25], [24].

An autoregressive process model for VBR video is defined as

$$\lambda(n) = \sum_{m=1}^{M} a_m \lambda(n-m) + be(n) \tag{2.27}$$

where $\lambda(n)$ represents the source bit rate during the nth frame,M is the order of the model, $e(n)$ is a Gaussian random process (with mean η and variance 1). $a_m (m = 1, 2, \ldots M)$ and b are constants. For $M = 1$ we have the first order AR process given by

$$\lambda(n) = a\lambda(n-1) + be(n)$$

(Since the value of the sequence depends only on its previous instant it is called a continuous state Auto Regressive Markov model). The parameters of this model are a,b and the mean value η of $e(n)$.

The mean and autocovariance of the AR process are given by

$$E(\lambda) = \frac{b\eta}{1-a} \tag{2.28}$$

$$C(n) = \frac{b^2}{1-a^2}a^n \qquad n \geq 0 \tag{2.29}$$

Hence the parameters a,b and η are obtained by matching the equations (2.28) and (2.29) with empirical data.

Due to its simplicity and accuracy the AR(1) process is an excellent candidate for modeling VBR video sources. But it does not lend itself to a queueing analysis easily. Hence, this model has its utility limited to simulations.

Another important utility of the AR process is the fact that it can be used to statistically characterize a multiplex of video sources [21]. If $\Lambda(n)$ denotes the signal which results from multiplexing N, AR processes, $\lambda_1(n), \lambda_2(n), \lambda_3(n), \ldots \lambda_N(n)$, we have

$$\Lambda(n) = \sum_{i=1}^{N} \lambda_i(n)$$

If $\lambda_i(n)$ are mutually independent then the mean and variance of the resulting multiplex are given by

$$E[\Lambda(n)] = \sum_{i=1}^{N} E[\lambda_i(n)]$$

$$E[\Lambda(n)\Lambda(n+S)] = \sum_{i=1}^{n} E[X_i(n)X_i(n+S)]$$

Hence, if $\lambda_i(n)$ are identical AR processes, the resulting multiplex $\Lambda(n)$ is also an AR process with parameters a and b same as the original AR processes and whose mean and variance are N times those of $\lambda_i(n)$.

Though first order AR processes AR(1) were found to be reasonably accurate in modeling VBR video sources, a better matching may be achieved if the order of regression is increased. In this case the autocovariance of the resultant process is a sum of several exponentials. [24] proposes an alternative solution to achieve the same effect. Here the bit rate per frame $\lambda(n)$ is modeled as a sum of N, AR(1) processes $\beta_i(n)$. i.e.,

$$\lambda(n) = \sum_{i=1}^{N} \beta_i(n)$$

where

$$\beta_i(n) = a_i x_i(n-1) + b_i e_i(n)$$

$e_i(n)$ are Gaussian random processes with mean μ_i and unit variance. It is shown that a choice of N=2 provides a fair accuracy/complexity tradeoff. The method of determining the parameters of the process is discussed in [24]

Markov models

The Markov models have the memoryless property and lend themselves quite well to an analytical treatment. Due to this reason they have an edge over the AR process model described before. Two types of markov models have been proposed:
(a) Continuous time, discrete state, Markov models.
(b) Discrete time, discrete state, Markov models.

(a) Continuous time, discrete state, Markov models

The bit rate process $\lambda(t)$ from a video source is modeled as a continuous time, discrete state, Markov model in [23],[27]. The spectrum of possible values of bit rates from the video source is quantized into M discrete levels (where state M corresponds to peak bit rate level) of stepsize A and these $M+1$ levels (including 0) correspond to the state space of the Markov process. Now, the continuous process $\lambda(t)$,[2] describing the bit rate of the video source at time t is sampled at random points in the time domain, and is quantized into the nearest level $\lambda'(t)$ (Figure 2.12). Hence the pro-

[2] since the bit rate is of the order of several Mbps and the packet length is small, this model assumes the data as a continuous bit stream, ignoring the effects of packetization.

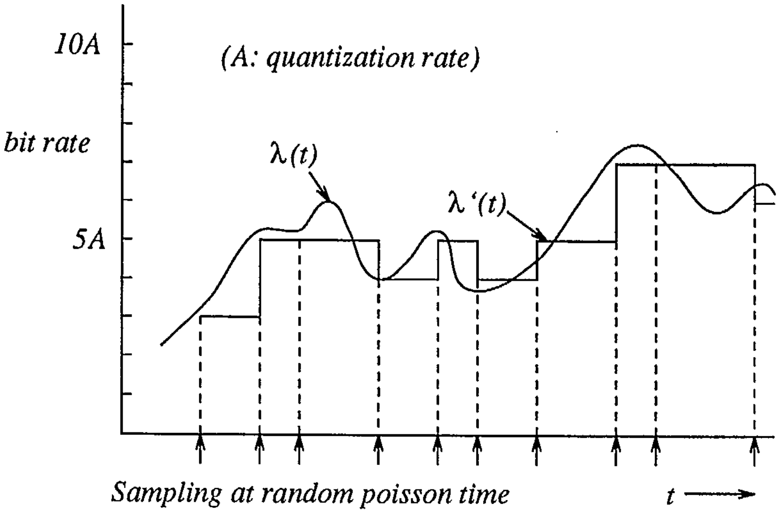


Figure 2.12: Poisson sampling and quantization of the source rate

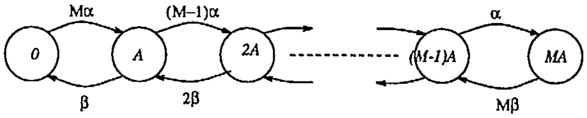


Figure 2.13: State transition rate diagram of Discrete time, discrete space,Markov process

cess can be seen as switching between different states (as determined by the value of $\lambda'(t)$), spending exponentially distributed time periods in each state (due to the poisson sampling). Since the process being modeled is of uniform activity, only state transitions to nearest neighbour states are allowed. The approximation of $\lambda(t)$ by $\lambda'(t)$ can be improved by decreasing the quantization step size A (and thus increasing M) and increasing the sample rate.

This model can be used to model both a single video source or a multiplex of N video sources. In the latter case the state space is formed by quantizing the aggregate rate of the multiplex $\lambda_N(t)$ into M discrete levels. As before state changes between nearest neighbours are only allowed. Hence it results in a birth death process whose state transition rate diagram is shown in Figure 2.13. The exponential transition rates between states iA and jA are given by

$$\gamma_{i,i+1} = (M-i)\alpha \qquad i < M \tag{2.30.1}$$

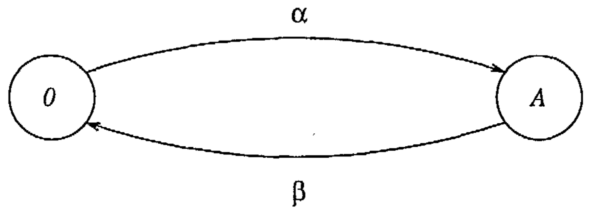


Figure 2.14: Minisource model

$$\gamma_{i,i-1} = i\beta \qquad\qquad i > 0 \tag{2.30.2}$$

$$\gamma_{i,i} = 0 \tag{2.30.3}$$

$$\gamma_{i,j} = 0 \qquad\qquad |i-j| > 1 \tag{2.30.4}$$

The birth death process of Figure 2.13 can be considered to represent a population of 'minisources', where each minisource is as given in Figure 2.14, i.e., each minisource is in one of the states ON or OFF. When ON it generates information at the rate of A bits/sec. Then the probability the system is in state kA is same as the probability that there are k minisources out of M minisources in their ON state. It can be shown that $\lambda'_N(t)$ has a binomial distribution

$$P\{\lambda'_N(t) = kA\} = \begin{pmatrix} M \\ k \end{pmatrix} p^k(1-p)^{M-k} \tag{2.31}$$

where

$$p = \frac{\alpha}{\alpha+\beta}$$

$$E(\lambda'_N) \;=\; M\,A\,p \tag{2.32}$$

$$C'_N(0) \;=\; M\,A^2\,p(1-p) \tag{2.33}$$

$$C'_N(\tau) \;=\; C'_N(0)\exp -(\alpha+\beta)\tau \tag{2.34}$$

Here, the parameters of the continuous time, discrete state Markov model namely α, β and A can be determined by matching the mean $E(\lambda'_N)$, variance $C'_N(0)$ and

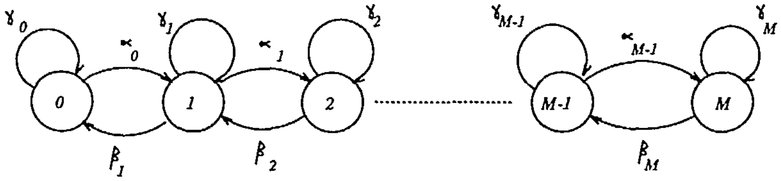


Figure 2.15: Discrete time, discrete state, Markov process model

exponential autocovariance $C'_N(\tau)$ as given by equations (2.32) to (2.34) with the corresponding measured values. The number of min isources required for a good approximation of a multiplex of N video sources was found experimentally to be $20N$ [23], [27].

As already mentioned Markov models lead to tractable analytical treatment. In [23], [27] a fluid flow analysis has been carried through to arrive at the survivor function of buffer occupancy.

(b) Discrete time, discrete space, Markov model

This model is used to characterize a multiplex of N video sources in [28]. In this model the total range of bit rates of the multiplex are quantized into M discrete levels. As before these levels form the state space of the Markov chain. However the time is discrete, corresponding to a duration of a frame. Since it is discrete, each state of the Markov chain has three possible transitions: increase, decrease or remain at the same level, as shown in Figure 2.15. There are 3 parameters $(\alpha_i, \beta_i, \gamma_i)$ associated with each state i, $i = 0, 1, 2, \ldots M$ where α_i is the transitionprobability of moving forward one state, β_i is the transition probability of going from state i to state $i-1$ and γ_i is the probability of staying in the same state i. As before if $\lambda(n)$ represents the bitrate at the nth frame, then

$$\lambda(n+1) = \lambda(n) + W_{\lambda(n)} \tag{2.35}$$

where

$W_{\lambda(n)}$ is a discrete random variable and

$$W_\lambda(n) = \begin{cases} 1 & \text{with probability } \alpha_\lambda(n) \\ -1 & \text{with probability } \beta_\lambda(n) \\ 0 & \text{with probability } \gamma_\lambda(n) \end{cases}$$

And the parameters at each state i satisfy

$$\alpha_i + \beta_i + \gamma i = 1 \qquad i = 0, 1, 2, \ldots M \tag{2.36}$$

The parameters α_i, β_i can be selected suitably. In the selection of α_i, β_i, it is desirable to have the property that the sequence $\{\alpha_i\}$ is a decreasing sequence and the sequence $\{\beta_i\}$ is an increasing sequence. This assures that when the bit rate is below the average there is a tendency to increase and when it is above theaverage, there is a tendency to decrease. With appropriate selection of parameters, the steady state probability of being at state i, $P(i)$ can be determined.

Autoregressive moving average models

In [29] an Autoregressive moving average (ARMA) model has been proposed for characterizing the output of a non-frame buffered video codec. The ARMA models have autocovariances that exhibit recorrelation. Since the output bit rate from a non-frame buffered video codec also exhibits recorrelation (temporal and spatial), ARMA models serve as a better choice to model the output bit rate process from a non-frame buffered video codec. The ARMA model was used to represent the cell arrival in intervals of typically 100μs. The number of cells in the *ith* interval is modeled by a discrete state, autoregressive moving average process, X_i given by

$$X_i = g(\alpha Z_{i-m} + Y_i + v_i) \qquad \text{with } \alpha < 1 \tag{2.37}$$

where Y_i and Z_i are a sequence of correlated Gaussian random variables with zero mean (since a white noise sequence σ_i, with zero mean is applied at the filter's input). The moving average part, i,e., the sequence Y_i models frame correlations and the autoregressive part, Z_i, models scene and frame correlations. The sequence of uncorrelated Gaussian random variables v_i, with zero mean, models the white noise stochastic component. $g(.)$ is a Zero memory Non linear (ZMNL) operator which converts the output of the ARMA filter into strictly positive random variables. The method of parameter estimation of the ARMA process is described in detail in [29].

TES models

TES (Transform expand sample) [30, 31, 32] is a non-parametric method which can accurately capture the histogram and approximate autocorrelation function of any data set. TES methodology assumes that some stationary empirical time series (such as traffic measurements over time) is available and then it tries to construct a model such that the marginal distribution (or histogram), leading autocorrelation and sample path realizations (histories) matches with the empirical values quite well.

TES processes come in 2 flavours: TES^+ and TES^- process (i.e with positive and negative lag - 1 autocorrelations respectively). TES^+ gives rise to the sequence $\{U_n^+\}$ given by

$$U_n^+ = \begin{cases} U_0 & \text{if} \quad n = 0 \\ < U_{n-1}^+ > & \text{if} \quad n > 0 \end{cases} \tag{2.38}$$

while TES^- gives rise to the sequence $\{U_n^-\}$,

$$U_n^- = \begin{cases} U_n^+ & n \quad \text{even} \\ 1 - U_n^+ & n \quad \text{odd} \end{cases} \tag{2.39}$$

Here, U_0 is distributed uniformly on [0,1); $\{V_n\}$ is a sequence of IID random variables, independent of U_0, called the *innovation sequence* and angular brackets denote modulo - 1 (fractional part) operator $< x > = x - \max\{\text{ integer } n : n \leq x\}$. The sequences $\{U_n^+\}$ and $\{U_n^-\}$ of the form (13) and (14) are called *background sequences* and give rise to a sequence of stationary random variables with uniform marginals on [0,1) and different autocorrelation structures. For practical purposes, transformed

TES processes $\{X_n^+\}$ and $\{X_n^-\}$, obtained from (13) and (14) by some transformation D (called a *distortion*) are of importance. i.e.,

$$X_n^+ = D(U_n^+); \qquad X_n^- = D(U_n^-) \tag{2.40}$$

The sequences $\{X_n^+\}$ and $\{X_n^-\}$ are called *foreground sequences* . The idea is to create suitable foreground sequences with marginal distributions matching the given (empirical) distribution, by using the inversion method [33]. For a given distribution function F, the inversion method uses distortion $D = F^{-1}$ to genarate stationary sequences $\{X_n^+\}$ and $\{X_n-\}$ with marginal distribution F. In the empirial TES methodology, the distortion is effected in two stages. First, in order to "smooth" TES sample paths, a family of transformations called stitching transformations S_ξ, $0 < \xi < 1$ is employed.

$$S_\xi(y) = \begin{cases} \frac{y}{\xi}, & \text{if} 0 \leq y < \xi \\ \frac{1-y}{1-\xi}, & \text{if} \xi \leq < 1 \end{cases} \tag{2.41}$$

Processes of the form $\{S_\xi(U_n^+)\}$ and $\{S_\xi(U_n^-)\}$ are called *stitched TES processes.* For $0 < \xi < 1$ the effect of S_ξ is to render the sample paths of background TES sequences more "continuous-looking". In the second stage the inversion method is applied to the stitched processes to generate the foreground sequences with matched distributions as the given distribution F. Thus the distortion is given by

$$D = F^{-1}(S_\xi(U_n^-)) \quad \text{or} \quad F^{-1}(S_\xi(U_n^+))$$

However TES methodology models empirical densities as histograms, as is explained in [31].

TES methodology also fits the autocorrelation of the empirical data with that of the model. This is carried out by a heuristic search for a pair (ξ, f_v), (where ξ is a stitching parameter and f_v is an innovation density) such that the autocorrelation function approximates its empirical counterpart. The search can efficiently be carried out using the visual, interactive software environment called *TEStool* [34].

GOB (group of block) level source model, for compressed H.261 standard VBR video over a local area network was constructed in [31, 30]. The GOB is a suitable unit of packet transport. (Each DCT coded frame is divided into 12 group-of-block coded subscreens). At the GOB level the bit rate process is characterized by an autocorrelation that is periodic both at the spatial GOB scan rate and at temporal frame rate. In order to fit a TES model to this data, the raw data is first transformed into a new sequence called the *residual sequence* $\{R_n\}$ which has a faster decaying autocorrelation function. This transformed sequence could effectively and easily be fitted with a TES model as explained in [31, 30].

TES models can be used to generate synthetic streams of realistic traffic to drive simulations of communication networks. However they suffer from the handicap of not leading to tractable mathematical analysis.

2.2.3 Modeling scene changes

These models are useful in describing video sources with high motion and scene changes as in broadcast applications. Models that are proposed for video sources with scene changes must capture both short term and long term correlations. In this section we examine a few models that have been proposed to characterize video sources with scene changes.

Continuous time, discrete state, Markov process model

This model proposed in [35] is an extension of the model by [23] (discussed in section 3.2). Here, as in [23] the source changes between various fixed rate levels, with exponentially distributed times in each level. However, here, the possible data rate levels are built from a linear combination of two basic rates, a higher rate A_h and a lower rate A_l. This model can represent the bit rate from a single video source or an aggregate of N video sources. The state transition rate diagram for an aggregate of

N video sources using this model is shown in Figure 2.16. (The labels in each state indicate the data rate in that state). The basic rate A_l corresponds to parameter A in the model [23]. Transitions based on A_l model the short term correlations, while transitions based on A_h model the long term correlations. Hence with no transitions based on A_h, this model reduces to the model of intrascene variations as in [23]. For an aggregate of N video sources, there are $NM + 1$ low rate levels and $N + 1$ high rate levels, where M is chosen arbitrarily. The parameters of the model are determined by matching the theoretical values with measured values. For $N = 1$, the parameters c and d are determined by matching the fraction of time spent in high activity level $q(= c/(c+d))$ and the average time spent in high activity level $\frac{1}{d}$ with the actual measured data. For determining a, b, A_l and A_h, the autocovariance, variance, mean ratio (γ)(ratio of average bit rate in high level tot that of low level) and the overall mean bit rate $\overline{\lambda}$ as given by equations 2.43 to 2.45 are matched with the actual measured values.

$$C(\tau) = C(0) \exp -(a+b)\tau \qquad (2.42)$$

$$C(0) = Np(1-p)A_l^2 \qquad \text{where } p = \tfrac{a}{a+b} \qquad (2.43)$$

$$\gamma = \frac{NpA_l + A_h}{NpA_l} \qquad (2.44)$$

$$\overline{\lambda} = NpA_l + qA_h \qquad \text{where } q = \tfrac{c}{c+d} \qquad (2.45)$$

For the sake of analysis this model can be viewed as a superposition of simpler ON-OFF mini-processes, NM of the type shown in Figure 2.17a, and N of the type shown in Figure 2.17b, then the state of the aggregate process model is the couplet (i, j) where i, j denote the number of each type of mini-processes which are in the ON state. A fluid flow analysis (as in [23]) has been carried out in [35] to determine the survivor function of buffer occupancy.

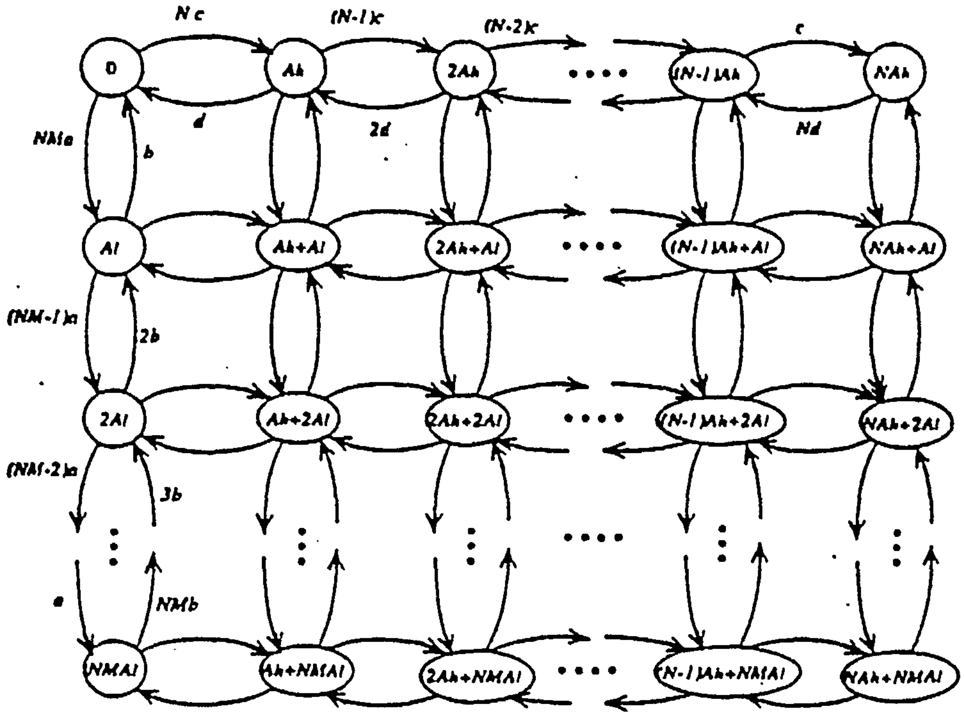


Figure 2.16: Continuous time, discrete state, Markov process model

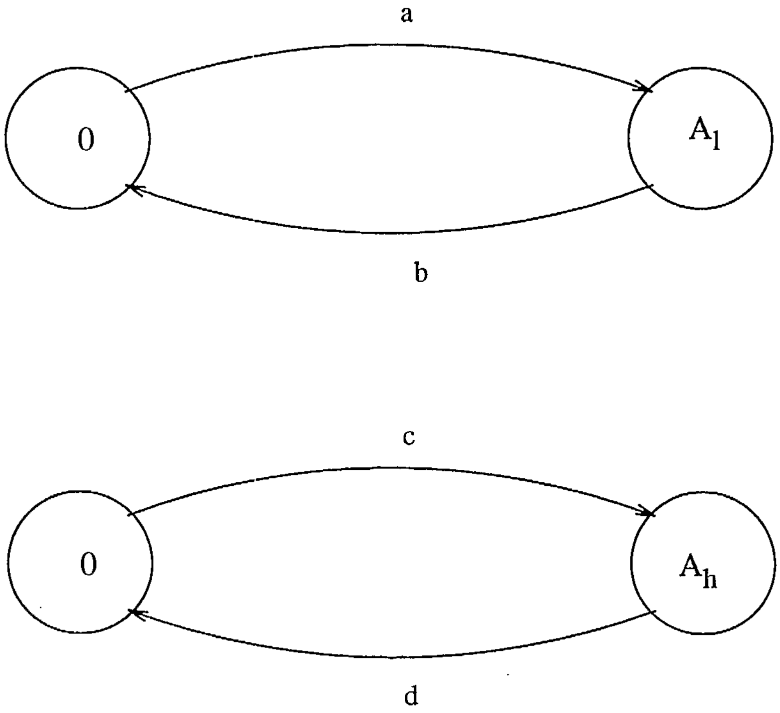


Figure 2.17: Miniprocess models

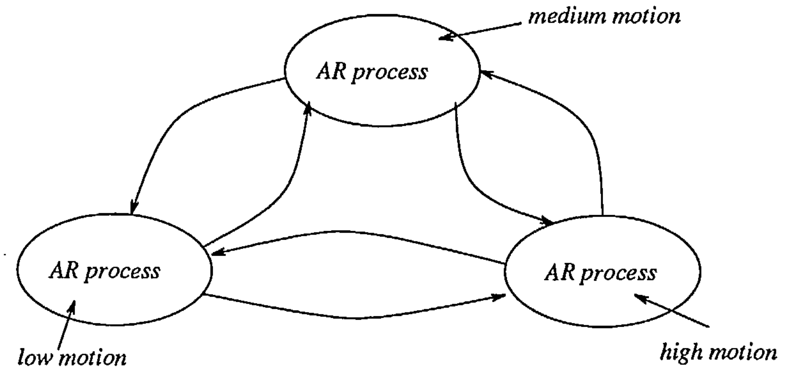


Figure 2.18: Markov modulated AR process

Markov modulated AR process

As seen before, an AR process captures short term correlations quite accurately . In [36], [37], [38] an AR process with time varying parameters is proposed as a model to characterize the bit rate process from a motion adaptive video codec (one that adapts the encoding scheme to the motion in the scene picturised). The time dependence of the parameters of the AR process captures long term corre lations. According to this model the no. of bits in a frame is given by a first o rder Gaussian AR process whose parameters are determined by the state of a Marko v chain(Figure 2.18). Thus each state of the Markov chain, with its own set of parameters, represents the various classes of motion. A Gaussian density was used because it was found from the study [36, 37, 38] that the bit rate distribution of the VBR coded full motion video can be represented by a composite Gaussian PDF.

In this model, the range of bit rates are separated into N adjacent intervals demarked by thresholds γ_i, $i = 1,2,\dots N-1$ and $0 \leq \gamma_1 \leq \gamma_2 \leq \gamma_{N-1}$. These bit rate intervals form the state space of the Markov chain. i.e., state 1 corresponds to the range $0 \leq \lambda_n \leq \gamma_1$ and state i corresponds to range $\gamma_{i-1} \leq \lambda_n \leq \gamma_i$, where λ_n represents the number of bits in frame n. If S_n denotes the state of the process at frame n, then the model can be represented mathematically as

$$\lambda_n = \begin{cases} a(i)\lambda_{n-1} + G(\mu(i), \sigma^2(i)) & \text{if } S_n = S_{n-1} = i \\ G(\eta(i), v(i)) & \text{if } S_n \neq S_{n-1};\ \ S_n = i \end{cases} \tag{2.46}$$

where G(.) denotes a Gaussian random variable with specified mean and variance. $\eta(i)$ and $v(i)$ denote the mean and variance of λ_n conditioned on state i. i.e., $\eta(i) = E[\lambda_n | S_n = i]$; $v(i) = var(\lambda_n | S_n = i)$. $a(i)$ is the correlation coefficient between the bit rates of two successive frames when the Markov chain is in state i.

Increasing the number of states N, results in an accurate model at the expense of increasing its complexity. The number of parameters of the model depends on the number of states, as a set of parameters characterize the AR process in each state. The parameters of the AR process in various states are obtained by matching the following statistics with the measured values, for each state:

$$\text{(a) mean bit rate in state } i, \eta(i) \quad = \quad \frac{\mu(i)}{1 - a(i)} \tag{2.47}$$

$$\text{(b) variance of bit rate in state } i, v(i) \quad = \quad \frac{\sigma^2(i)}{1 - a^2(i)} \tag{2.48}$$

$$\text{(c) measure of correlation of bit rates}, D^2(i) \quad = \quad \frac{2\,\sigma^2(i)}{1 + a(i)} \tag{2.49}$$

where $D^2(i)$ is the measure of correlation of bit rates between two succesive frames. i.e.,

$$D^2(i) = E[(\lambda_n - \lambda n - 1)^2 | S_n = S_{n-1} = i]$$

Hence the parameters of the AR process in each state, namely, $\mu(i)$, $\sigma(i)$, and $a(i)$ are obtained from equations 2.47 to 2.49.

The duration of a state is geometrically distributed with mean $\frac{1}{\theta_i}$,as given by the following p.d.f.,

$$F_i(k) = \frac{\theta_i}{1 - \theta_i}(1 - \theta_i)^k \qquad (k = 1, 2, \ldots)$$

where k is interms of the number of frames. The quantity $\frac{1}{\theta_i}$ and $\pi_{i,j}$ (probability the next state is j given that the present state is i) can be obtained from measurements directly. Then the transition probability matrix P can be obtained as

$$P = \begin{bmatrix} 1-\theta_1 & \theta_1\pi_{12} & \theta_1\pi_{13} & \dots & \dots \\ \theta_2\pi_{21} & 1-\theta_2 & \theta_2\pi_{23} & \dots & \dots \\ \theta_3\pi_{31} & \theta_3\pi_{32} & 1-\theta_3 & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix} \tag{2.50}$$

If the vector $\mathbf{p} = [p_1, p_2, \dots p_N]$ denotes the steady state probability (obtained by solving $\mathbf{p} = \mathbf{p}$ P and $\sum p_i = 1$) then the number of bits generated according to this model has the following PDF

$$f(x) = \sum_{i=1}^{N} p_i \, G(\eta(i), v(i)) \tag{2.51}$$

This model can also be used to characterize an aggregate of N sources. As before, though this model is a good candidate for simulation, it does not provide a suitable framework for a queueing analysis.

Model of Indices

A novel method of video traffic characterization, that does not depend on the variable bit rate coding algorithm employed is discussed in [39]. Here, a set of simple parameters called the indices that sufficiently characterize the video sequence are identified by working on the uncoded video sequence. The bit rate process from any coder is then predicted from a linear combination of the corresponding indices.

The parameters developed are grouped into 3 classes. One is derived from the histogram of the pixel information. The second is derived from the spatial correlation of the pixel values in a frame and the third set of indices are derived from the temporal correlations of the pixel values along the time axis.

The first class of parameters are derived from histogram of the pixel values of a single frame. Three indices are considered under this category, namely the average index (which gives a measure of the brightness in a frame), variance index(which gives a measure of the variability of the pixel values in a frame) and the entropy index (which represents the best possible compression performance for the codes that use first order statistics of the pixel values).

For the second class of parameters based on spatial correlation, the indices chosen were vertical entropy (which is entropy of difference in intensity between adjacent rows in a pixel array of a frame) and the horizontal entropy index (entropy of the difference in intensity between adjacent columns in a pixel array of a frame)

For the third class of parameters, based on temporal correlation, the indices chosen were difference index (reveals the difference in the amplitude of pixels between consecutive frames), motion index(the magnitude of the displacement vector corresponding to the pixels within the frame) temporal entropy index (entropy of the temporal difference in the vertical consecutive frames).

Study of the several coding schemes in [39] revealed that the output bit rate process was strongly correlated with some of the indices. Hence the output bit rate was predicted using a linear predictor model, a model fitting algorithm was then used to reduce the number of parameters according to linear regression measures of fit.

Though the model averts the necessity of modeling the bit rate process from different encoders separately, it cannot be used for an analytical evaluation.

Switched fractal source

In [40] and [41] a switched fractal source has been proposed as a model to characterize video source of less than 5 Mbps, using a highly compressed encoding scheme. Here the cell generation process is modeled directly in order to reproduce the bursty characteristics of the VBR traffic. In the encoding characterized, the original image

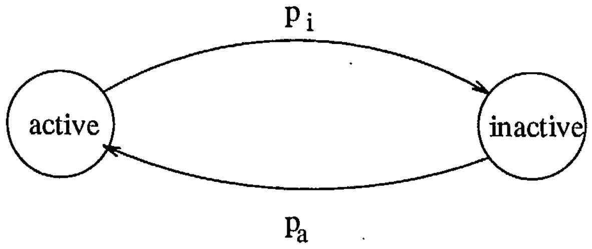


Figure 2.19: active/inactive process model

is divided into smaller sub-blocks, each block is transfered into another domain and the blocks of transform coefficients are scanned, coded and packaged into ATM cells. Due to the highly compressed nature of the coding scheme, the number of ATM cells produced after processing each block is small, either zero or one. Hence the cell generation process at the sub-block level can be modeled by a simple active/inactive process as shown in Figure 2.19. It was found that the transition probabilities p_a (inactive to active) and p_i (active to inactive) were dependent on the time spent in their present state. In particular the relationship is of the form $p(t) \propto t^D$ where D is known as the fractal dimension and the model is called a fractal model. Therefore

$$\begin{aligned} p_a(t) &= A_a\, t^{D_a} \qquad (2.52)\\ p_i(t) &= A_i\, t^{D_i} \qquad (2.53)\end{aligned}$$

where A_a, A_i are proportionality constants and D_a, D_i are fractal dimensions. The parameters are obtained from a logarithmic plot of experimentally measured active and inactive time periods. Such a fractal model accurately represents the cell traffic characteristics of uniformly active images.

In order to represent the traffic statistics of varying activity levels, a model that switches between multiple fractal sources is proposed. [41] discusses a five-mode fractal source model. The five cell generation modes correspond to average bit rates of 1, 2, 3, 4 and 5 Mbps. These five fractal sources were obtained by monitoring the traffic produced by five artificially constructed images, in which each of the image sub-blocks (when coded) produced average bit rates of 1-5 Mbit/sec, respectively. Logarithmic plots of the experimentally measured, active and inactive time periods were plotted. Parameters A and D of equations 2.52 and 2.53 are given by the y-axis intercept and straight line gradient of these plots, respectively.

To simplify the switching process, each row of sub-blocks in an image (thirty two 16 x 16 sub-blocks per row for 512 x 512 images) can be divided into image 'sections', each containing N ($N = 8$) sub-blocks. Switching between fractal sources is permitted only at the beginning of one of these sections. The switching scheme used is given by

$$L_n = \begin{cases} |L_0| & n = 0 \quad 1 \leq L_N \leq 5 \\ L_{n-1} + |\delta_n|, & n > 0 \end{cases} \tag{2.54}$$

where L_n is the activity level for image section n, L_0 is the 'starting' or 'average' activity level for the image (directly proportional to the average complexity of the image), δ is a normally distributed random variable with zero mean and standard deviation σ. The results of mean delay obtained by a queueing simulation using this model indicates that this model behaves similar to the 'real' traffic for network utilization levels upto 90%.

Self Similarity and VBR video

Recent studies [42] of VBR video have revealed that they exhibit the phenomenon of statistical self similarity. A self similar phenomenon exhibits structural similarities across all (or atleast a wide range) of time scales.

In [42], the results of detailed statistical analysis of a 2-hour long empirical sample of VBR video are discussed. The samples were obtained by applying a simple intraframe compression code to an action movie. The study showed that the autocorrelation function of the VBR video sequence decays hyperbolically, (a manifestation of long range dependence). The power spectral density or periodogram of the VBR video does not seem to approach zero near the origin, instead it obeys a power law of the form $\omega^{-\alpha}$ for $0 < \alpha < 1$, which is another indication of long range dependence. Also, the marginal bandwidth distribution possesses a "heavy-tail".

The above mentioned properties of long range dependence and heavy tailed marginals are not captured by conventional analytic source models. However these can be modeled by using self-similar processes. (Refer [42] for an explanation of self-similar processes). [42] also presents an algorithm for generating synthetic self-similar VBR video traffic.

2.3 Survey of data traffic models

Data traffic is highly bursty. Unlike real time traffic (voice or video), data traffic is delay or jitter tolerant, while being sensitive to losses. The statistical characteristics of data traffic are complex and application dependent. Modeling of data traffic is of fundamental importance in the performance evaluation and traffic engineering of packet (or BISDN) networks. Accurate models of packet data traffic may be used for analytical performance evaluation of packet data networks. Hence such models should be easy to implement and analyze, besides capturing the statistical characteristics of data traffic accurately. Many models have been proposed in the literature [43, 44, 45, 9, 46, 47, 48, 49] characterizing both individual traffic sources or a superposition of multiple sources.

Traditionally data traffic has been modeled by a Poisson process. Poisson process is characterized by arrivals with exponentially distributed inter-arrival times. Though the Poisson process provides an easy means for generating traffic, it is unrealistic [45, 50]. Other conventional models like fluid flow models [43], batch Poisson [44], MMPP [9] and HAP [46] incorporate some form of Markovian structure, to exploit the wealth of analytical tools already available. These models are appropriate for estimating many performance measures of interest and have been used with some degree of success.

Recent studies of packet data traffic [51, 52, 53] in local area networks have thrown more light on the modeling of data traffic. The studies revealed that data traffic exhibits long range dependence and *statistical self-similarity*, i.e, the traffic exhibits "burstiness" across a wide range of time scales, ranging from milliseconds to minutes to hours. These findings have paved the way for more accurate models called the *fractal* models [47, 48, 49] that capture the long term correlations and burstiness of data traffic.

2.3.1 Overview of data traffic models

The statistical characteristics of data models are complex and application dependent. There have been many models proposed in the literature for characterizing individual data traffic sources or a superposition of multiple sources. The conventional models like fluid flow, batch Poisson, MMPP and HAP incorporate some form of Markovian structure, either in the way the way the arrival processes are modulated or in the arrival process themselves, for reasons of mathematical tractability. Thus these models are good candidates for the analytical performance evaluation of packet data networks.

However, the recent studies of data traffic in local area networks [51, 52, 53] reveal that data traffic exhibits long range dependence and statistical self-similarity. Self-similar phenomena display structural similarities across all (or atleast a wide range of) time scales. Figure 2.20 reproduced from [50] helps explain the concept of self-similarity pictorially. Figure 2.20 (sets of figures on the left) shows a sequence of simple plots of the packet counts for five different choices of time units. From Figure 2.20 it is evident that

a) plots of traffic measurements at various time scales look intuitively similar to one another (statistical self-similarity).

b) plots are distinctively different from white noise.

c) plots show that at every time scale ranging from milliseconds to minutes and hours, bursts consists of bursty sub-periods separated by less bursty sub-periods.

The conventional models mentioned previously, do not capture the aspects of statistical self-similarity and long range dependence in data traffic, as shown in Figure 2.20 (right side)) which shows the traffic plots generated by a batch Poisson model. Hence new models that can represent self-similar (or fractal) characteristics

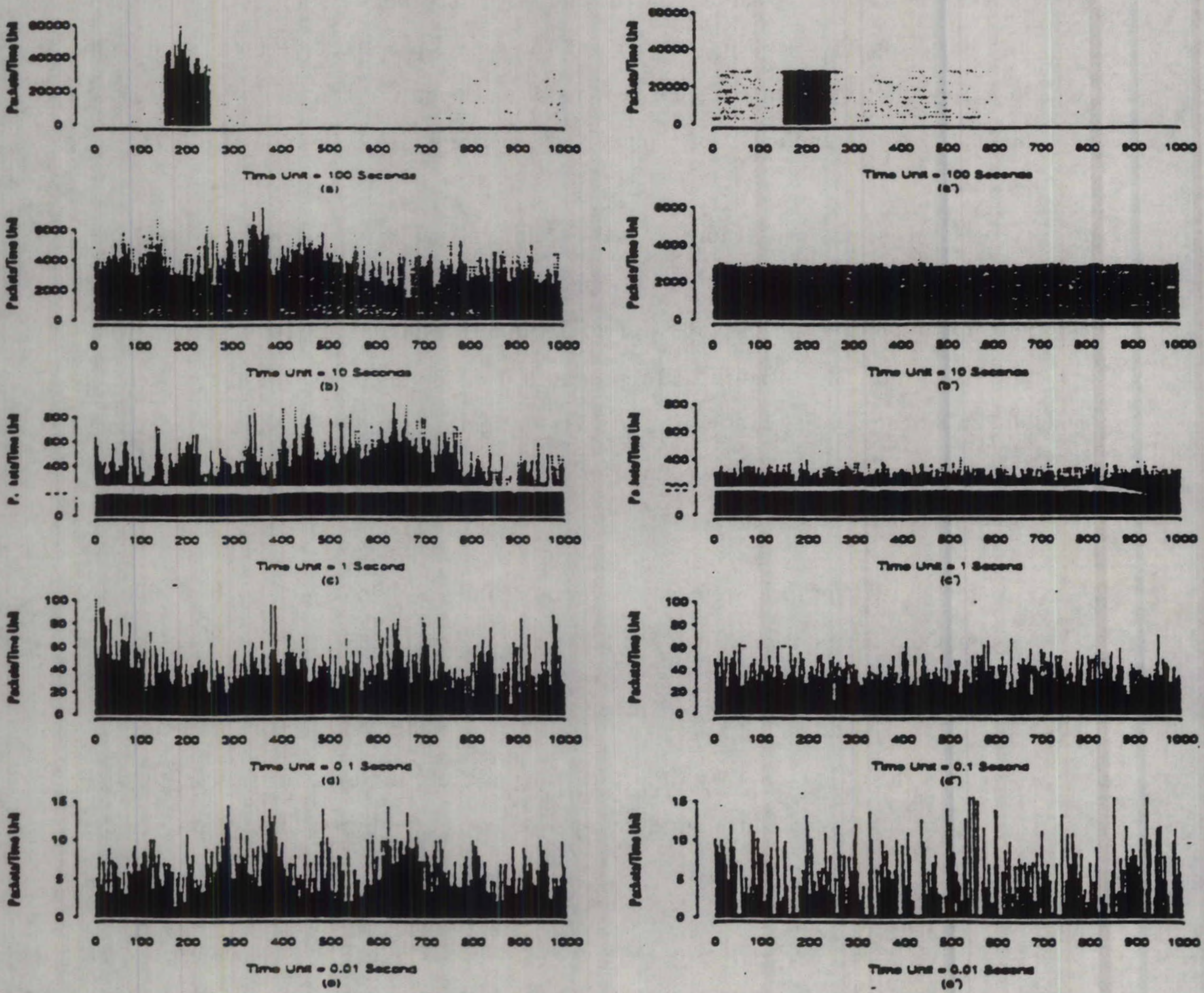


Figure 2.20: Ethernet traffic (packets per unit time) on five different time scales (left side). Synthetic traffic from compound Poisson model (right side).

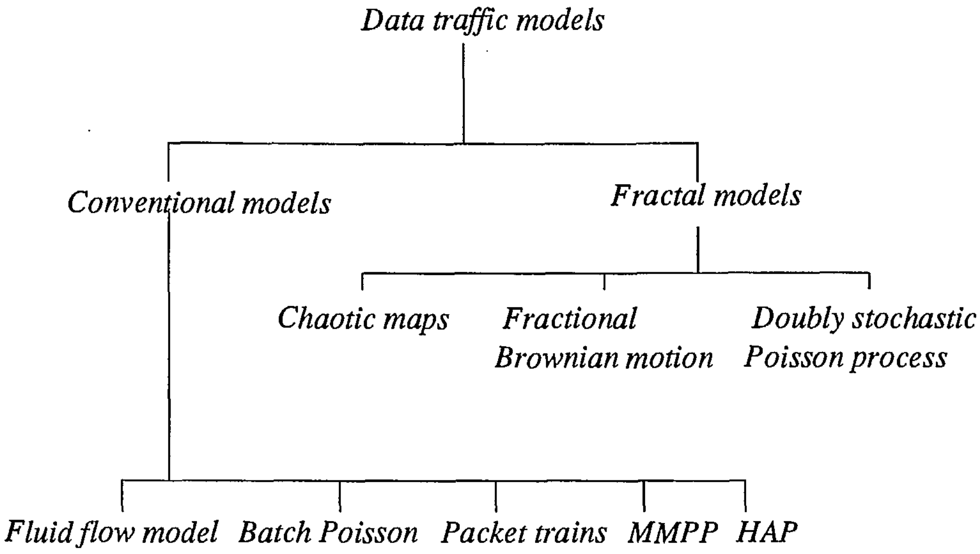


Figure 2.21: Models for data traffic

have been proposed [47, 48, 49]. Thus the modeling approaches for data traffic may broadly be classified as shown in Figure 2.21.

The implications of fractal nature of traffic are manifold. In order to understand them, we present a brief outlook on the manifestations of self-similarity in packet data traffic [54, 50, 55, 56].

Let $X = (X_1, X_2, X_3, \ldots)$ be a covariance stationary stochastic process. For each $m = 1, 2, 3, \ldots$, let $X^{(m)} = (X_k^{(m)} \; ; k = 1, 2, 3, \ldots)$ denote a new (aggregated) time series obtained by averaging the original series X over non-overlapping blocks of size m. i.e, for each $m = 1, 2, 3, \ldots$, $X^{(m)}$ is given by $X_k^{(m)} = 1/m(X_{m(k-1)} + \ldots + X_{km})$. Let a_1, a_2, a_3 be constants. The data traffic exhibits the following self-similar properties:

(a) *slowly decaying variances:* variance of sample mean decreases more slowly than the reciprocal of the sample size.

$$\text{i.e., } \mathrm{Var}(X^{(m)}) = a_1 m^{-\beta} \quad \text{with} \quad 0 < \beta < 1$$

- For conventional models the variance of the sample mean decreases like m^{-1}.

(b) *long range dependence:* autocorrelations decay hyperbolically rather than exponentially,

$$\text{autocorrelation} \quad r(k) = a_2 k^{-\beta} \quad \text{with} \quad 0 < \beta < 1$$

implying a nonsummable autocorrelation function, $\sum r(k) < \infty$. The self-similarity parameter or Hurst parameter $H = 1 - \frac{\beta}{2}$.

- For conventional models the autocorrelation decays exponentially and is thus summable.

(c) *1/f noise:* The spectral density f(.) obeys a power law behaviour near the origin. i.e.,

$$f(\lambda) = a_3 \lambda^{-\alpha} \quad \text{as} \quad \alpha \longrightarrow 0 \quad \text{with} \quad 0 < \alpha < 1$$

- Conventional models have spectral density broadened at the origin. i.e., $\alpha = 0$.

(d) *Fractal Dimension:* The fractal dimension (or correlation dimension) [55] for data traffic is less than 1.

- Conventional models have fractal dimension equal to 1.

Implications of self-similarity in data traffic

- Due to the fractal nature of data traffic, the expected number of arrivals in an interval of length t may scale as λt^D, where λ is packet arrival rate and D is the fractal dimension. Hence standard engineering measurements such as rates, utilizations and occupancies may be arbitrary in that they depend critically on the length of the measurement interval. (i.e., $\frac{N(t)}{t} = \lambda t^{D-1}$).

- The degree of self similarity measured in terms of the Hurst parameter H or the fractal dimension D, provide a satisfactory measure of burstiness (burstier the traffic, higher the value of H and lower the value of D). Other commonly used measures of burstiness such as index of dispersion (for counts), peak to mean ratio or coefficient of variation are meaningless, since for fractal traffic these measures can assume any value depending on the length of the interval over which these measurements are made.

- The presence of low frequencies in the spectral density (or equivalently the slowly decaying autocorrelation and variances) causes heavy losses and long delays during long time frame bursts. Hence nature of network congestion produced by fractal traffic differs drastically from that predicted by conventional traffic models.
- For fractal traffic the overall packet loss decreases very slowly with increasing buffer size.
- Source models for individual sources are expected to show extreme variability in terms of the inter arrival times of packets.
- Aggregation of bursty traffic streams does not produce smooth "Poisson-like," superposition process as previously assumed. Hence new traffic models that capture long range dependence and fractal properties are required.

The fractal models that have been proposed in the literature account for the self-similar phenomena exhibited in data traffic. However all the fractal or self-similar models proposed do not lead to tractable analytic solutions. On the other hand, the conventional traffic models that are blessed with a wealth of analytical tools, fail to capture the long term correlations and fractal properties of packet traffic. The models currently considered in literature (like Markov model, MMPP, ARIMA, etc.), may be used to capture fractal properties. However the process of modeling long range dependence with the help of short-range dependent processes is equivalent to approximating a hyperbolically decaying autocorrelation function by a sum of exponentials and hence requires a large number of parameters. Parsimonious modeling of fractal properties by conventional models can be achieved by resorting to some approximations.

2.3.2 Conventional models

This section gives a brief overview of each of the conventional models. As already mentioned, these models do not capture the long term correlations and self-similar properties of data traffic.

Fluid flow model

The fluid flow model [43] (also referred to as Uniform Arrival and Service model (UAS)) assumes that the information flow in and out of the buffer (at the multiplexer) is uniform and continuous rather than in discrete packets. In this model the source generates information to the transmitter at the rate of one unit of information per unit time and the server removes information from the buffer at a uniform rate not to exceed C units of information per unit time. With these assumptions the equilibrium queue distribution is described by a set of differential equations, which together with a set of boundary conditions can be solved to yield the equilibrium queue distribution. The method is outlined in the section on voice traffic models.

Though this modeling methodology leads to a tractable analysis, its largest drawback is that it cannot model the short-term queue increases that occur when two or more packets arrive almost simultaneously.

Batch Poisson model

The batch Poisson model [44] is an extension of the Poisson model. Here the arrivals occur in batches. The batch arrival is Poisson. The batch size can be random. However a geometric batch size helps to derive simple analytical results. Besides the burstiness captured by this model, correlations can also be modeled by choosing the batch size distribution of successive batch arrivals according to a Markov chain. The batch Poisson model is a special case of the general Batch Markovian Arrival Process (BMAP) for which extensive analytical (transient and steady state) results exist [57]. Hence this model provides an efficient means for analysis.

However study of data traffic [45] has indicated that simultaneous or back to back arrival of packets are rare (due to the finite packetization time). Hence the model is not realistic.

Packet trains model

In [45] a new model called the packet trains model is proposed to characterize the data traffic in a token passing ring LAN. The model is based on the observation that data traffic exhibits *source locality* (i.e., given a packet going from node A to B, there is a high probability that the next packet will be going from node A to B or from B to A. The traffic on the network (here a token passing ring) is divided into a number of packet streams between various pairs of nodes of the network. Each node-pair stream consists of a number of trains. Each train consists of a number of packets (or cars) going in either direction (from node A to B or node B to A), as shown in Figure 2.22. The intercar time is smaller than a (user) specified maximum called maximum allowed intercar gap (MAIG). The inter-train time is larger than MAIG. Hence the inter-train time is a user parameter, while the inter-car interval is a system parameter. Partitioning of the network into streams based on node-pair processes as

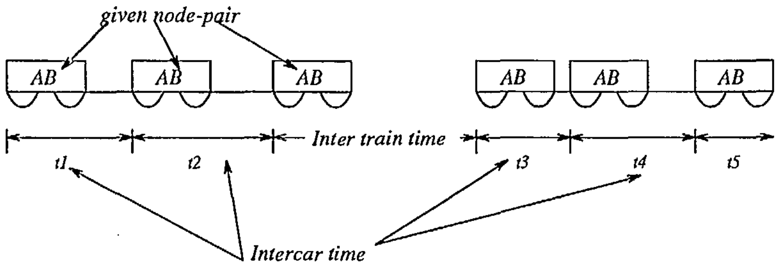


Figure 2.22: Packet train model

explained above helps increase the predictability of data traffic, since they make use of the property of source locality inherent in data traffic. Hence this model is good for simulation purposes.

MMPP models

Markov modulated Poisson Process (MMPP) is a nonrenewal, doubly stochastic Poisson process where the rate process is determined by the state of a continuous time Markov chain. In other words underlying is a continuous state Markov chain, where the sojourn time for state j is exponentially distributed with mean r_j^{-1}. When in state j, cells are generated according to a Poisson process with rate λ_j. [9] uses a two state MMPP and approximates the traffic of multiple data and voice sources. The details of fitting the MMPP to the data and voice traffic is discussed in the section on voice traffic.

HAP models

The HAP (Hierarchial Arrival Process) model is based on the fact that there are many processes modulating a single packet arrival stream. For example the long-term correlation depends on the user and application behaviour, while the short-

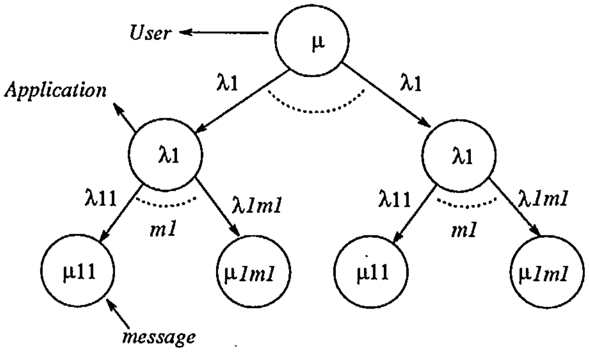


Figure 2.23: HAP model

term correlation depends primarily on the network hardware and software. HAP [46] models both the short-term and long-term correlations by modeling the arrival process at 3 levels - *user, application and message* (Figure 2.23). A set of parameters describe the arrival and departure processes at each level. As shown, users arrive in the system according to an interarrival time distribution (with mean λ) and stay in the system according to a service distribution (with mean μ). The user may invoke applications according to an interarrival time distribution (with mean λ_i) which may remain active according to a specified distribution (with mean μ_i). During the active interval, the application generates several types of messages with different rates and with different message size distributions. The HAP can be mapped into a MMPP [46] and analysis can be carried out with the resultant MMPP.

The HAP model captures the correlation at different levels. It also lends itself to analysis easily. However, the HAP, models the arrival process only at a message level.

2.3.3 Fractal models

This section briefly discusses the models proposed in the literature to capture fractal properties of packet traffic.

Chaotic Maps

[47] uses deterministic chaotic maps to model fractal properties in packet traffic. Chaotic maps are low dimensional non linear systems whose time evolution is described by a knowledge of an initial state and a set of dynamical laws. The trajectory of chaotic system are very often fractal in nature. Hence by adjusting the parameters of the chaotic maps it is possible to capture the fractal nature of packet traffic.

Consider a one-dimensional map in which the state variable x_n evolves over time according to the non linear map:

$$x_{n+1} = f_1(x_n) \qquad y_n = 0 \qquad (0 < x_n \leq d)$$

$$x_{n+1} = f_2(x_n) \qquad y_n = 1 \qquad (d < x_n < 1)$$

The packet generation process is modeled as follows:

- The source alternates between a passive and active state.
- When $y_n = 0$ $(0 < x_n \leq d)$ the source is in passive state and when $y_n = 0$ $(d < x_n < 1)$ the source is in active state (Figure 2.24).
- Every iteration of the map in the active state is taken to generate a packet (or batch of packets).
- suitable $f_1(.)$ and $f_2(.)$ should be chosen so that properties of y(n) match those of actual packet traffic.

The *Intermittancy Map* with $f_1(.)$ and $f_2(.)$ as given below captures fractal properties of data traffic well [47]

$$x_{n+1} = \begin{cases} \epsilon + x_n + cx_n^m & 0 < x_n \leq d \\ \frac{x_n - d}{1-d} & d < x_n < 1 \end{cases}$$

where $c = \frac{1-\epsilon-d}{d^m}$ (Figure 2.25)

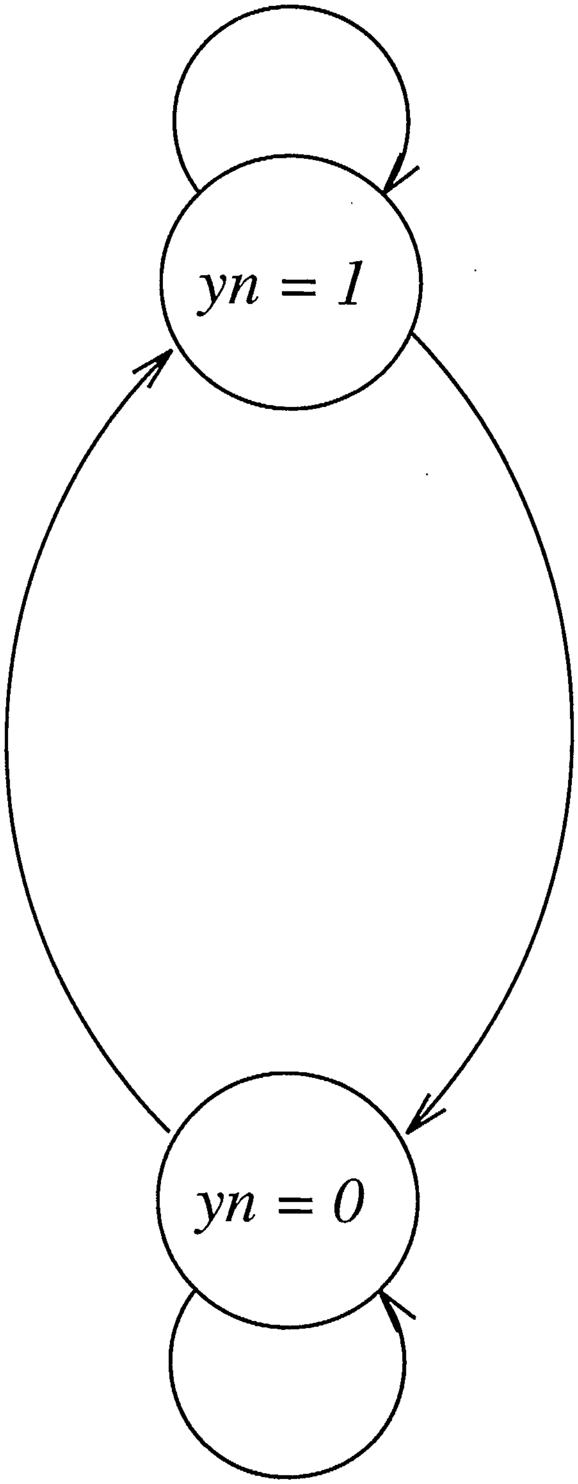


Figure 2.24: Basic source model (Chaotic Map)

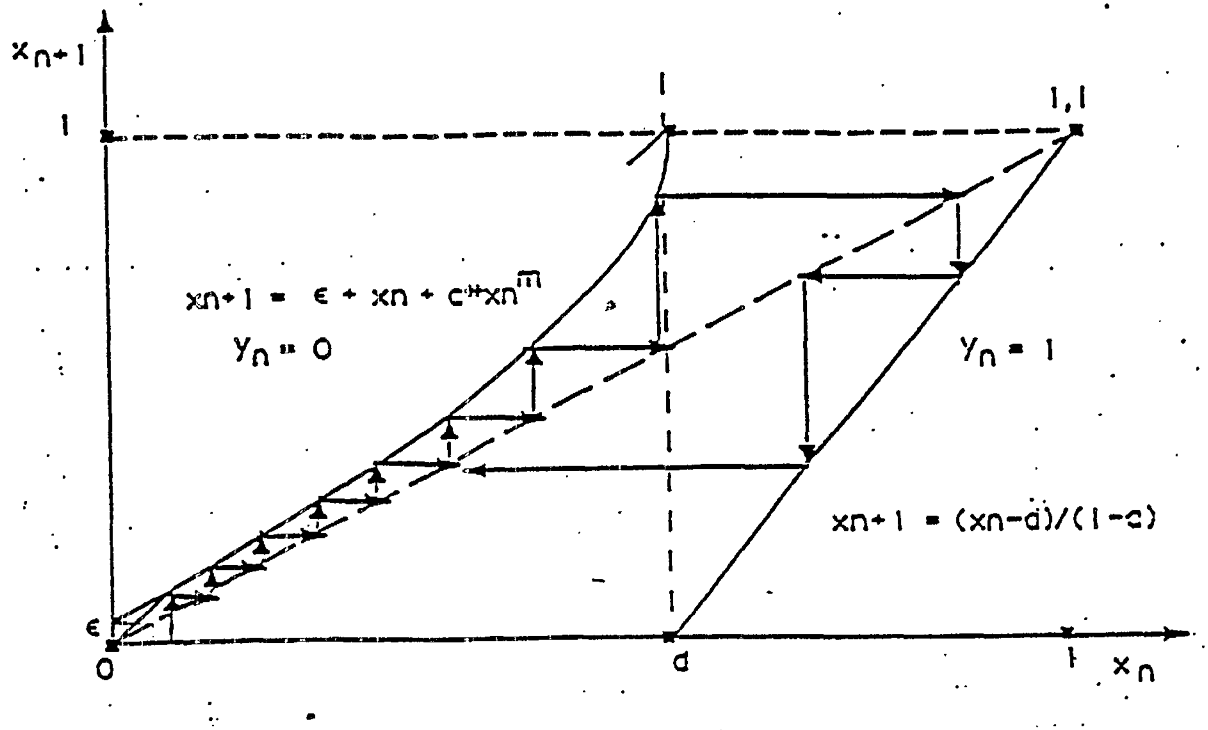


Figure 2.25: Intermittancy map

While chaotic maps is effective in characterizing much of the fractal properties of data traffic like 1/f noise, "thick-tail" behaviour of interarrival time densities, etc., using very few parameters, there are considerable analytical difficulties in their application.

Fractional Brownian Motion model

The fractional brownian motion is a self-similar process. i.e., if $Z(t)$ is a brownian motion process then $Z(\alpha t)$ is identical in distribution to $\alpha^H Z(t)$, where $(1/2 < H < 1)$ is the self-similarity parameter. In [48] a model based on Fractional Brownian Motion is proposed to characterize the self-similar properties of packet traffic. The following model is studied :

$$A(t) = mt + \sqrt{am}Z(t)$$

where $A(t)$ is the number of cell arrivals to the multiplexer in the time interval $(0, t]$, m is the arrival rate of a Poisson process and $Z(t)$ is a fractional brownian motion with self-similarity parameter H. The above model is based on a diffusion approximation of the number of arrivals from a Poisson process. The parameters of the model are H, m and a. The above model could also be used to characterize the superposition of N independent and identically distributed cumulative traffic processes. Hence now $A(t) = \sum_{i=1}^{N} A_i(t)$. Now, the parameters H and a characterize the type of the traffic mix while m gives its amount. In [48] an analysis is done by using a storage model based on $A(t)$ as the input process.

Doubly stochastic Poisson process

Doubly stochastic Poisson process is a time dependent Poisson process in which the rate of the Poisson process $\lambda(t)$ is a stationary stochastic process in continuous time.

By choosing an appropriate stochastic process for the intensity of the Poisson process, the fractal properties of packet traffic can be modeled [49]. [49] considers the following stochastic process for $\lambda(t)$

$$\lambda(t) = \mathbf{a}(1 + \sum_{i=1}^{\infty} a_i cos(\omega_i t + \theta_i))$$

where $\mathbf{a}$ is Rayleigh distributed random variable and θ_i are i.i.d. random variables uniform over the interval $[-\pi, \quad +\pi]$. It is shown in [49] that by choosing suitable values for the constants a_i such that the increment $\lambda(t+h) - \lambda(t)$ is a fractional brownian motion process, the self-similar nature of packet traffic can be captured. This model is easy to simulate but will not lead to a queueing model of manageable complexity. To allow the ease of analysis, this model can be approximated by a discrete state Markov model, with the intensity of the Poisson process quantized into discrete levels. Now, the transition between levels are assumed to occur with exponential transition rates, depending on the current level.

Chapter 3

Modeling of aggregate multi-media traffic

In this chapter we address the problem of modeling aggregate multi-media traffic, discussing in detail the new model proposed. As outlined in the previous chapter, various constituents of aggregate multi-media traffic exhibit a diverse mixture of traffic characteristics. Our goal is to develop a model that aptly characterizes the variability and statistical correlations in the packet arrival process. The developed model is to be used for network performance evaluation or evaluation of multiple access schemes or for evaluating/devising connection admission control and source policing algorithms. In other words the model developed need only characterize the statistical correlations and burstiness (or variability) present in the arrival process. Also, recent studies of LAN data traffic indicate that such traffic exhibits long range dependence and self-similar (or fractal) characteristics, i.e., the traffic exhibits "burstiness" across a wide range of time scales ranging from milliseconds to hours. Hence, in a multi-media environment fractal traffic co-exists with non-fractal traffic. Characterizing such a mix of traffic by an unique model poses a great challenge to the modeller . The model proposed should be versatile in the sense that it should be able to capture the long term and short term correlations of the multiplex. Also, the model should be parsimonious in the number of parameters, lest the parameters lose their physical significance.In the few aggregate models proposed [58] [59] [60] [61] [62] [63] for multimedia traffic, the self-similarcharacteristics of the component traffic has not been accounted for. Motivated by this fact we suggest a new aggregate model consisting of switched Poisson processes.

The doubly stochastic Poisson process model was examined as a candidate model. The doubly stochastic Poisson process $N(t); t <= 0$ is a time dependent Poisson process $\Lambda(t)$, in continuous time, i.e, in each realization of the series of events $\lambda(t)$ varies with time and is itself a realization of a stochastic process. The MMPP (Markov modulated Poisson process), a special case of doubly stochstic Poisson process, has previously been succesfully used to model the arrival process from a set of voice sources [9], [10] and a set of video sources [58], [59], [60], [61] .The MMPP is itself a correlated non-renewal stream. In these methods the MMPP models can accurately characterize the aggregate arrival process (either from a set of voice sources or from a set of video sources as the case may be) because a large number of statistics can be matched and the correlations among the arrival process accurately captured. Hence an aggregate of voice and aggregate of video sources may be accurately characterised by the MMPP.Hence once again, the stress is on capturing the statistical correlations as found in the aggregate traffic.

However, data traffic exhibits long range dependence and statistical self-similarity. Also, earlier measurements of data traffic [64] indicate that the message length distribution is bimodal. Since a burst of packets are produced for each message, this also suggests that the burst of data packets may be bimodally distributed. Thus the data traffic may consist of short and long bursts. Also, as observed in [65], the number of bytes in each burst has a very heavy upper tail. This suggests that when one of the burst states begin, they extend for a longer time. Based on these observations, we propose a new model for data traffic. We model the data traffic by a 2-state doubly stochastic Poisson process, with sojourn times in each state having an independent and identical heavy tailed distribution, such as the Pareto distribution. The two states of the switched Poisson process may correspond to the long and short burst rates. This model captures the long range dependence present in data traffic.

This chapter of the report is organised as follows. The next section discusses about the MMPP model used for voice and video. The following section presents the new model (PMPP) proposed in this research. The last section discusses about the aggregate model and the simulation results.

3.1 Characterization of aggregate voice and video traffic

3.1.1 Characterization of aggregate voice traffic

A single voice source may be modeled by the well-known ON-OFF process in which the voice source alternates between exponentially distributed ON and OFF periods with parameters a and b respectively. While in the ON state the source generates packets at a constant rate r aand in the OFF state no packets are generated.

Now consider the superposition of N voice sources of the type mentioned above. This results in a $N+1$ state birth-death process. The state space grows as the number of sources in the superposition is increased. Three main approaches are proposed in the literature [58] for the representation of the superposition of on-off sources. The first approach explicitly takes into account the individual component of each of the sources [66, 16, 67]. The second is based on matching a few of the statistical parameters of the aggregate arrival process with that of a suitably chosen simple arrival process such as that of the MMPP. The last approach resorts to the fluid flow approximation [43, 14]. The first approach has the limitation that the computational complexity dramatically increases in practical cases while the fluid flow approach cannot account for the cell level and its results are not accurate for large buffer sizes [14]. The second approach is the widely used one. Here, as originally proposed in [9], the aggregate packet arrival process from the superposition of many voice sources may be represented by a doubly stochastic Poisson process which is modulated in a

Markovian manner. Heffes and Lucantoni approximate the aggregate packet arrival process by a 2 state Markov modulated Poisson process (MMPP). The approximating MMPP model is chosen in such a way that its statistical characteristics match those of the aggregate traffic from the voice sources. There are 4 parameters for the chosen 2 state MMPP, namely, the mean sojourn times r_A^{-1} and r_B^{-1} in states A and B respectively and the Poisson arrival rates, λ_A and λ_B in states A and B respectively. They propose a matching technique by which the four parameters of the MMPP may be determined from the statistical characteristics of the original superposition.

3.1.2 Characterization of aggregate video traffic

Video sources with uniform activity level may be modeled by the model originally proposed by Maglaris et. al. [23]. In this model each video source is represented by a continuous-time, discrete state Markov chain. The bitrate from a source is quantized into M discrete levels of stepsize γ . The model switches between the various levels spending exponentially distributed time in each level. As noted in [23] the continuous-time, discrete-state Markov chain may be constructed from the superposition of M mini-sources, where each mini-source is in one of the states ON or OFF. When ON it generates packets at a constant rate and when OFF, does not generate any packets. Thus an ON-OFF characterization is given to the video traffic as well and following the same approach as in the case of voice, the superposition of video sources may also be approximated by a 2-state MMPP. Several matching techniques [58] [59] [60] [61] have been proposed to obtain the parameters of the resultant MMPP, when the constituent ON-OFF processes are bursty.

Hence the aggregate of voice sources and the aggregate of video sources may each be mapped into a MMPP. The MMPP is a correlated nonrenewal stream and hence it can account for correlations for the input process. Also, with the MMPP a large number of statistics can be matched and the correlation among the arrival process can be captured over larger time intervals. The MMPP is a special class of the random hazard function considered in [68] and [69]. The statistics of the MMPP like mean, Variance, IDC (Index of Dispersion for Counts) may be derived from the probability generating function of the process. This is outlined in the Appendix. (The IDC for MMPP has already been derived in [9]. However, here we take a different approach). The IDC of MMPP is given by

$$\begin{aligned} I(t) &= 1 + \frac{2(\lambda_A - \lambda_B)^2 r_A r_B}{(r_A + r_B)^2(\lambda_A r_B + \lambda_B r_A)} \\ &\quad - \frac{2(\lambda_A - \lambda_B)^2 r_A r_B}{(r_A + r_B)^3(\lambda_A r_B + \lambda_B r_A)t}(1 - \exp^{(-(r_A+r_B)t})\end{aligned} \qquad (3.1)$$

where λ_A (λ_B) is the rate in state A (state B) and r_A^{-1} (r_B^{-1}) is the mean sojourn time in state A (state B).

The IDC of a process is indicative of the burstiness of the process. In figure 3.1 the dotted curve shows the log-log plot of IDC of MMPP for $\lambda_A = 100$, $\lambda_B = 120$, $r_A = r_B = 0.33$. As seen inthe curve the IDC settles down at a value after an initial linear behaviour (in log-log plot). A linear behaviour of IDC with time (in log-log plot) for ever, as is the curve shown in solid lines would indicate the presence of long term correlations. Hence MMPP is able to capture the burstiness only over a certain range of intervals. In the case of data traffic, which has been found to possess long term correlations, the MMPP may not be an adequate model. This stresses the need of a new model that could capture the long term correlations. The next section proposes a simple model called the PMPP, the IDC of which is plotted in figure 1.(λ_1 = 100, λ_2 = 120 and α = 1.5.)

3.2 Characterization of aggregate data traffic

Earlier measurements of data traffic [64] indicate that the message length distribution of data traffic is bimodal. Since a burst of packets are produced for each message, this also suggests that the burst of data packets may be bimodally distributed. As noted in [70], if a source generates a long burst of data like file transfer among short bursts which may correspond to commands, the source traffic essentially consists of short and long bursts. Hence the net data traffic from many such data sources is also likely to be bimodal and can be rightly characterized as a switched Poisson process that switches between the longer and shorter burst rates. Till now many models based on switched Poisson processes (like Batch Poisson, MMPP,etc.,) have been proposed to characterize data traffic in broadband networks. But recent studies [51] [52] [53] [54] [50], of packet data traffic in local area networks have thrown more light on the characteristics of data traffic. The studies revealed that data traffic exhibits long range dependence and statistical self-similarity (or *fractal* characteristics), i.e., the traffic exhibits "burstiness" across a wide range of time scales ranging from milliseconds to minutes to hours. The characteristics of such traffic are markedly different from those of the traditionally used models to characterize data traffic. Such traffic is characterized by long range dependent correlations, a spectral density that diverges at the origin and by variances that decay as fractional power of the sample size.

Mandelbrot originally suggested [71] that the superposition of many sourc es which exhibit the "Noah effect" (or infinite variance syndrome) results in a self-similar stream. In [54] [50] Leland *et. al.* employ this method to provide an explanation for the observed self-similarity of the traffic interms of the nature of the traffic generated by an individual source. They suggest that each of the individual sources contributing to the self-similar traffic stream can be represented by the familiar on-off abstraction. However, these on-off sources exhibit the "Noah effect" in that they have a highly variable on and off periods (sojourn times). i.e., the sojourn times of the on-off sources are charcterized by "heavy-tailed" distributions. Similar conclusions were also made in [53] based on studies on individual ISDN data traffic sources. Hence the sojourn times of individual sources can aptly be characterised by a heavy tailed distribution like the stable Pareto distribution.This distribution has a survival function of the form:

$$P\{X \geq x\} = x^{-\alpha} \qquad \alpha > 0, x > 1 \tag{3.2}$$

The density function is given by

$$p_X(x) = \frac{\alpha}{x^{\alpha+1}} \qquad \alpha > 0, x > 1 \tag{3.3}$$

The parameter α denotes the thickness of the tail of the distribution. If $1 < \alpha < 2$, then the Pareto distribution posseses an infinite variance but a finite mean as given by

$$E(X) = \frac{\alpha}{\alpha - 1} \tag{3.4}$$

The tail of the stable Pareto distribution decays far more slowly (by a power law) than an exponential distribution. A Pareto distributed random variable takes a larger value with a higher probabilty than an exponentially distributed random variable. Higher the value of α, thicker the tail of the distribution. It has also been proved in [72] that if $1 < \alpha < 2$ for the sojourn times of the constituent on-off processes then the resultant superposition process is self-similar with Hurst parameter H = (3 - α)/2. (The Hurst parameter is a measure of the self- similarity).

The above characterization gives an insight into the behaviour of the individual sources. However, in practice, it is just sufficient to capture the self-similar characteristics of the aggregate traffic stream. Self-similarity is measured by the Hurst parameter H. Self-similar processes with $0.5 < H < 1$ are also long range dependent. Processes with a high Hurst parameter (in the vicinity of 1) are highly bursty while those with a low Hurst parameter (near 0.5) are less bursty. Hence the Hurst parameter H is indicative of the resultant aggregated stream. The aggregated stream may directly be modelled by processes which can exhibit self-similar characteristics.

Based on these observations, we approximate the aggregate data traffic by a doubly stochastic Poisson process. A doubly stochastic Poisson process is a time dependent Poisson process where the intensity function or the mean rate of occurence of events is a stochastic process. [49] illustrates the versatility of this process in characterizing self-similarity. The stochastic process considered in [49] is a continuous one. Here we model data traffic as a Poisson process alternated between 2 levels λ_1 and λ_2. The sojourn times in these two states are independent and identically distributed with

Pareto distribution with parameter α. The two states of this switched Poisson process would correspond to the long and short burst rates. The sojourn time distribution is chosen to be a thick tailed one in order to capture the long term dependencies in the net arrival process. Since the Poisson process is switched between two rates by the underlying Pareto distribution, we call this model a Pareto modulated Poisson process (PMPP).

In order to determine if the model captures the long term correlations, we looked at the IDC (Index of dispersion of Counts) and the Variance time plots of the model. For a given time interval of length t, the Index of dispersion of count s is given by the ratio of the variance of the no. of arrivals during the interval to the mean of the number of arrivals in the same interval. If we divide the time axis into equal intervals called frames and if $(X_1, X_2, X_3, \ldots)$ a re the number of packets generated by the process in succesive frames, then IDC is defined as follows.

$$IDC(t) = Var(X_1 + X_2 + X_3 + ...X_t)/nX_{avg} \tag{3.5}$$

where X_{avg} is the average number of packets generated in a frame. IDC of a process is indicative of the burstiness of the process. Pure Poisson process has a IDC of 1. A process having IDC greater than one is overdispersed while that having IDC below one is underdispersed. For a self-similar stream of Hurst parameter H, IDC increases monotonically and is proportional to t^{2H-1}. Hence such an IDC when plotted in a log-log plot produces a straight line appearance.The value of the Hurst parameter, H, of the stream may then be calculated from the slope m of the IDC curve, in log-log plot. i.e.,

$$H = (m+1)/2 \tag{3.6}$$

The PMPP model considered is akin to the random hazard doubly stochastic Poisson process considered in [68] and [69]. Generally stated such processes alternate between two levels λ_1 and λ_2, with the sojourn times in each state forming an alternating renewal process with interval p.d.f.'s $f_1(x)$ and $f_2(x)$ respectively. If ν_1 (ν_2) is the average sojourn time in state 1 (state 2), $f_1^*(s)$ ($f_2^*(s)$) is the laplace transform of the p.d.f of the sojourn time in state 1 (state 2) and if $R_1^*(s)$ $(R_2^*(s))$ is the laplace transform of the survivor function in state 1 (state 2), then the laplace transform of the probability generating function of the process is given by [69]

$$\begin{aligned}\phi^*(z,s) &= \frac{1}{\nu_1+\nu_2}\left(\frac{\nu_1}{s+\lambda_1(1-z)}+\frac{\nu_2}{s+\lambda_2(1-z)}\right)\\ &\quad -\frac{(\lambda_1-\lambda_2)^2}{\nu_1+\nu_2}\left(\frac{(1-z)^2}{(s+\lambda_1(1-z))(s+\lambda_2(1-z))}\right)\\ &\quad \mathrm{x}\left(\frac{R_1^*(s+\lambda_1(1-z))R_2^*(s+\lambda_2(1-z))}{(1-f_1^*(s+\lambda_1(1-z)f_2^*(s+\lambda_2(1-z)))}\right)\end{aligned} \tag{3.7}$$

An explicit expression for the laplace transform of the mean and variance may be obtained from the above equation. These may inturn be inverted to yield the mean and variance of the doubly stochastic process, from which the expression for IDC may be derived. For the case of PMPP an explicit expression for IDC has been derived as detailed in the Appendix. The expression for the IDC of PMPP is

$$IDC(t) = 1+\frac{(\lambda_1-\lambda_2)^2}{\lambda_1+\lambda_2}\left(\frac{\alpha-1}{\alpha}\right)t^{2-\alpha} \tag{3.8}$$

As seen from the above expresion, IDC increases as a fractional power of the interval under consideration. Such is the characteristics of a long range dependent self-similar process. When plotted in a log-log scale the IDC has a slope m equal to $2-\alpha$. From (6) the Hurst parameter, H may be derived from the slope m as follows

$$H = \frac{3-\alpha}{2} \tag{3.9}$$

We arrive at the same realation as in [72]. Hence as we vary the parameter α of the pareto distribution, the Hurst parameter of the packet stream generated varies.

The PMPP model was simulated on OPNET and the IDC was computed. Figures 3.3 and 3.4 compare the IDC curves obtained from equation 3.8 against simulation for values of $\lambda_1 = 100$, $\lambda_2 = 120$ and $\alpha = 1.3$ and 1.5 respectively. As seen from these curves, the results obtained from simulation agree fairly with the theoretical results. The IDC plot for $\lambda_1 = 100$ and $\lambda_2 = 120$ and for various values of α of the Pareto distributed sojourn times is shown in Figure 3.5. As seen, the linear characteristics of IDC in a log-log plot suggest that the model exhibits self-similar characteristics. Also, given in the figure are the Hurst parameter H estimated from the slope of the IDC. It is seen that the Hurst parameter so obtained satisfies the relation 3.9, quite fairly.

The Variance time curves for the PMPP were also obtained from simulation. The variance time curve is obtained by computing the variance of the arithmeic mean of the count process. i.e., if as before $X = (X_1, X_2, X_3, \ldots)$ denote the number of packets generated by the process in succesive frames, let $X^{(m)} = (X_k^{(m)} \; ; k = 1, 2, 3, \ldots)$ denote a new (aggregated) time series obtained by averaging the original series X over non-overlapping blocks of size m.i.e, for each $m = 1, 2, 3, \ldots$, $X^{(m)}$ is given by $X_k^{(m)} = 1/m(X_{m(k-1)} + \ldots + X_{km})$. Then plotting Variance($X^{(m)}$ against various values of m gives the variance time plots. While for conventional models the variance of the sample mean is inversely proportional to the sample size, for long-range dependent processes, it decreases as a fractional power of sample size (i.e.,it decreases more slowly than the reciprocal of the sample size). Hence in the case of long-range dependent processes

$$\mathrm{Var}(X^{(m)}) = a_1 m^{-\beta} \quad \text{with} \quad 0 < \beta < 1$$

where a_1 is a constant. When the variance time curve is plotted in a log-log scale, the slope β is related to the Hurst parameter, H, by the relation

$$H = 1 - |\beta|/2 \tag{3.10}$$

Figure 3.6 shows the variance time plot for various values of α with $\lambda_1 = 100$ and $\lambda_2 = 120$) obtained from simulation. The linear behaviour of Variance time curve in a log-log plot shows the presence of slowly decaying variances. The Hurst parameter estimated from these graphs also indicate that the relation 3.9 holds well.

Hence the PMPP is efficient in characterizing the fractal nature of the data traffic.

Also the proposed model captures the presence of the long and short bursts inherent in data traffic. This model is easy to simulate when compared to other methods for generation of self-similar traffic. Hence this method may be used to generate a self-similar traffic stream with H = (3 - α)/2. The other 2 parameters of the model namely λ_1 and λ_2 are to be matched with that of the aggregate traffic stream by a suitable matching technique.

The PMPP model was used to match an actual traffic trace from Bellcore Etherne t traffic data from October 1989. The traffic file is available via anonymous FTP from *flash.bellcore.com.* The Hurst parameter was estimated from a log-log plot of the IDC of the trace, to be 0.8202. From the Hurst parameter H of the trace the parameter α was determined to be $\alpha = 1.3596$ using relation 3.9. The parameters λ_1 and λ_2 were obtained by matching the average number of packets generated from the traffic data and IDC(1) (i.e, IDC at lag 1) of the data with equations 5.1 and 3.8 respectively. The est imated values of λ_1 and λ_2 are $\lambda_1 = 2.8565$ packets/ $10ms$ and $\lambda_2 = 6.8235$ packets/$10ms$. The PMPP model was simulated using these values for the parameters λ_1, λ_2 and α and the IDC was plotted. The IDC plot obtained from simulations (circled plot) is compared against the original plot (starred plot) in figure 3.7. Also shown in the figure is the plot of IDC (bold line) obtained by using the equation 3.8. As can be observed from the figure the plots obtained from simulation closely follow the IDC plot obtained from the experimental data.

3.3 Aggregate traffic model

The aggregate traffic model that we propose is shown in Figure 3.8. We approximate the aggregate of voice, video and data sources each by a two stated doubly stochastic Poisson process. As has already been explained the aggregate packet arrival process from Voice sources and Video sources may each be approximated by a 2-state MMPP. The data traffic may be modelled by the 2-state switched Poisson process proposed in the previous section. The resulting model is the superposition of three 2-state switched Poisson processes, giving rise to an eight state switched Poisson process as shown in Figure 3.9. The model is simple and easy to simulate.

If the data traffic is also approximated by a 2 state MMPP, then by the property that the superposition of MMPP is again an MMPP, we obtain an eight state MMPP. However, this model is not accurate in the sense that it does not capture the long term correlations of data. However, if the PMPP is selected for data traffic, we obtain an eight state switched Poisson process, which may not simplify into a simple form as in the case of MMPP.

3.4 Performance evaluation of a G/D/1 queue

In this section we present the queueing performance of the aggregate traffic model proposed. The queueing system that we consider here is a G/D/1 queueing system. The server serves a fixed number of packets per second, as is the case in an ATM multiplexer. The arrival process to the queue, that we study are a PMPP or a MMPP or the aggregate traffic, which is constituted by 2 MMPPs one each for voice and video and a PMPP for data. We present the simulation results, i.e., the survivor function of the queue length distributions for all these cases.

First we investigate the queueing performance of the long-range dependent model proposed in this research - the PMPP model. The parameters for the PMPP that we consider are $\lambda_1 = 200$ and $\lambda_2 = 250$. The value of α determines the Hurst parameter of the aggregate stream.

Figure 3.10 shows the survivor function of queue length for various Hurst parameters, for a loading of $\rho = 0.9$. As can be seen from the graph the queue length distributions appear to be Weibullian or "stretched exponential". Such behaviour is due to the long range dependent correlations exhibited by this model. In [73], the

G/D/1 queue fed by FBM (fractal brownian motion) traffic was also shown to result in a Weibull distribution of the form

$$P(X > x) \approx \exp -\gamma x^(2 - 2H) \qquad (3.11)$$

Also, similar results were obtained in [74] by aggregating many ON/OFF sources with heavy tailed sojourn times. Hence higher the Hurst parameter of the traffic stream, more heavy tailed the queue length distribution is. This indicates that for a long range dependent stream with a high Hurst parameter may suffer more loss. Also, increasing the buffer size does not result in a significant reduction in the loss probabilty. Figure 3.11 plots the Probability of loss against the Hurst parameter, for a finite buffer size of K = 150. It can be noted that higher the Hurst parameter of the input stream, higher the loss.

In order to compare the performance of a MMPP model in the same scenario, we simulated the equivalent MMPP for the PMPP under consideration. i.e, we choose λ_1 and λ_2 to be the same as the ones before that is $\lambda_1 = 200$ and $\lambda_2 = 250$. Also, as in the case of PMPP, the sojourn times in both the states are identical, but exponentially distributed. The average sojourn time in each state is equated to the average sojourn time in the corre sponding PMPP, giving us a fair basis of comparison of the 2 models. Figure 3.12 compares the queuing behavior of a PMPP with H = 0.95 and its equivalent MMPP model. As seen from the figure the residual function of the queue decays at a faster rate in the case of the MMPP arrival process. This is due to the fact that the MMPP model is a short range dependent model.

In the PMPP model the two rates λ_1 and λ_2 may correspond to the long and short burst rates inherent in data traffic. The difference between these two rates λ_1 and λ_2, $\delta\lambda$ may intuitively be thought as representing the burstiness of the traffic stream. The previous set of results with $\lambda_1 = 200$ and $\lambda_2 = 250$ had a $\delta\lambda$ of 50. Figure 3.13 plots the queue length distributions for the case when $\delta\lambda = 100$. The load is 0.9 as before, however now the $\lambda_1 = 175$ and $\lambda_2 = 275$. As seen from the figure the residual function of queue length is more heavy tailed than in the case of $\delta\lambda = 50$. Hence with an increase $\delta\lambda$, we see a more burstier traffic stream.

Now we investigate the G/D/1 queue by feeding it with the aggregate traffic consisting of voice, video and data. As discussed in the previous section, voice and video traffic are modelled by a MMPP, while data is modelled by a PMPP. To illustrate the effect of the long term dependent data on the aggregate traffic, we also consider additionally an aggregate model where data is model by a MMPP. The difference in the queueing performance of both the aggregate model illustrates the impact of long term dependent correlations. The parameter values used are as follows:

- for voice (MMPP): λ_1 = 28 pkts/ms ; λ_2 = 41 pkts/ms; α_1 = 0.000956 ms^{-1}; α_2 = 0.0250 ms^{-1}
- for video (MMPP): λ_3 = 24 pkts/ms ; λ_4 = 39 pkts/ms; α_3 = 0.0087 ms^{-1}; α_4 = 0.0483 ms^{-1}
- for data (PMPP): λ_5 = 10 pkts/ms; λ_6 = 38 pkts/ms; $alpha_5 = 1.4$; $\alpha_6 = 1.4$.

where for the MMPPs the α stand for the transition rate from one state to another and for the PMPP the α is the parameter of the Pareto distribution. The above traffic composition has a ratio of 1:1:1. i.e., togehther voice and video are twice that of the data traffic. The server serves at the rate of 100 pkts/ms, thus giving a ρ of 0.8.

Another simulation was run with the same parameters for voice and video and a MMPP with the following parameters for data: λ_5 = 10 pkts/ms; λ_6 = 38 pkts/ms; $\alpha_5 = \alpha_6 = 0.2857$. Figure 3.14 shows the survivor function distribution for queue length. As seen there the aggregate model using PMPP for data has a similar behaviour as the other aggregate model using MMPP. This is due to the fact that the volume of the long range dependent traffic is less when compared with the others.

To verify if the converse is true, we simulated a traffic composition consisting twice as much data as voice and video. The same parameters were used for voice and video, while for data (PMPP), the following values were used λ_5 = 60 pkts/ms; λ_6 = 150 pkts/ms and $\alpha_5 = \alpha_6 = 1.4$. Now the service rate was increased to 200 pkts/ms. Again in order for a comparison to be made with a MMPP, an aggregate model using

MMPP data was used. The following parameters were used for the MMPP used for data: $\lambda_5 = 60$ pkts/ms; $\lambda_6 = 150$ pkts/ms and $\alpha_5 = \alpha_6 = 0.2857$. This gives a combination of 2:1 for data vs voice and video.

Figure 3.15 shows the queue length distributions for this case. It can be clearly noted that the model using PMPP has a heay- tailed distribution than the other model using MMPP. Hence the traffic composition also plays a role in the determination of how the net traffic will behave.

To summarise the queueing performance of the traffic model under consideration: Results indicate that the PMPP has a queueing behaviour similar to that of the long range dependent models reported in the literature. Also, the results for the aggregate traffic indicate that the composition of the traffic is also important in engineering the length of the buffers at the multiplexers. Finally, the finding of this project opens up a new avenue of research, which is the analysis of queues fed by PMPP arrival process.

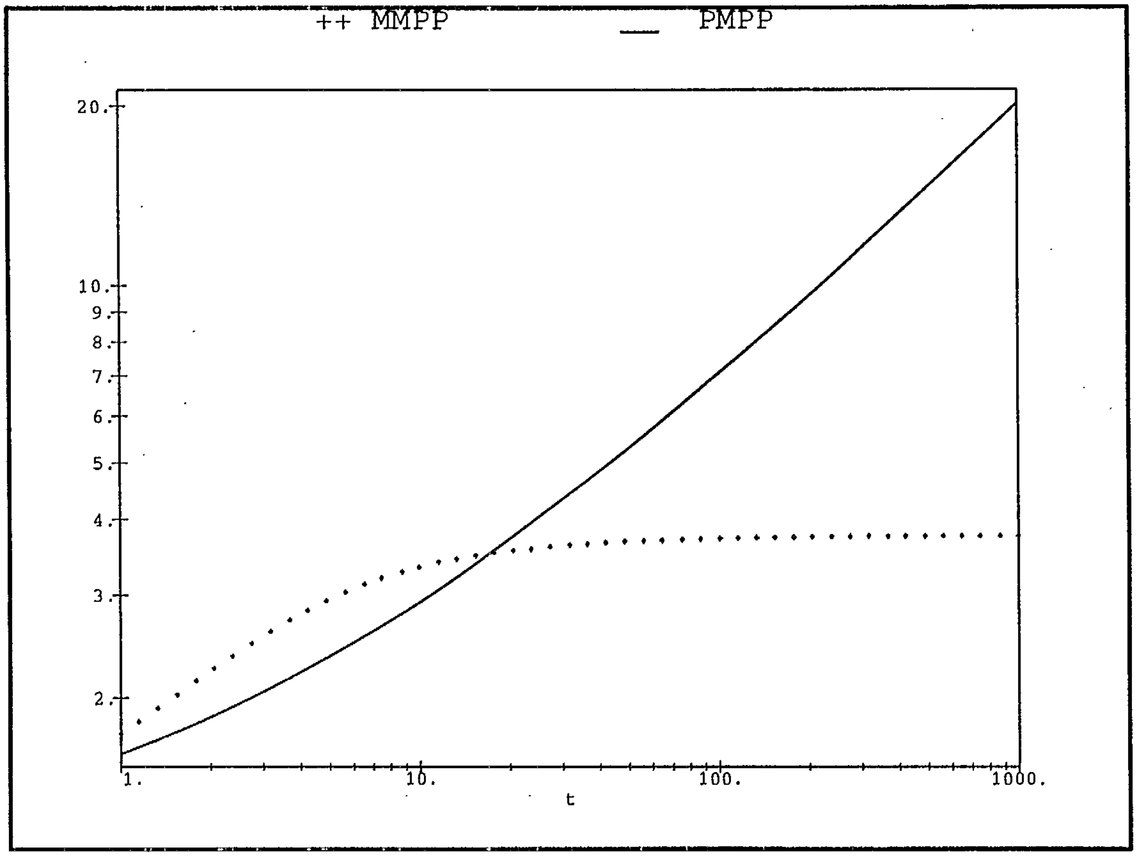


Figure 3.1: IDC curves of MMPP and PMPP plotted in a log-log scale

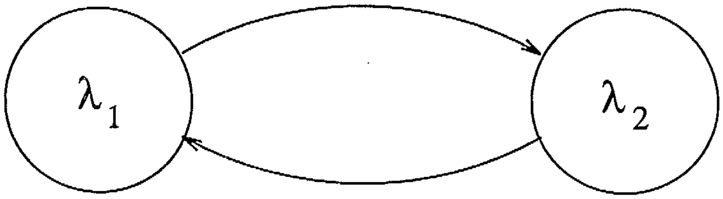


Figure 3.2: PMPP model

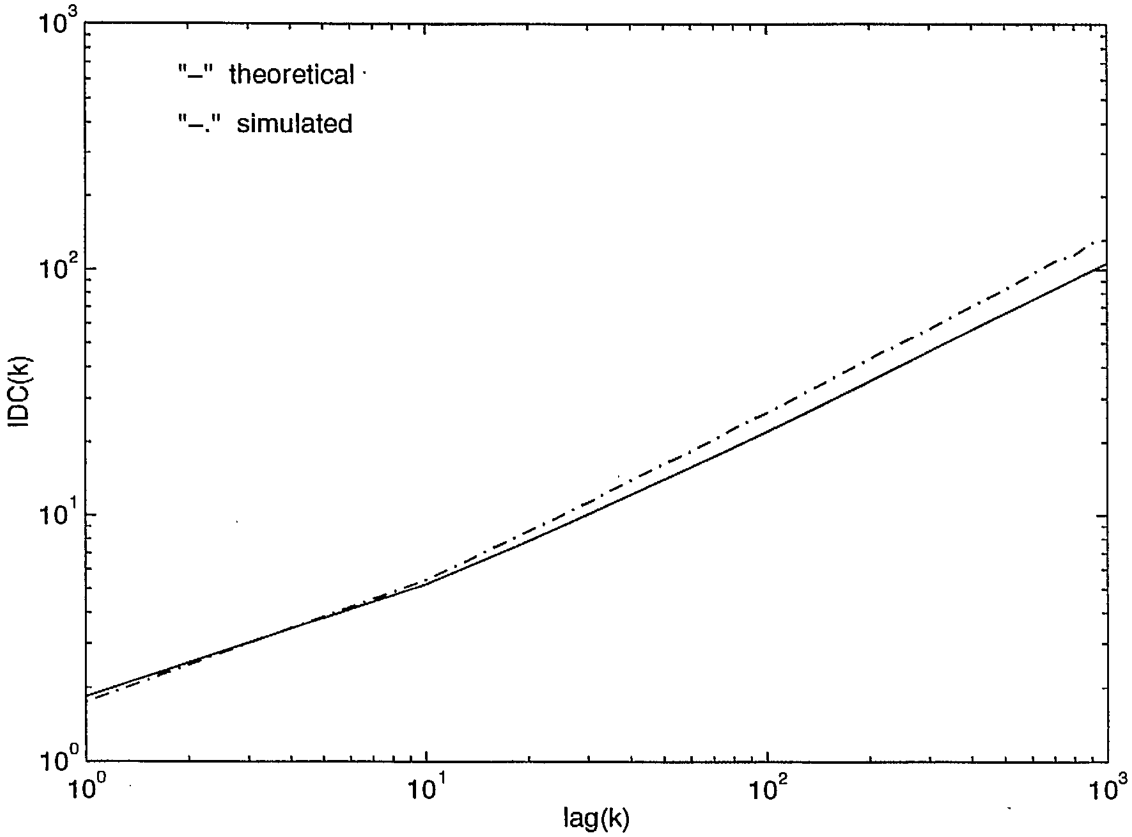


Figure 3.3: Simulated and theoretical curves of IDC, for PMPP, with $\lambda_1 = 100$, $\lambda_2 = 120$ and $\alpha = 1.3$

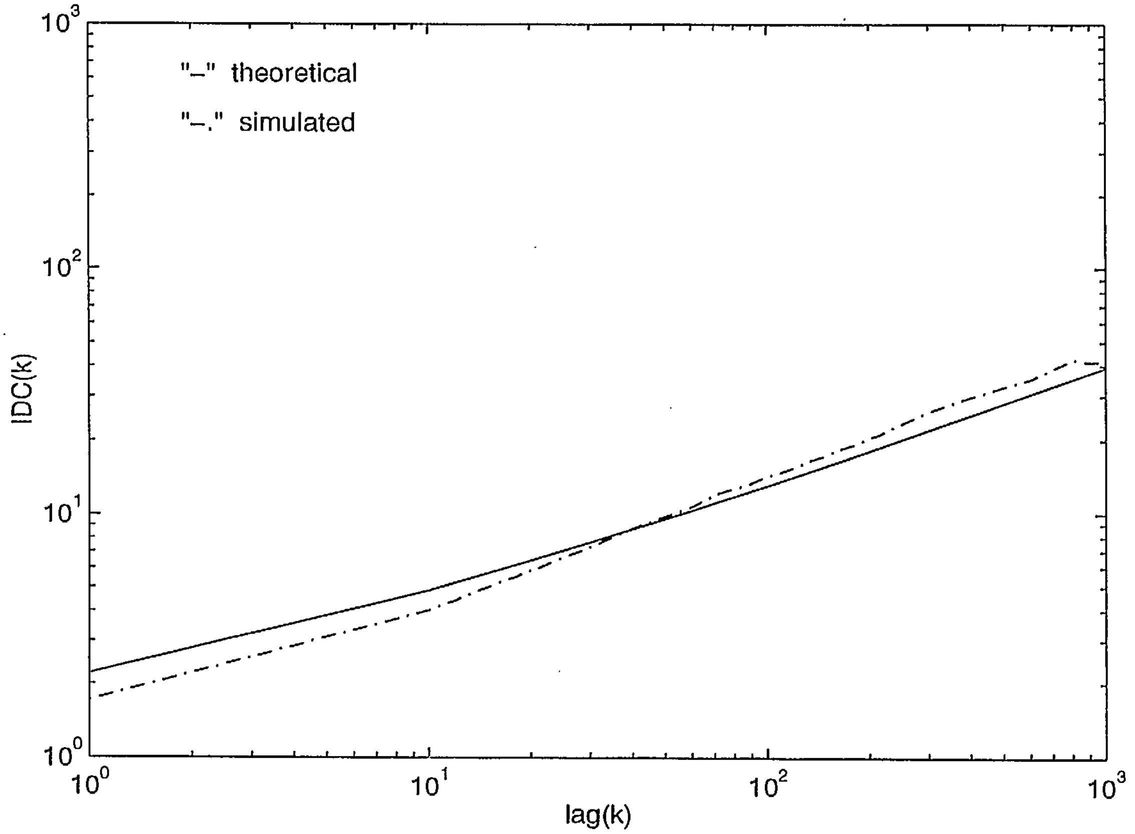


Figure 3.4: Simulated and theoretical curves of IDC, for PMPP, with $\lambda_1 = 100$,$\lambda_2 = 120$ and $\alpha = 1.5$

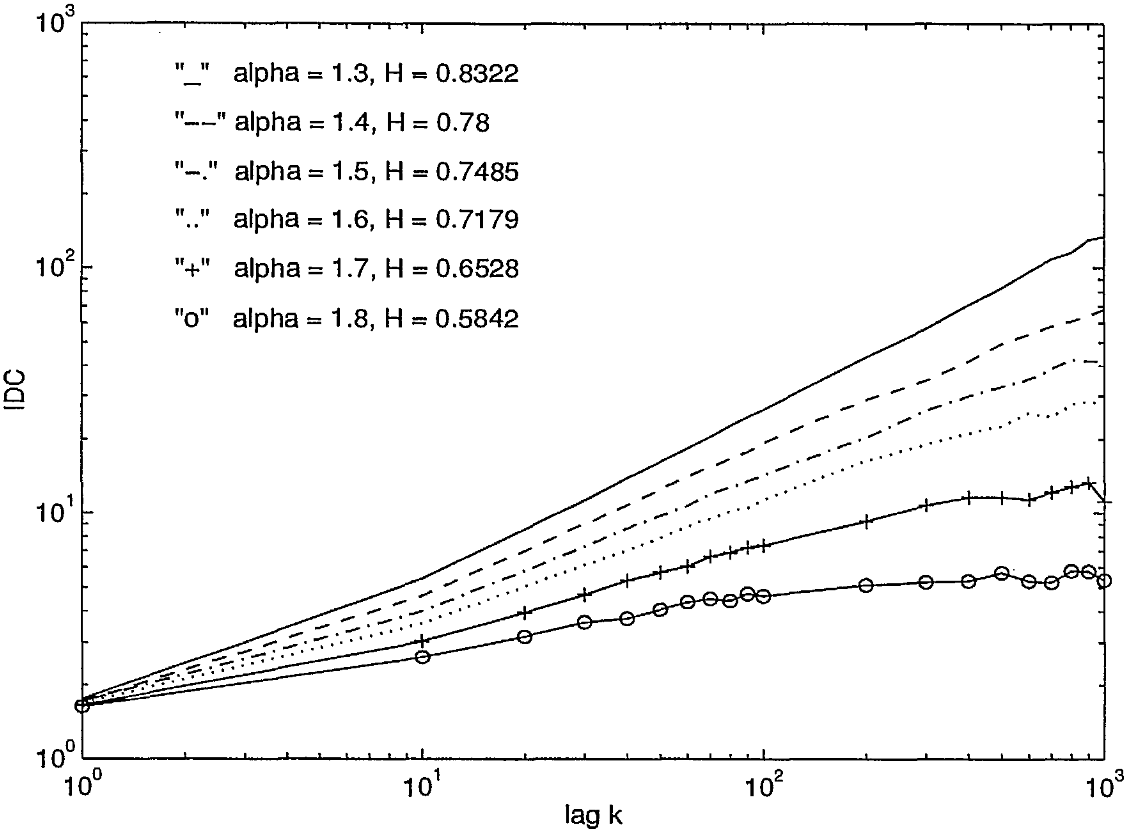


Figure 3.5: IDC curves for various values of α, with $\lambda_1 = 100$, $\lambda_2 = 120$

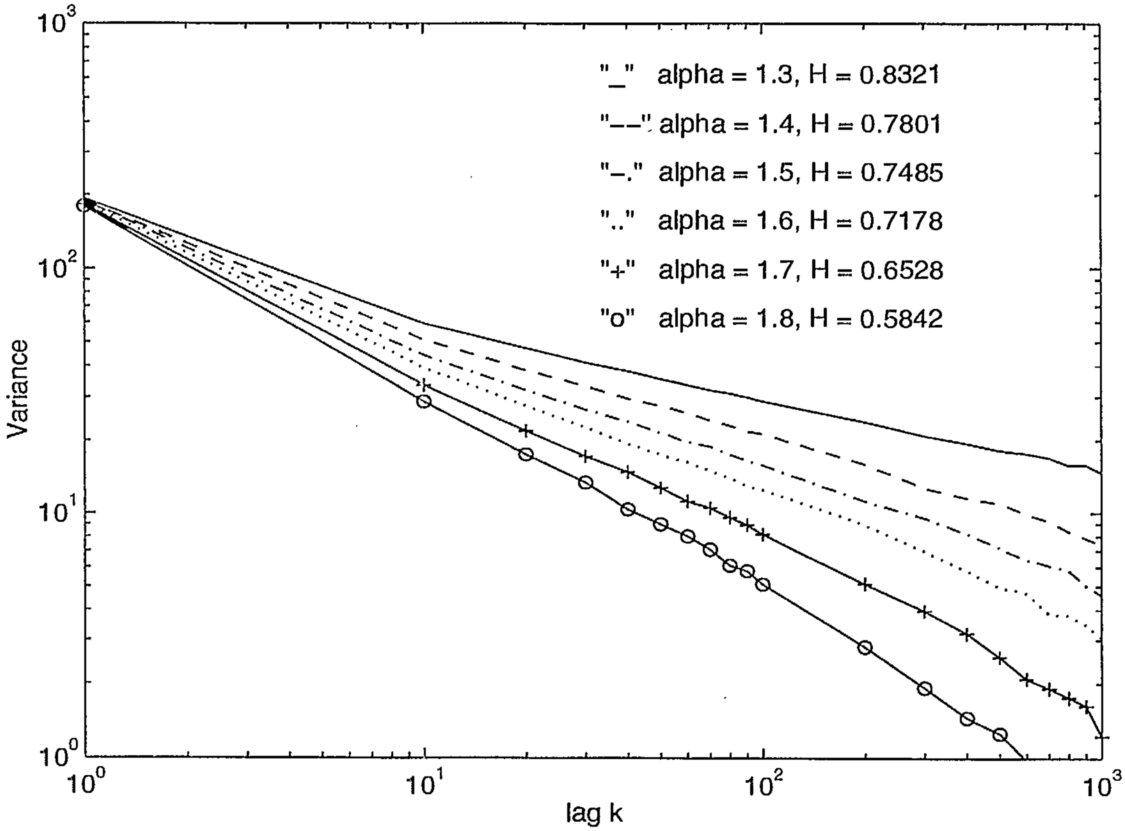


Figure 3.6: Variance time curves for various values of α, with $\lambda_1 = 100$, $\lambda_2 = 120$

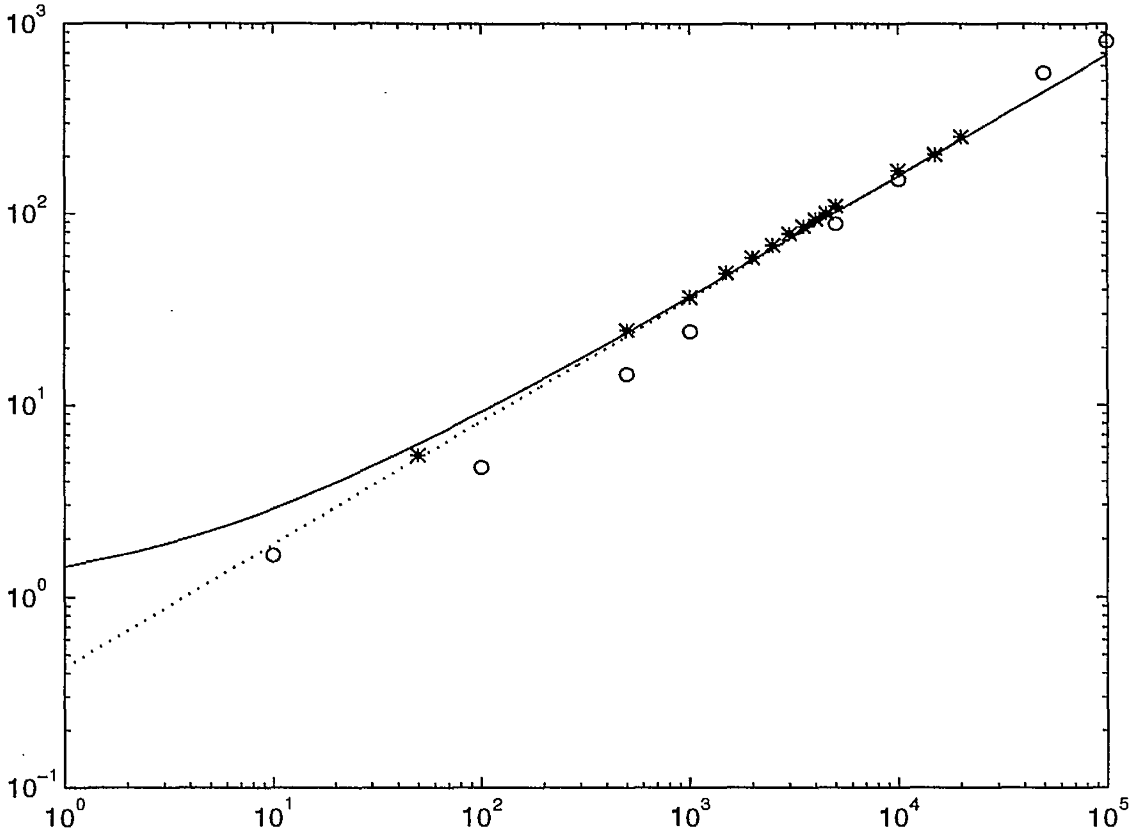


Figure 3.7: IDC plots of the traffic trace obtained from Bellcore traffic data compared against the plots obtained from the simulation of PMPP model.

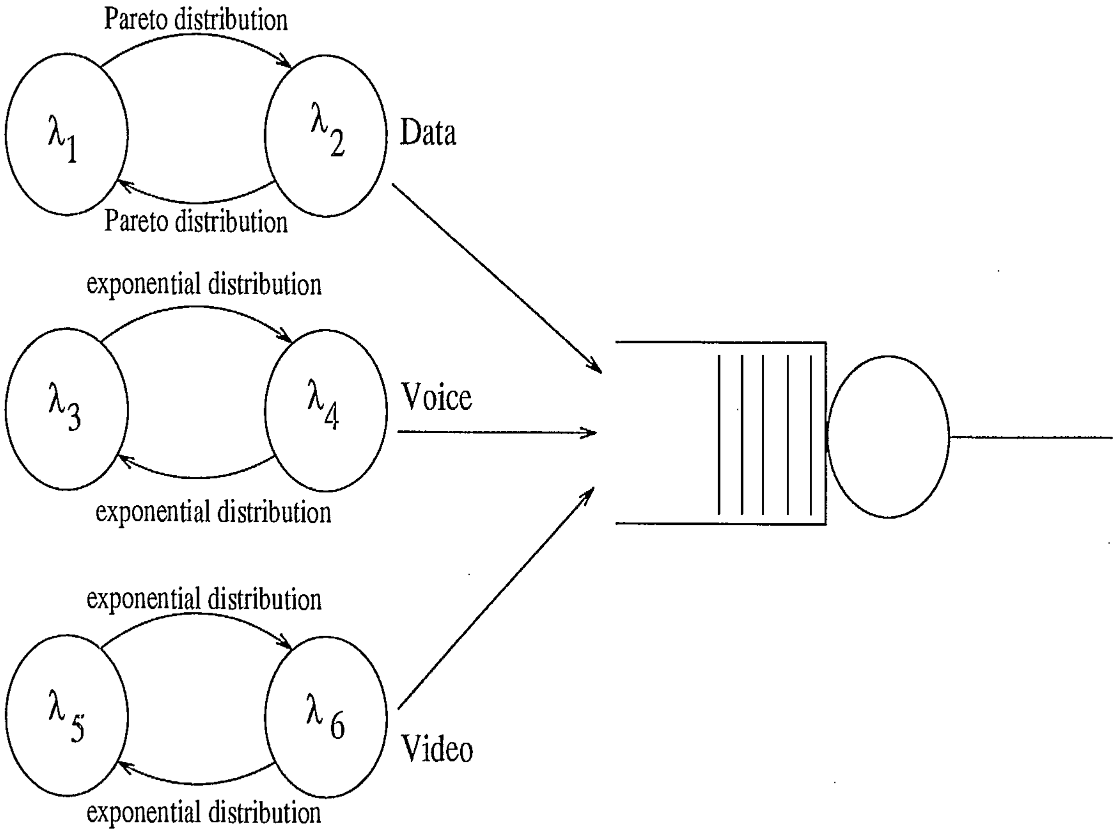


Figure 3.8: Aggregate traffic model

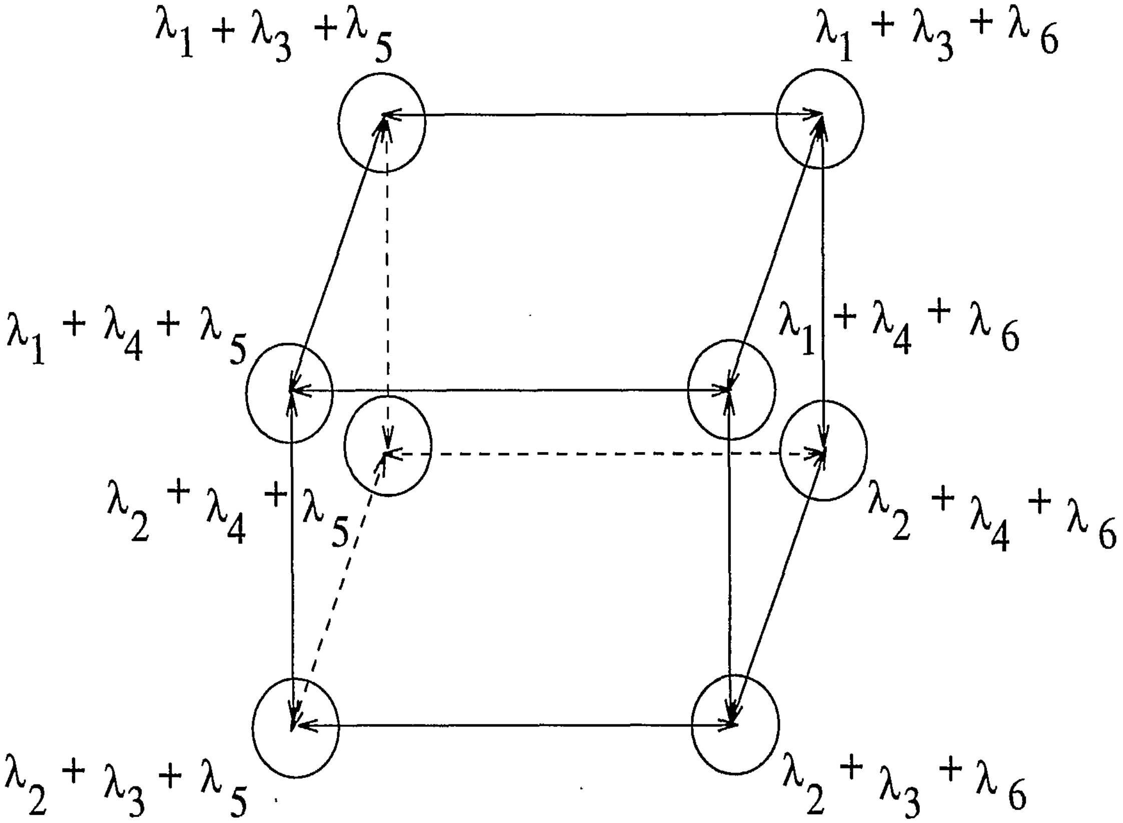

$\lambda_1 + \lambda_3 + \lambda_5$
$\lambda_1 + \lambda_3 + \lambda_6$
$\lambda_1 + \lambda_4 + \lambda_5$
$\lambda_1 + \lambda_4 + \lambda_6$
$\lambda_2 + \lambda_4 + \lambda_5$
$\lambda_2 + \lambda_4 + \lambda_6$
$\lambda_2 + \lambda_3 + \lambda_5$
$\lambda_2 + \lambda_3 + \lambda_6$

Figure 3.9: Superposed process

Survivor function of queue length

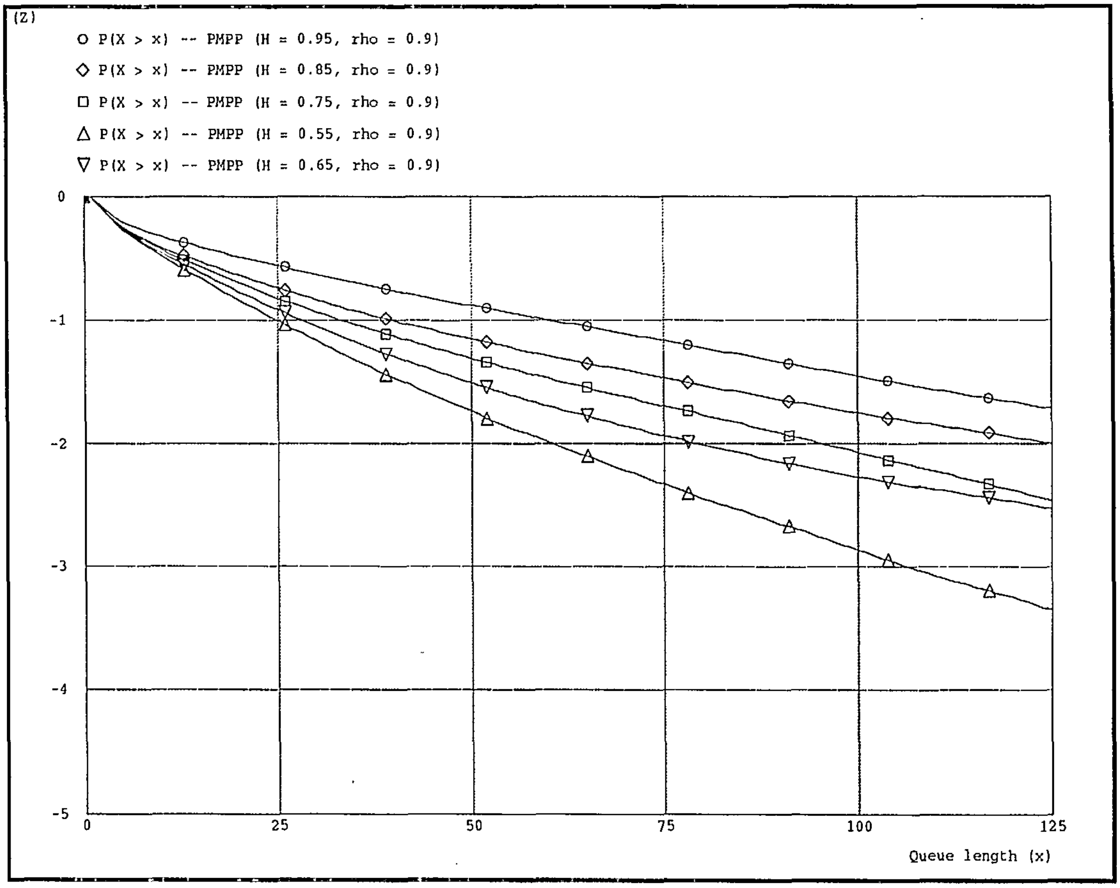


Figure 3.10: Survivor function of Queue length of PMPP arrival in a semi-log scale

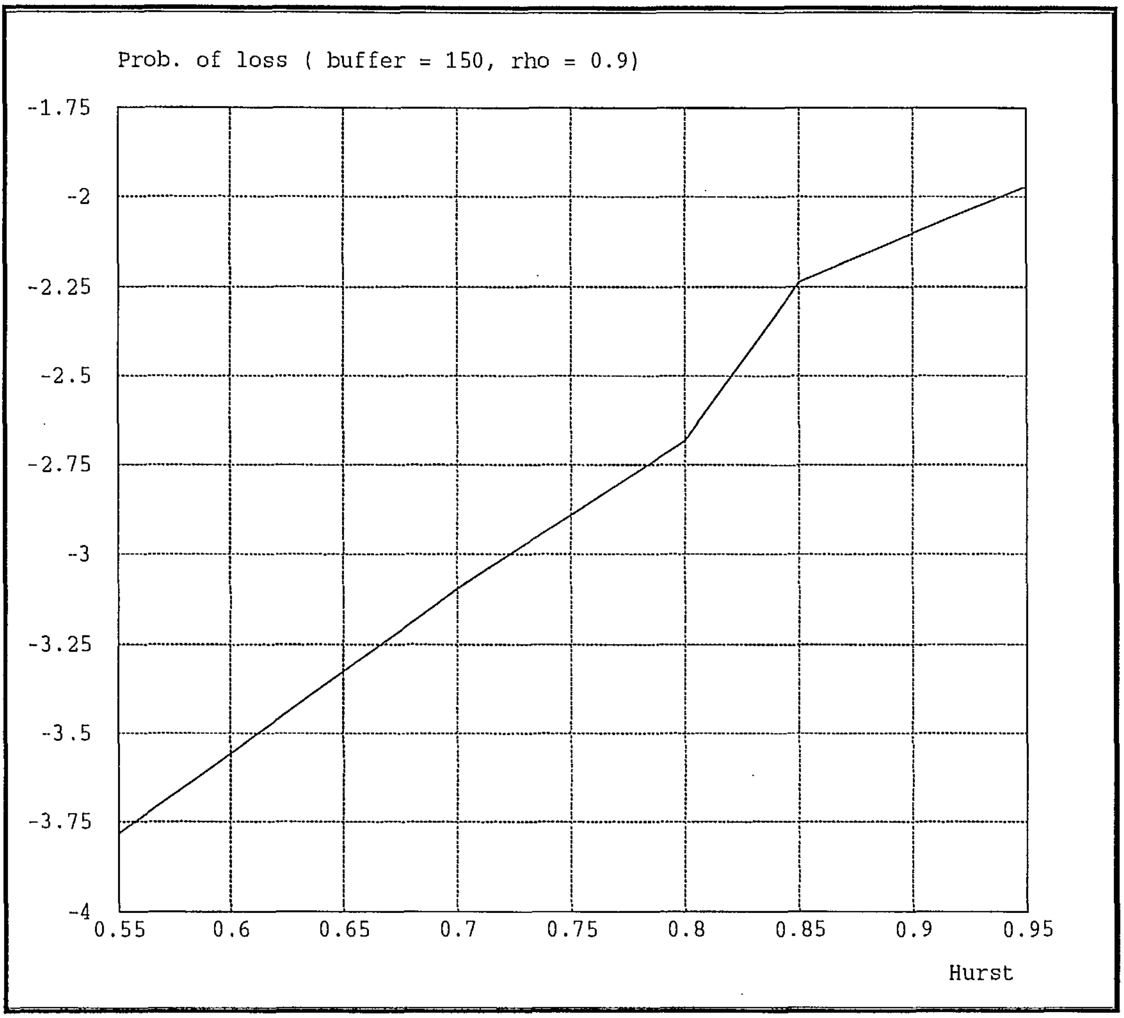


Figure 3.11: Probability of loss as a function of Hurst parameter in a semi-log scale

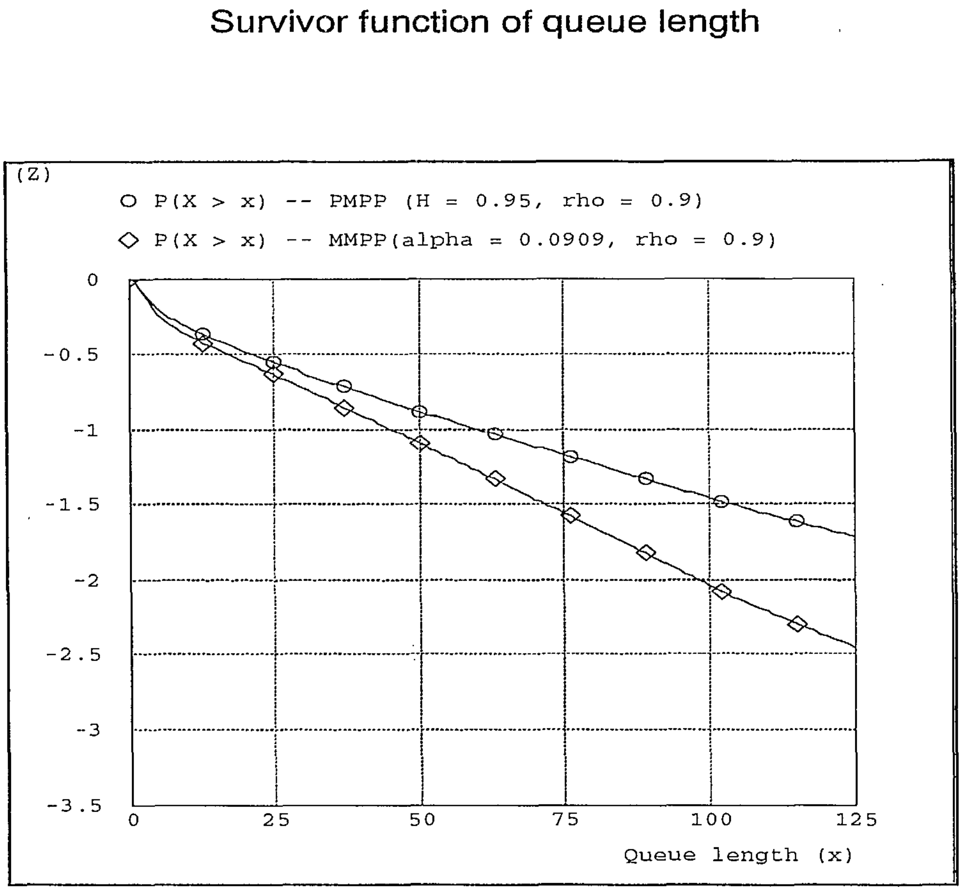


Figure 3.12: Survivor function of Queue length of PMPP and MMPP arrival in a semi-log scale

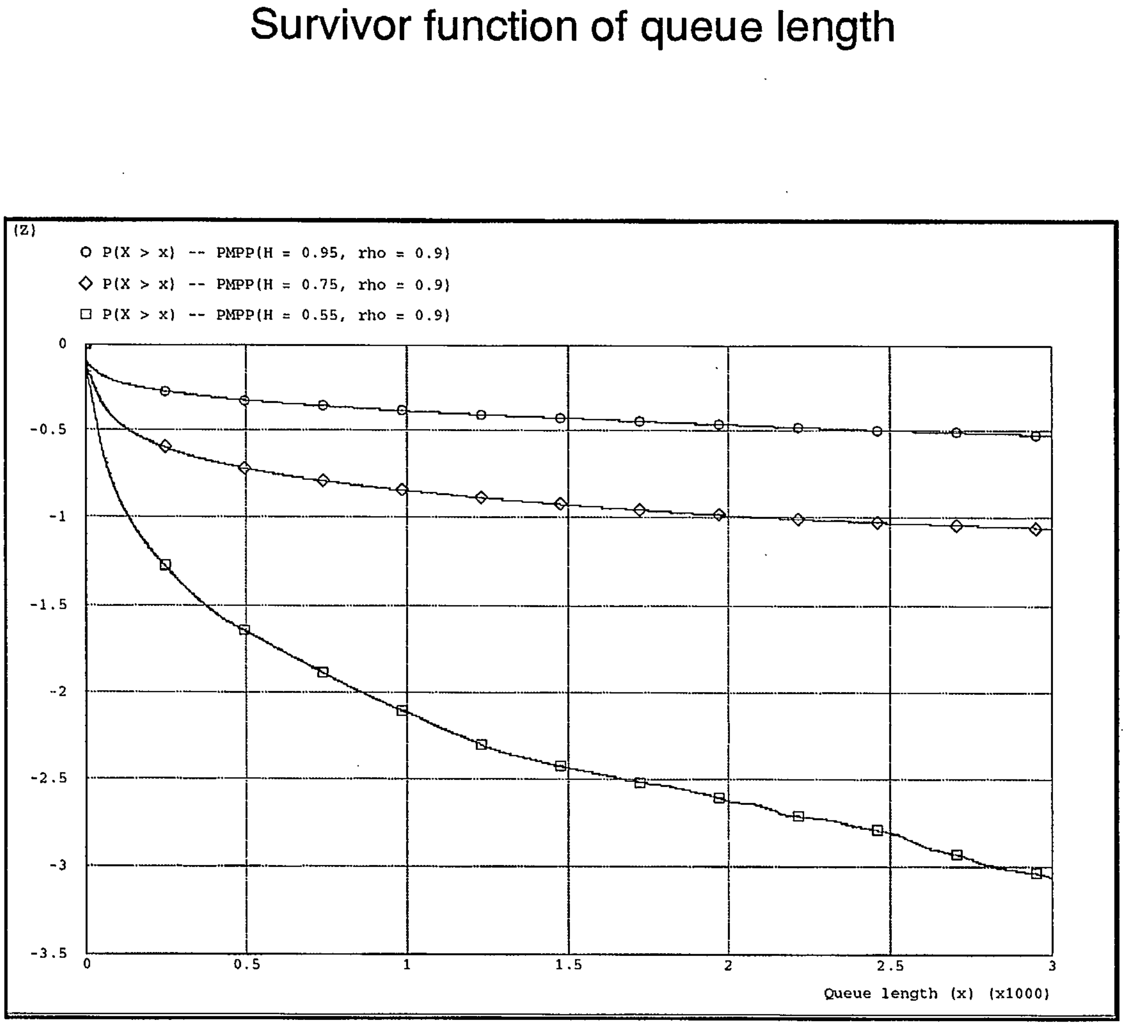


Figure 3.13: Survivor function of Queue length of PMPP with $\delta\lambda = 100$ and $\rho = 0.9$, in a semi-log scale

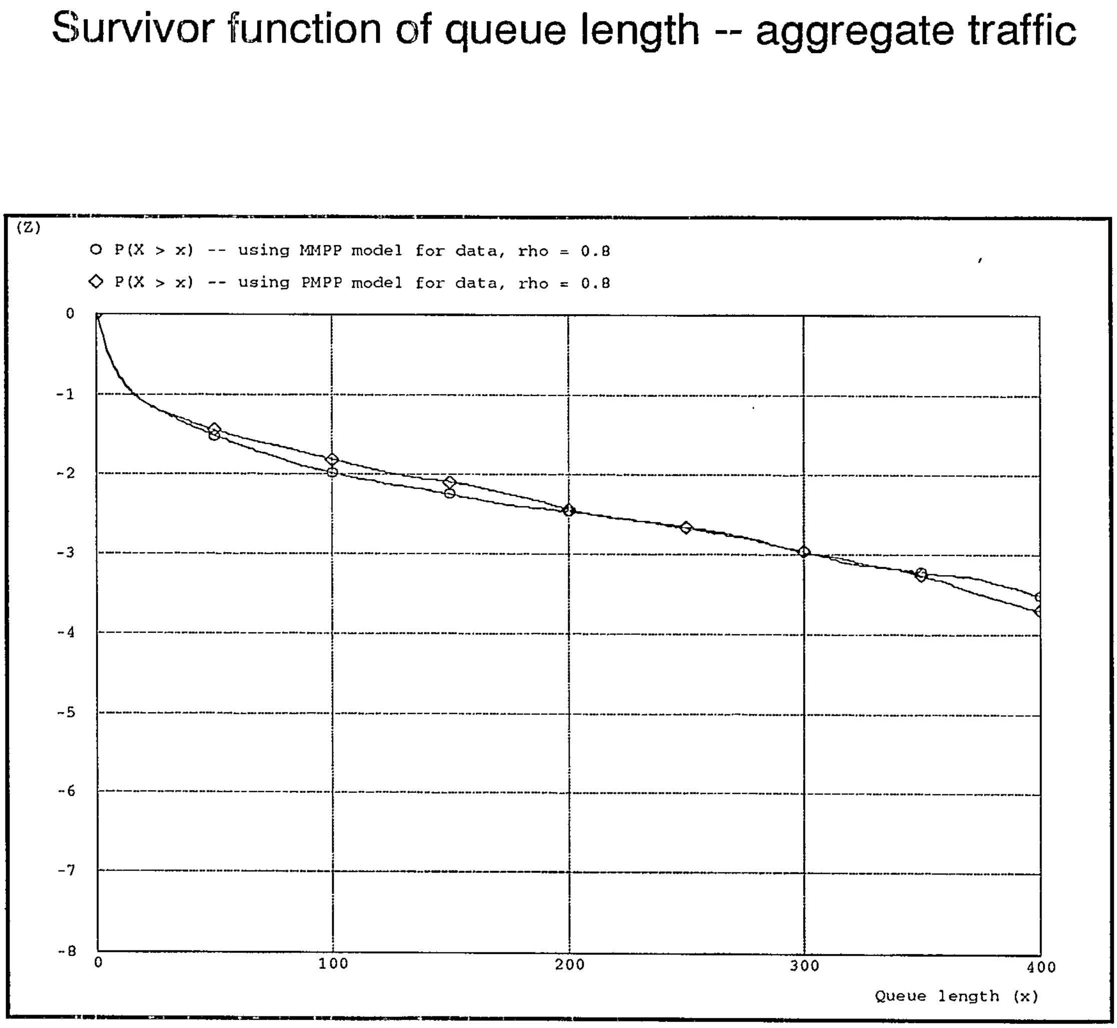
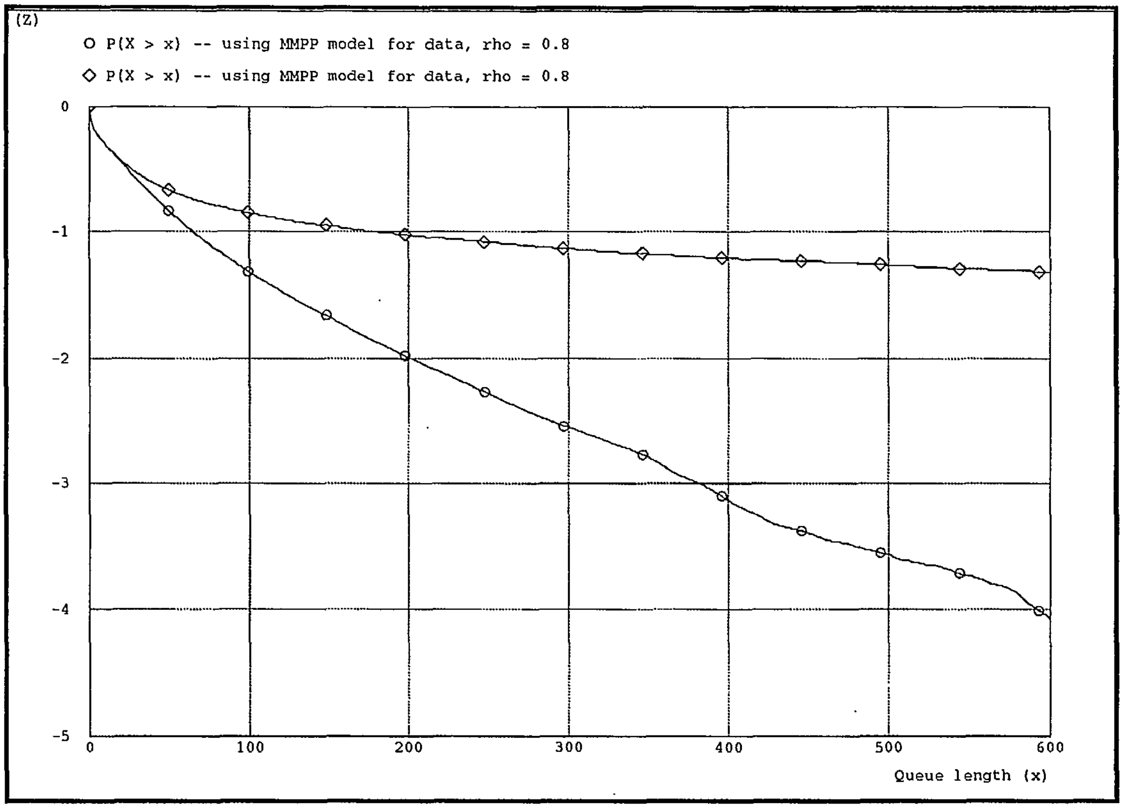


Figure 3.14: Survivor function of Queue length for aggregate traffic with 1:1:1 composition



Survivor function of queue length -- aggregate traffic

Figure 3.15: Survivor function of Queue length for aggregate traffic with 2:1 (data vs. voice + video) composition

Chapter 4

Conclusion

4.1 Summary of completed tasks

This report presented the results of the study conducted during this project on traffic modeling. Following aspects have been addressed by this study

- The various traffic models proposed in the literature for voice, video and data traffic were surveyed and classified.

- A traffic generator comprising of standard traffic models like on/off model, MMPP, etc., was built on OPNET.

- A new model for the aggregate packet traffic was proposed. This model was simulated on OPNET and studied.

- The queueing performance of aggregate multi-media traffic was studied by simulating the aggregate model on OPNET.

- The performance of a CFDAMA (Combined free/demand assignment multiple access) protocol in a multi-media SATCOM system was studied, using the proposed aggregate model.

4.2 Results and conclusions

Various models proposed for the constituents of the multi-media traffic were surveyed. The models proposed for aggregate traffic in the literature did not account for the fact that in the multi-media environment, fractal (long-range dependent) traffic co-exists with non-fractal traffic. The aggregate model consisting of doubly stochastic Poisson processes (MMPP and PMPP), proposed in this research, accounts for both the fractal and non-fractal traffic. Also, the simulation results of the new model for data-traffic, PMPP (Pareto modulated Poisson process) showed that it had an IDC that increased with lag and a variance-time curve that decreased with lag (on a semi-log plot), characteristics reminiscent of long-range dependence. These properties were also analytically verified.

The results of the performance study indicate that the PMPP has a queueing behaviour similar to that of the long-range dependent models proposed in the literature, i.e., the survivor function of queue length has a "stretched exponential" behaviour. The performance study of the aggregate traffic model indicates that the queueing behaviour is affected by the ratio of the long-range dependent traffic in the aggregate traffic mix. Thus implying that the composition of aggregate traffic is also important in engineering the buffers at the statistical multiplexers of multi-media traffic.

The aggregate traffic model proposed is easy to simulate and forms a part of the traffic generator developed during this study period.

4.3 Suggestions for future study

The findings of this project opens up a new avenue of research: the analytical techniques for evaluating the queueing performance of the aggregate model. This in itself is a complex task, given the non-Markovian nature of the processes involved.

Appendix

Derivation of Index of dispersion of counts

Consider the two state, random hazard doubly stochastic Poisson process as in [69]. Let the Poisson rates in states 1 and 2 be λ_1 and λ_2 respectively. Let

$f_1(t)$ be the p.d.f. of sojourn time in state 1.

ν_1 be the average sojourn time in state 1.

$F_1(t)$ be the cumulative distribution function of sojourn time in state 1.

$R_1(t)$ be the survivor function of sojourn time in state 1.

$f_1^*(s)$, $F_1^*(s)$, $R_1^*(s)$ be the Laplace transform of $f_1(t)$, $F_1(t)$ and $R_1(t)$ respectively.

Let $f_2(t)$, $F_2(t)$, $R_2(t)$, ν_2, $f_2^*(s)$, $F_2^*(s)$, $R_2^*(s)$ be the corresponding quantities for state 2.

Let $N(t)$ be the number of arrivals in time t. From 3.7 the general form of the Laplace transform of the probability generating function of $N(t)$ is given by

$$\begin{aligned}
\phi^*(z,s) &= \frac{1}{\nu_1+\nu_2}\left(\frac{\nu_1}{s+\lambda_1(1-z)}+\frac{\nu_2}{s+\lambda_2(1-z)}\right) \\
&\quad -\frac{(\lambda_1-\lambda_2)^2}{\nu_1+\nu_2}\left(\frac{(1-z)^2}{(s+\lambda_1(1-z))(s+\lambda_2(1-z))}\right) \\
&\quad \text{x}\left(\frac{R_1^*(s+\lambda_1(1-z))R_2^*(s+\lambda_2(1-z))}{(1-f_1^*(s+\lambda_1(1-z)f_2^*(s+\lambda_2(1-z)))}\right)
\end{aligned}$$

Differentiating the above expression partially with respect to z and setting z = 1, gives the Laplace transform of the average number of packets generated.

$$\mathcal{L}\{E[N(t)]\} = \frac{\lambda_1\nu_1 + \lambda_2\nu_2}{(\nu_1+\nu_2)s^2}$$

Inverting the above equation, the mean of the counting process is obtained as

$$E[N(t)]\} = \left(\frac{\lambda_1\nu_1 + \lambda_2\nu_2}{\nu_1+\nu_2}\right) t \qquad (5.1)$$

The Laplace transform of variance of $N(t)$ can be obtained by differentiating 3.7 twice and setting z = 1.

$$\begin{aligned}\mathcal{L}\{Var[N(t)]\} &= \frac{\lambda_1\nu_1 + \lambda_2\nu_2}{(\nu_1+\nu_2)s^2} + \frac{2(\lambda_1-\lambda_2)^2}{(\nu_1+\nu_2)^2}\frac{\nu_1\nu_2}{s^2} \\ & \quad \text{x}\left[\frac{1}{s} - \left(\frac{\nu_1+\nu_2}{\nu_1\nu_2}\right)\left(\frac{R_1 * (s)R_2^*(s)}{1-f_1^*(s)f_2^*(s)}\right)\right] \qquad (5.2)\end{aligned}$$

An explicit equation for the variance may be obtained by inverting the above equation depending on the sojourn time densities $f_1(t)$ and $f_2(t)$.

MMPP

In the case of MMPP (Markov Modulated Poisson Process),

$$\begin{array}{ll} f_1(t) = r_1 \exp^{-r_1 t} & f_2(t) = r_2 \exp^{-r_2 t} \\ R_1(t) = \exp^{-r_1 t} & R_2(t) = \exp^{-r_2 t} \\ \nu_1 = 1/r_1 & \nu_2 = 1/r_2 \end{array}$$

The corresponding Laplace transforms are

$$\begin{array}{ll} f_1^*(s) = \frac{r_1}{s=r_1} & f_2^*(s) = \frac{r_2}{s+r_2} \\ R_1^*(s) = \frac{1}{s+r_1} & R_2^*(s) = \frac{1}{s+r_2} \end{array}$$

Substituting the above in equation 5.2, we have the Laplace transform of variance as

$$\mathcal{L}\{Var[N(t)]\} = \frac{\lambda_1 r_2 + \lambda_2 r_1}{(r_1+r_2)s^2} + \frac{2(\lambda_1-\lambda_2)^2}{(r_1+r_2)^2} r_1 r_2$$

$$\text{x}\left[\frac{1}{s^3} - \frac{r_1+r_2}{s^3(s+(r_1+r_2))}\right]$$

Inverting we have

$$\begin{aligned} Var[N(t)] &= \left(\frac{\lambda_1 r_2 + \lambda_2 r_1}{r_1+r_2}\right)t + \frac{2(\lambda_1-\lambda_2)^2}{(r_1+r_2)^3} r_1 r_2 t \\ &\quad - \frac{2(\lambda_1-\lambda_2)^2}{(r_1+r_2)^4} r_1 r_2 (1 - \exp^{-(r_1+r_2)t}) \end{aligned} \quad (5.3)$$

From 5.1 and 5.3 the IDC of 2-state MMPP may be obtained as

$$\begin{aligned} I(t) &= 1 + \frac{2(\lambda_1-\lambda_2)^2 r_1 r_2}{(r_1+r_2)^2(\lambda_1 r_2 + \lambda_2 r_1)} \\ &\quad - \frac{2(\lambda_1-\lambda_2)^2 r_1 r_2}{(r_1+r_2)^3(\lambda_1 r_2 + \lambda_2 r_1)t}(1 - \exp^{(-(r_1+r_2)t}) \end{aligned}$$

PMPP

For PMPP (Pareto Modulated Poisson process)

$$\begin{array}{lll} f_1(t) = f_2(t) = \alpha t^{-(\alpha+1)} & 1 < \alpha < 2 & t \geq 1 \\ R_1(t) = R_2(t) = t^{-\alpha} & 1 < \alpha < 2 & t \geq 1 \\ F_1(t) = F_2(t) = 1 - t^{-\alpha} & 1 < \alpha < 2 & t \geq 1 \\ \nu_1 = \nu_2 = \frac{\alpha}{\alpha-1} & & \end{array}$$

Since here, $t \geq 1$ the Laplace transforms of these functions have to be computed from definition as follows. Let

$$R_1^*(s) = R_2^*(s) = \int_1^\infty \exp^{-st} t^{-\alpha} dt = g(s,\alpha) \quad (5.4)$$

Integrating by parts, we have the following recursion,

$$g(s,\alpha) = \frac{1}{s}\left[\exp^{-s} - \alpha g(s,\alpha+1)\right] \quad (5.5)$$

Extending the recursion,

$$g(s,\alpha) = \frac{\exp^{-s}}{s}\left[1 + \sum_{i=1}^{\infty}(-1)^i \frac{\prod_{k=0}^{i-1}(\alpha+k)}{s^i}\right] \quad (5.6)$$

Now,

$$f_1^*(s) = f_2^*(s) = g(s, \alpha + 1)$$

Substituting the above functions in equation 5.2 and making use of recursion 5.5 we have

$$\begin{aligned}\mathcal{L}\{Var[N(t)]\} &= \frac{\lambda_1 + \lambda_2}{2s^2} + \\ &\quad \frac{(\lambda_1 - \lambda_2)^2}{2s^3}\left[1 + \frac{2(\alpha-1)}{\alpha s}\frac{[\exp^{-s} - \alpha g(s,\alpha+1)]^2}{[1 - \alpha^2 g^2(s, \alpha+1)]}\right]\end{aligned}$$

Since we want to determine the behaviour of variance as $t \to \infty$ or equally as $s \to 0$. Now, as $s \to 0$, $\exp^{-s} \approx 1$, then

$$\begin{aligned}\mathcal{L}\{Var[N(t)]\} &= \frac{\lambda_1 + \lambda_2}{2s^2} + \\ &\quad \frac{(\lambda_1 - \lambda_2)^2}{2s^3}\left[1 + \frac{2(\alpha-1)}{\alpha s}\frac{[1 - \alpha g(s,\alpha+1)]}{[1 + \alpha g(s, \alpha+1)]}\right]\end{aligned}$$

Now from recursion 5.5, $1 + \alpha g(s, \alpha + 1) = 2 - s g(s, \alpha)$ a nd
$1 - \alpha g(s, \alpha + 1) = s g(s, \alpha + 1)$, hence

$$\begin{aligned}\mathcal{L}\{Var[N(t)]\} &= \frac{\lambda_1 + \lambda_2}{2s^2} + \\ &\quad \frac{(\lambda_1 - \lambda_2)^2}{2s^3}\left[1 + \frac{2(\alpha-1)}{\alpha}\frac{g(s,\alpha+1)}{2 - s g(s, \alpha)}\right]\end{aligned}$$

Now, for small s, $s g(s, \alpha) \approx 1$, (from equation 5.6). Also for small s, the first term inside the braces my be neglected. Inverting, the final equation we have

$$Var[N(t)] = \frac{(\lambda_1 + \lambda_2)}{2} t + \frac{(\lambda_1 - \lambda_2)^2}{2}\left(\frac{\alpha - 1}{\alpha}\right) t^{3-\alpha} \tag{5.7}$$

Also, mean is given by,

$$E[N(t)] = \frac{\lambda_1 + \lambda_2}{2} t \tag{5.8}$$

Hence the IDC may be obtained by dividing the variance by the mean and is give n by

$$I(t) = 1 + \frac{(\lambda_1 - \lambda_2)^2}{\lambda_1 + \lambda_2}\left(\frac{\alpha - 1}{\alpha}\right) t^{2-\alpha}$$

Bibliography

[1] S. J. Campanella, "Digital speech interpolation," *Comsat Technical Review*, vol. 6, pp. 127–158, Spring 1976.

[2] P. T. Brady, "A model for generating on-off speech patterns in two way conversation," *Bell Syst. Tech. J.*, pp. 2445–2472, Sep 1969.

[3] P. T. Brady, "A statistical analysis of on-off patterns in 16 conversations," *Bell Syst. Tech. J.*, pp. 73–91, Jan 1968.

[4] K. Sriram and W. Whitt, "Characterizing superposition arrival processes in packet multiplexers for voice and data," *IEEE J. Select. Areas Commun.*, pp. 833–846, Sept. 1986.

[5] I. Ide, "Superposition of Interrupted Poisson Processes and its application to packetized voice multiplexers," in *Proc. International Teletraffic Congress - 12*, pp. 1399–1405, 1989.

[6] Y. C. Jenq, "Approximations for packetized voice traffic in statistical multiplexer," in *Proc. IEEE Infocom '84*, pp. 256–259, 1984.

[7] K. Sriram and W. Whitt, "Characterizing superposition arrival processes and the performance of multiplexers for voice and data," in *Proc. IEEE Globecom '85*, Dec. 1985.

[8] R. Gusella, "Characterizing the variability of arrival processes with index of dispersion," *IEEE J. Select. Areas Commun.*, pp. 203–211, Feb. 1991.

[9] H. Heffes and D. M. Lucantoni, "A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance," *IEEE J. Select. Areas Commun.*, vol. SAC-4, no. 6, pp. 856–868, 1986.

[10] R. Nagarajan, F. Kurose, and D. Towsley, "Approximation techniques for computing packet loss in infinite buffered voice multiplexers," *IEEE J. Select. Areas Commun.*, pp. 368–377, April 1991.

[11] T. E. Stern, "A queueing analysis of packet voice," in *Proc. IEEE Globecom '83*, pp. 71–76, 1983.

[12] J. N. Daigle and J. D. Langford, "Models for analysis of packet voice communication systems," *IEEE J. Select. Areas Commun.*, pp. 847–855, Feb. 1986.

[13] J. N. Daigle and J. D. Langford, "Queueing analysis of a packet voice communicating system," in *Proc. IEEE Infocom '85*, pp. 18–26, 1985.

[14] R. C. F. Tucker, "Accurate method for analysis of a packet speech multiplexer with limited delay," *IEEE Trans. Communications*, pp. 479–483, April 1988.

[15] C. Yuan and J. A. Sylvester, "Queueing analysis of delay constrained voice traffic in packet switching system," *IEEE J. Select. Areas Commun.*, pp. 729–739, 1989.

[16] S. Q. Li, "Study of packet loss in a packet switched voice system," in *Proc. ICC '88*, pp. 1519–1526, 1988.

[17] W. Whitt, "Approximating a point process by a renewal process: Two basic methods," *Operations Research*, Jan-Feb 1982.

[18] W. Whitt, "The queueing network analyzer," *Bell Syst. Tech. J.*, pp. 2779–2813, Nov 1983.

[19] S. L. Albin, "Approximating a point process by a renewal process, superposition arrival process to queues.," *Operations Res.*, pp. 1133–1162, Sept 1984.

[20] M. F. Neuts, *Matrix-geometric solutions in stochastic models: An algorithmic approach.* The John Hopkins University press, 1981.

[21] N. Ohta, *Packet video.* Artech House, Boston, MA, 1994.

[22] M. Nomura, T. Fujii, and N. Ohta, “Basic characteristics of variable rate video coding in ATM environment,” *IEEE J. Select. Areas Commun.*, pp. 752–760, June 1989.

[23] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, and J. D. Robbins, “Performance models of statistical multiplexing in packet video communications,” *IEEE Trans. Commun.*, vol. 36, pp. 834–843, July 1988.

[24] A. LaCorte, S. Lombardo, S. Palazzo, and S. Zinna, “Modeling activity in VBR video sources,” *Signal processing: Image Communication*, pp. 167–178, June 1991.

[25] T. Fujii, M. Nomura, and N. Ohta, “Characterization of variable rate inter-frame video coding for ATM-based networks,” in *Proc. IEEE Globecom '88*, pp. 1063–1067, Dec 1988.

[26] W. Verbiest, L. Pinnoo, and B. Voeten, “The impact of the ATM concept on video coding,” *IEEE J. Select. Areas Commun.*, pp. 1623–1632, Dec 1988.

[27] B. Maglaris, B. Anastassiou, G. K. P. Sen, and J. D. Robbins, “Performance analysis of statistical multiplexing for packet video sources,” in *Proc. IEEE Globecom '87*, pp. 1890–1899, 1987.

[28] S. S. Huang, “Source modeling for packet video,” in *Proc. ICC '88*, pp. 1262–1267, June 1988.

[29] R. Grunenfelder, J. P. Cosmas, S. Manthorpe, and A. Odinma-Okafor, “Characterization of video codecs as autoregressive moving average processes and related queueing system performance,” *IEEE J. Select. Areas Commun.*, pp. 284–293, April 1991.

[30] B. Melamed, D.Raychaudri, B. Sengupta, and J. Zdepski, "TES-based traffic modeling for performance evaluation of integrated networks," in *Proc. IEEE Infocom '92*, pp. 75–84, 1992.

[31] B. Melamed, D.Raychaudri, B. Sengupta, and J. Zdepski, "TES-based video source modeling for performance evaluation of integrated networks," *IEEE Trans. Commun.*, pp. 2773–2777, October 1994.

[32] B. Melamed, "The TES methodology: modeling temporal dependence in empirical time series," in *Proc. MASCOTS '93. International workshop on modeling, analysis and simulation of computer and telecommunication systems*, pp. 11–16, 17-20 Jan 1993.

[33] P. Bratley, B. L. Fox, and L. E. Schrage, *A Guide to Simulation.* Springer Verlag, New York, NY, 1987.

[34] D.Geist and B.Melamed, "TEStool: An environment for visual interactive modeling of autocorrelated traffic," in *Proc. ICC '92*, (Chicago, IL), pp. 1285–1289, 1992.

[35] P. Sen, M. Maglaris, N. Rikli, and D. Anastassiou, "Models for packet switching of variable bit rate video sources," *IEEE J. Select. Areas Commun.*, pp. 865–869, June 1989.

[36] F. Yegenoglu, B. Jabbari, and Y. Q. Zhang, "Modeling of motion classified VBR video sources," in *Proc. IEEE Infocom '92*, pp. 105–109, 1992.

[37] F. Yegenoglu, B. Jabbari, and Y. Q. Zhang, "Motion-classified autoregressive modeling of variable bit rate video," *IEEE Trans. on Circuits and Syst. for Video Technolgy*, pp. 42–53, Feb 1993.

[38] B. Jabbari, F. Yegenoglu, S. Z. Y. Kuo, and Y. Q. Zhang, "Statistical characterization and block-based modeling of motion-adaptive coded video," *IEEE Trans. on Circuits and Syst. for Video Technolgy*, pp. 199–207, June 1993.

[39] R. M. Rodriguez-Dagnino, M. R. K.Khansari, and A. Leon-Garcia, "Prediction of bit rate sequences of encoded video signals," *IEEE J. Select. Areas Commun.*, pp. 305–314, April 1991.

[40] D. L. McLaren and D. T. Nguyen, "Modeling low bit rate video traffic as switched-fractal source," *Electronics Letters*, pp. 745–747, April 1991.

[41] D. L. McLaren and D. T. Nguyen, "Variable bit-rate source modeling of ATM-based video services," *Signal processing: Image Communication*, pp. 233–244, June 1992.

[42] M. W. Garrett and W. Willinger, "Analysis, modeling and generation of self-similar VBR video traffic," in *Proc. ACM SIGCOMM '94. Conference on communication architecture, protocols and applications*, pp. 269–281, Aug. 31 - Sep. 2 1994.

[43] S. Anick, D. Mitra, and M. M. Sondhi, "Stochastic theory of a data-handling system with multiple source," *Bell Syst. Tech. J.*, vol. 36, pp. 834–843, July 1988.

[44] B. Meister, "Waiting time in a preemptive resume system with compound-Poisson input," *Comput.*, no. 1, pp. 17–28, 1980.

[45] R. Jain and S. A. Routhier, "Packet trains - meaurements and a new model for computer network traffic," *IEEE J. Select. Areas Commun.*, pp. 986–995, September 1986.

[46] Y. D. J. Lin, T. C. Tsai, S. C. Huang, and M. Gerla, "HAP: A new model for packet arrivals," in *Proc.ACM SIGCOMM '93*, pp. 212–223, September 1993.

[47] A. Erramilli, R. P. Singh, and P. Pruthi, "Chaotic maps as models of packet traffic," in *The fundamental role of teletraffic in the evolution of Telecommunications networks (Proc. 14th ITC)*, (Antibes Juan-les-Pins), pp. 329–338, 1994.

[48] I. Norros, "A storage model with self-similar input," *Queueing Systems*, no. 16, pp. 387–396, 1994.

[49] B. S. Slimane and T. Le-Ngoc, "A doubly stochastic Poisson model for self-similar traffic," in *Proc. ICC '95*, (Seattle, Washington), pp. 456–460, 1995.

[50] W. E. Leland, S. M. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of ethernet traffic (extended version)," *IEEE/ACM Trans. on Networking*, vol. 2, pp. 1–14, February 1994.

[51] W. E. Leland and D. V. Wilson, "High time-resolution measurement and analysis of LAN traffic: implications for LAN interconnection," in *Proc. IEEE Infocom '91*, (Bal Harbour, FL), pp. 1360–1366, 1991.

[52] H. J. Fowler and W. E. Leland, "Local area network traffic characteristics, with implications for broadband network congestion management," *IEEE J. Select. Areas Commun.*, pp. 1139–1149, September 1991.

[53] K. S. Meier-Hellstern, P. E. Wirth, Y. L. Yan, and D. A. Hoeflin, "Traffic models for ISDN data users: office automation application," in *Teletraffic and data traffic in a period of change (Proc. 13th ITC)*, (Copenhagen, Denmark), pp. 167–172, A.Jensen, V.B.Iversen (Eds.), North Holland, 1991.

[54] W. E. Leland, S. M. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of ethernet traffic," in *Proc. ACM SIGCOMM '93*, (Sanfransisco, CA), pp. 183–193, September 1993.

[55] A. Erramilli and W. Willinger, "Fractal properties in packet traffic measurements," in *Proc. ITC Seminar*, 1993.

[56] A. Erramilli, J. Gordon, and W. Willinger, "Applications of fractals in engineering for realistic traffic processes," in *Proc. International Teletraffic Congress - 14*, (Antibes Juan-les-Pins), pp. 35–43, 1994.

[57] D. M. Lucantoni, "New results for the single server queue with a batch Markovian arrival process," *Stoch. Models*, pp. 1–46, 1991.

[58] A. Baiocchi, N. B. Melazzi, M. Listanti, A. Roveri, and R. Winkler, "Loss performance analysis of an ATM multiplexer loaded with on-off sources," *IEEE J. Select. Areas Commun.*, vol. SAC-9, pp. 388–393, April 1991.

[59] S. B. Kim, M. Y. Lee, and M. J. Kim, "Σ-Matching technique for MMPP modeling of heterogeneous on-off sources," in *Proc. IEEE Globecom '94*, (San Fransisco, CA), pp. 1090–1094, 1994.

[60] J. W. Lee and B. G. Lee, "Performance analysis of ATM cell multiplexer with MMPP input," *IEICE Trans. Commun.*, vol. E75-B, pp. 709–714, August 1992.

[61] E. D. Sykas, K. M. Vlakos, and N. G. Anerousis, "Performance evaluation of statistical multiplexing scheme in ATM networks," *Computer Networks*, vol. 14, pp. 273–286, June 1991.

[62] H. Kobayashi, "Performance Issues of Broadband ISDN Part II: Statistical multiplexing of multiple types of traffic," in *Proc. ICC '90*, (New Delhi, India), pp. 355–360, 1990.

[63] S. S. Wang and J. A. Silvester, "Estimate the loss and tail probabilities for multimedia communication systems," in *Proc. IEEE Globecom '94*, (San Fransisco, CA), pp. 908–912, 1994.

[64] R. Gusella, "A measurement study of diskless workstation traffic on an ethernet," *IEEE Trans. Communications*, vol. 38, no. 9, pp. 1557–1568, 1990.

[65] V. Paxson and S. Floyd, "Wide-area traffic: The failure of Poisson modeling," in *Proc.ACM SIGCOMM '94*, pp. 257–268, 1994.

[66] S. Q. Li, "A new performance measurement for voice transmission in burst and packet switching," *IEEE Trans. Communications*, vol. 35, p. 1083, Oct 1987.

[67] T. C. Hou and H. K. Wong, "Queueing analysis for ATM switching of mixed continuous-bit-rate and bursty traffic," in *Proc. IEEE Infocom '90*, pp. 660 – 667, June 1990.

[68] D. P. Gaver, “Random hazard in reliability problems,” *Technometrics*, vol. 5, pp. 211–226, May 1963.

[69] A.J.Lawrence, “Some models for stationary series of univariate events,” *Stochastic Point processes : Statistical Analysis Theory and Applications*, pp. 199–256, 1972.

[70] H. Saito, *Teletraffic Technologies in ATM Networks.* Artech House, 1993.

[71] B. B. Mandelbrot, “Long-run linearity, locally Gaussian processes, H-spectra and infinite variances,” *Technometrics*, vol. 10, pp. 82–113, 1969.

[72] M. S. Taqqu and J. B. Levy, “Using renewal processes to generate long range dependence and high variabilty,” in *Dependence in Probabilty and Statistics*, pp. 73–89, E.Eberlein and M.S.Taqqu (Eds.), Progress in Prob. and Stat. Vol. 11, Birkhauser, Boston, 1986.

[73] I. Norros, “On the use of Fractional Brownian motion in the theory of connectionless networks,” *IEEE J. Select. Areas Commun.*, vol. 13, pp. 953 – 962, August 1995.

[74] P. Pruthi and A. Erramilli, “Heavy tailed on/off source behaviour and self-similar traffic,” in *Proc. ICC '95*, (Seattle, Washington), pp. 445–450, 1995.

ATTACHMENT # 1

User Manual
of
TRAFFIC GENERATOR

Bach Ngoc Quang
Selvakumaran Subramanian

Department of Computer and Electrical Engineering
Concordia University
1455 DeMaisonneuve Blvd. Ouest
Montréal, Québec
Canada H3G 1M8

Contents

List of Figures

1 INTRODUCTION

Future high-speed networks are expected to support various services with different characteristics such as voice, data, and video. The traffic which is generated from these services is substantially different in their nature. Studying and understanding the impact of different traffic types with different characteristics on the performance are crucial for a successful and efficient design of such networks. To ease these tasks, a simulation model has been implemented in opnet (optimized network engineering tool) environment. Several traffic models have been implemented: ON/OFF model (for all types of traffic), MMPP (Markow Modulated Poisson Process, for voice and video traffic), PMPP (Pareto Modulated Poisson Process, for data traffic).

1.1 ON/OFF SOURCES MODEL

The basic ON/OFF source model is characterized by alternating independent ON (burst) and OFF (silence) periods, where this periods may have general distributions. Cells are generated during the ON period with constant rate λ (cells/sec) . No cell is generated during OFF period. All type of traffic sources can be decomposed into a superposition of the basic ON/OFF sources.

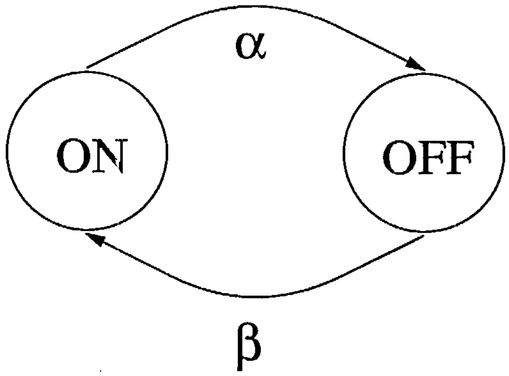


Figure 1: The basic ON/OFF source model

Each voice source is represented by an ON/OFF source. ON and OFF periods are exponentially distributed with the mean $1/\alpha$ and $1/\beta$, respectively.

Data source is modeled as a set of ON/OFF sources. The ON period is Pareto distributed with the mean $1/\alpha$, the OFF period is exponentially distributed with the mean $1/\beta$.

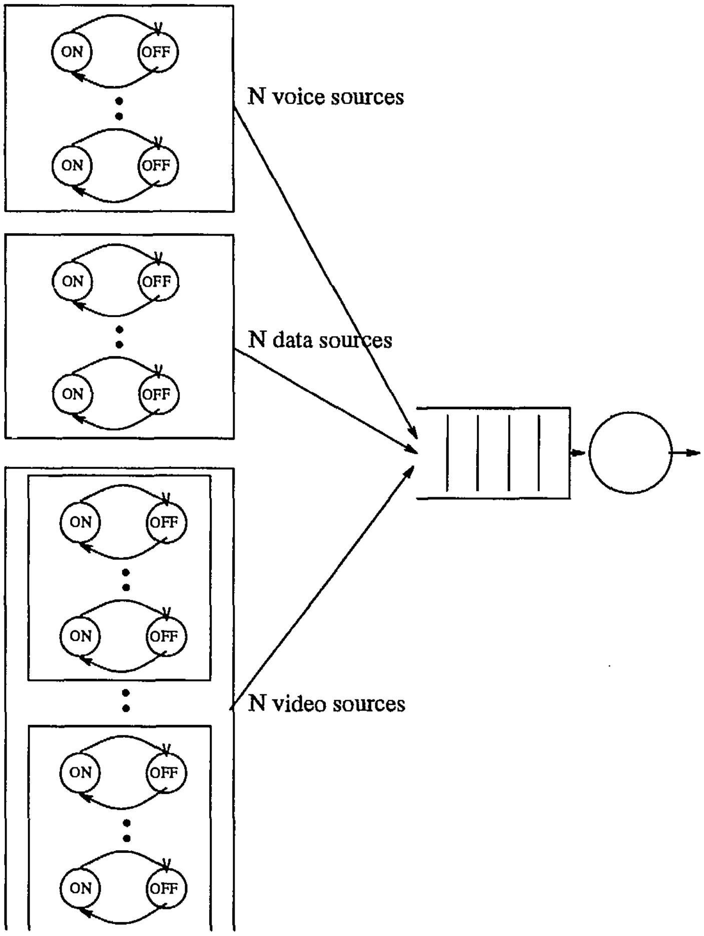


Figure 2: The ON/OFF sources model for voice, data, and video

Each video source is represented by a set of n (varies from 10 to 20) independent mini ON /OFF sources. The ON and OFF periods are exponentially distributed with the mean $1/\alpha$ and $1/\beta$, respectively.

1.2 OTHER SOURCES MODEL

Other traffic models also have been implemented in this integrated traffic model. They are MMPP for voice, PMPP for data, and MMPP for video. These are proposed models and are used popularly.

1.3 IDI AND IDC PROBE MODEL

The dependence among successive interval times in the aggregated packet arrival process is characterized by the indices of dispersion of intervals (IDI) and counts (IDC). To calculate IDI and IDC, a large amount of data must be collected. The IDI and IDC probe model will do this job.

2 MODEL INTERNAL STRUCTURE

This section describes the internal structure of all the process models: the primary traffic integrated (trf_generator) process model, the basic supporting ON/OFF (trf_vc_on_off, trf_dt_on_off, trf_vd_on_off) process model, and the statistic probe (trf_idi_idc) process model.

In each section, first the State Transition Diagram (STD) is displayed and discussed, and then each individual state is described in detail.

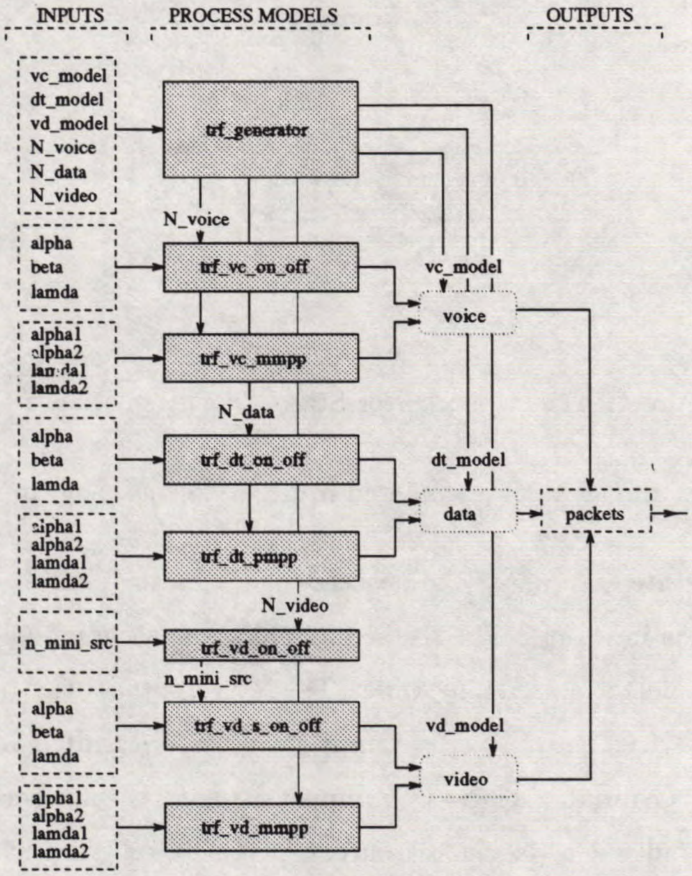


Figure 3: Structure Programming of Integrated Traffic Model

2.1 PRIMARY PROCESS MODEL: trf_generator

The task of this primary process model is to generate the number of traffic sources according to the number of sources and type of model wanted.The trf_generator process model is composed of four states, and transitions that define the transfer of control between

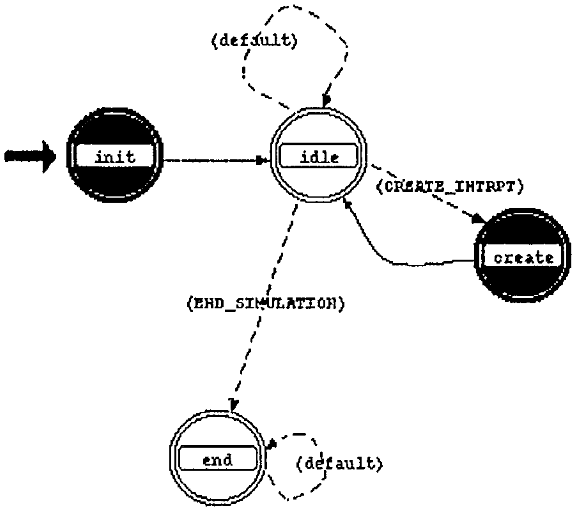


Figure 4: The trf_generator State Transition Diagram

the states. The process model STD is depicted in the following diagram.

init This is the first state entered by the process model and the initial interrupt should be a begin simulation interrupt. The traffic model for voice (vc_model), video (vd_model) and data (dt_model) are user selectable. This state obtains the model used in simulation for voice (trf_vc_on_off or trf_vc_mmpp), for data (trf_dt_on_off or trf_dt_pmpp), and for video (trf_vd_on_off or trf_vd_mmpp). It gets the number of voice (N_voice), data (N_data), and video (N_video) sources for ON/OFF models. For other models, these values should be set to 1.

idle This unforced state stays idle all the time till the end of simulation.

creator This state gets the control from idle state via CREATE_INTRPT transition. It invokes the coressponding number of sources for each traffic type models based on inputs obtained in init state and returns the control to idle state.

end This state is entered when the simulation time is finished.

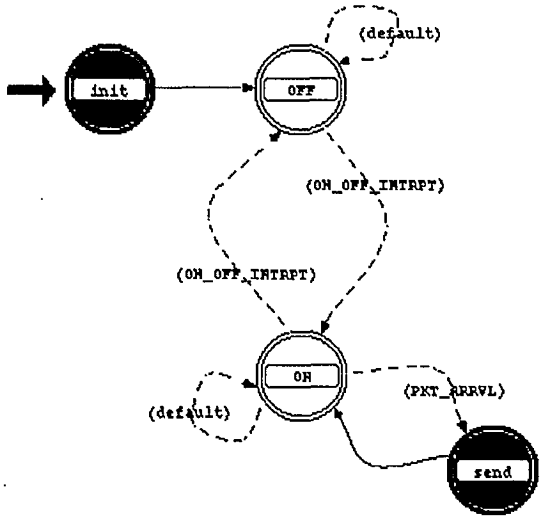


Figure 5: The general ON/OFF State Transition Diagram

2.2 SUPPORTING PROCESS MODEL: trf_vc_on_off

All the ON/OFF traffic model processes has the similar structure, called ON/OFF source structure. This section will examine the general structure, and also the difference between them. The general ON/OFF source model process has four states: init, off, on and send.

Its task is to generate packets (or format packets) with any distribution. Normally, the process will generate packets during ON time with the constant rate λ (lambda) . During the OFF time, no packet is generated.

init This state will get all the parameters needed: the rate of transition from ON to OFF α (alpha), the rate of transition from OFF to ON β (beta), the rate of generating packets during ON time λ (lambda).

off This state schedules a transition to on state according to exponential distribution with the mean $1/\beta$. During OFF period, it stays idle.

on This state schedules a transition to off state according to exponential distribution (except for data, where the Pareto distribution is used) with the mean $1/\alpha$. During ON period, it generates packets by transferring the control to send state.

send This state forms the unformatted packets (or formatted packets), and send them at a constant rate λ.

2.3 THE STATISTIC PROBE MODEL: trf_idi_idc

The definition of IDI and IDC is given in Appendix B. This section only discusses the statistic probe model trf_idi_idc which is implemented as a probe to collect data need in calculating the IDI and IDC curves. This process model is placed in a node located between sources and destination. No delay time for packets when they go through the node. The node collects the interarrival time between two successive packet arrivals, and also the number of packets count per unit time (1 sec). The process model has four states: init, idle, idi, and idc.

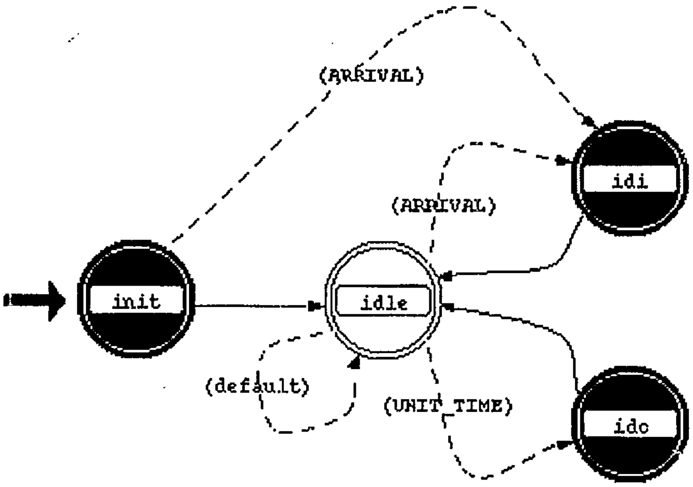


Figure 6: The trf_idi_idc State Transition Diagram

init This state prompts the user to input the filenames for storing the data for IDI and IDC calculation.

idle This is the only unforced state in the process. It gives control to idi state whenever a packet arrives or to idc state whenever a unit time has elapsed.

idi This state calculates the time interval between two successive packet arrivals, writes it to the specified IDI data filename. This also advances the incoming packets.

idc This state is entered every unit time (1 sec). It records the number of packets that have passed through the node during that unit time, and writes it to the specified IDC data filename.

3 EXAMPLE USAGE

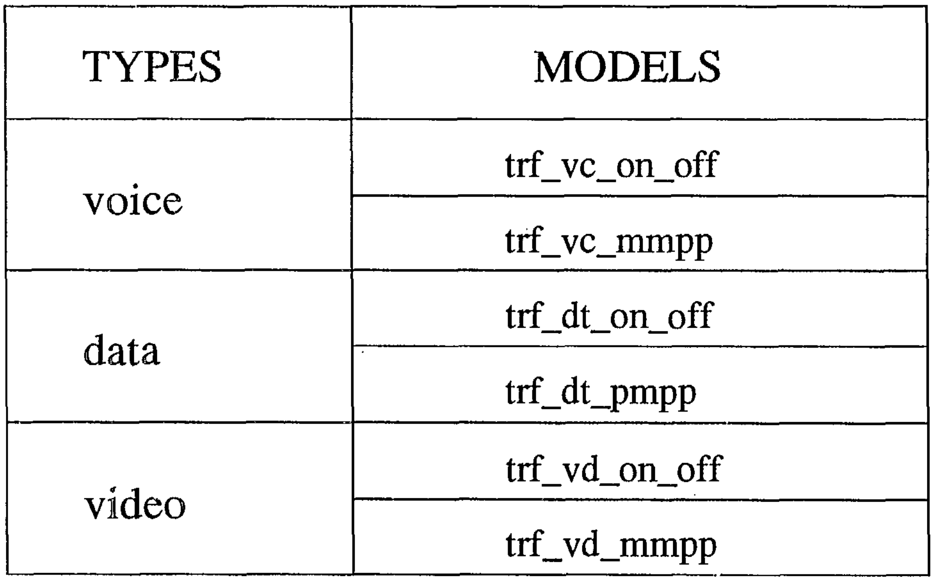
3.1 DEFINING THE TASK

This chapter presents a practical example of creating a simulation executable, running simulations, and analyzing simulation results. The user will be guided step by step how to solve the problem. He will use the Process Editor, Node Editor, Simulation Tool, and provided programs to perform the analysis. The task in this chapter is to use the integrated model to simulate the traffic behavior of a set of voices, data, and video sources.

3.2 CREATING A SIMULATION EXECUTABLE

3.2.1 CREATING THE PROCESS MODEL

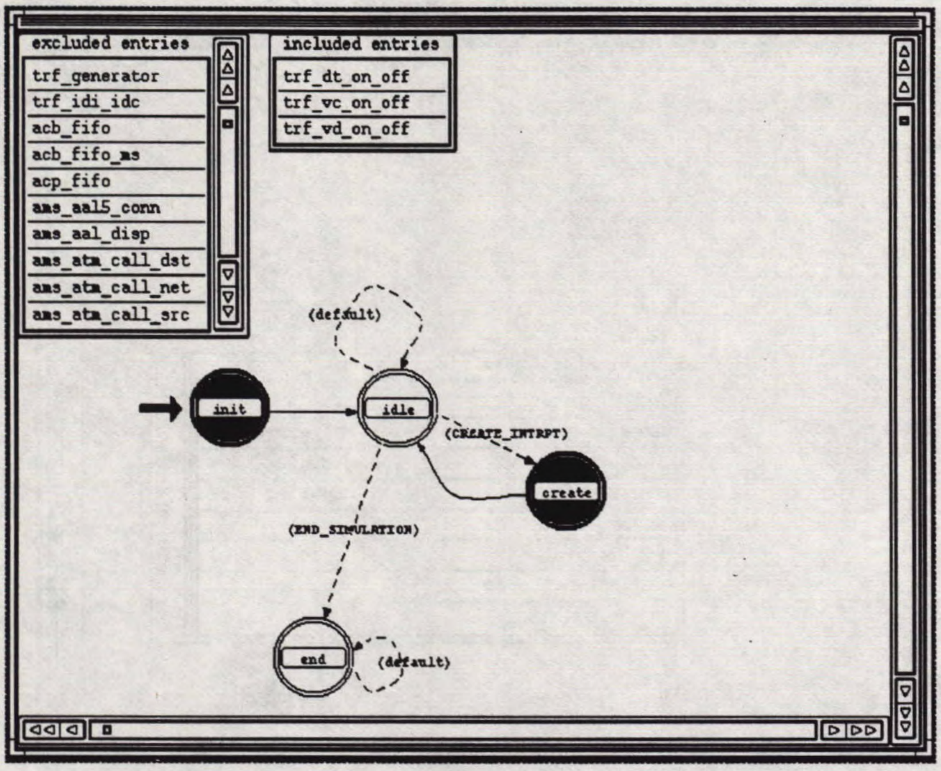
The Process Editor is used to create the process models. Open the primary process model trf_generator. The next step is to defined which type of models we want to use for each type of traffic. Here is the table of all models provided.



TYPES	MODELS
voice	trf_vc_on_off
	trf_vc_mmpp
data	trf_dt_on_off
	trf_dt_pmpp
video	trf_vd_on_off
	trf_vd_mmpp

Figure 7: All models implemented

Assume that the user wants to use ON/OFF sources for all types of traffic. So the process



excluded entries

trf_generator
trf_idi_idc
acb_fifo
acb_fifo_ms
acp_fifo
ams_aal5_conn
ams_aal_disp
ams_atm_call_dst
ams_atm_call_net
ams_atm_call_src

included entries

trf_dt_on_off
trf_vc_on_off
trf_vd_on_off

Figure 8: Choose the desired models for each type of traffic

models used are: trf_vc_on_off, trf_dt_on_off, and trf_vd_on_off.

- Open the "child process" (OPNET concept of sub- processes invoked from a principal running process) attribute menu by clicking at the child process icon.

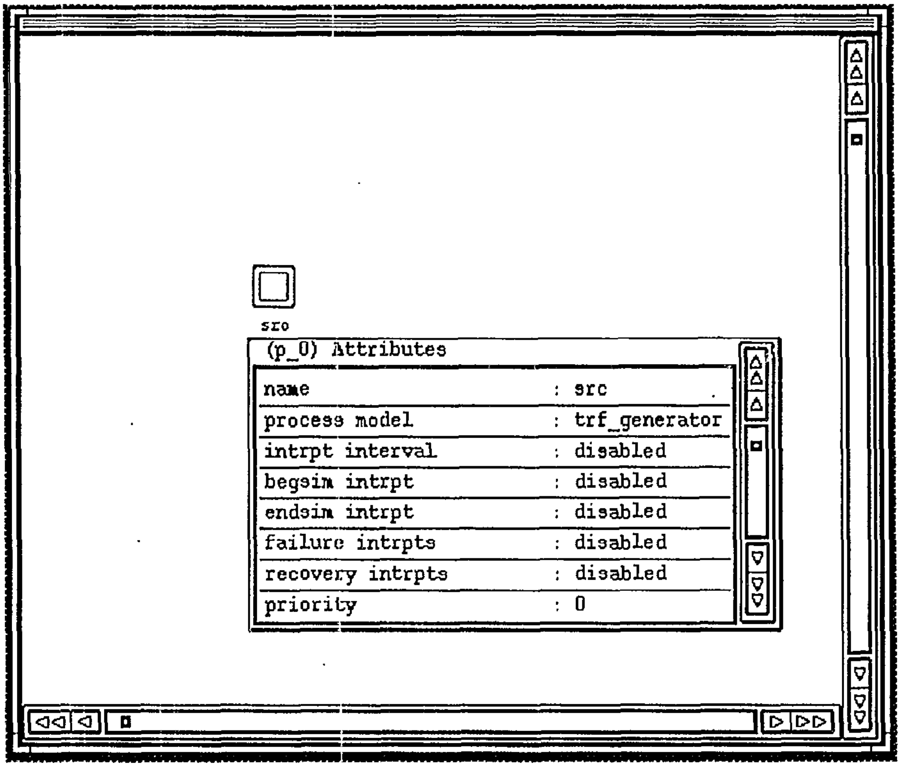
- Chose the proper process models by clicking at their names. All the chosen names will appear at the right hand side.

- Correct mistake by chose the name at the right hand side.

The last step in this Process Editor is to combine and save the process model under the name Traffic.

3.2.2 CREATING THE NODE MODEL

The Node Editor is used to create the node model. In this example, the user will create three modules: the src (source), the prb (probe), and the sink (sink). The src module will generate the packets. The prb module will collect the data need for IDI and IDC calculations. The sink module will destroy all generated packets.



(p_0) Attributes

name	:	src
process model	:	trf_generator
intrpt interval	:	disabled
begsim intrpt	:	disabled
endsim intrpt	:	disabled
failure intrpts	:	disabled
recovery intrpts	:	disabled
priority	:	0

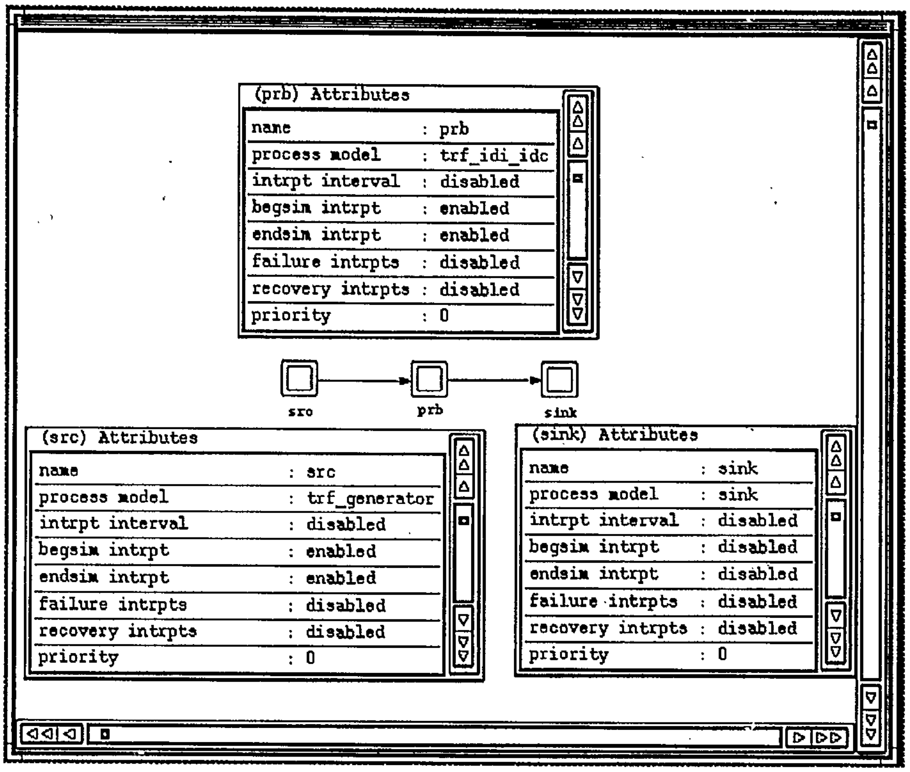
Figure 9: Create the src module

3.2.2.1 Creating the src module

- Create a processor module. Name the module src.
- Open the processor attribute menu.
- Chose trf_generator for the process model. Enable the begsim intrpt and the endsim intrpt.
- Close the processor attribute menu.

3.2.2.2 Creating the prb module

- Create a processor module. Name the module prb.
- Open the processor attribute menu. Chose trf_idi_idc for the process model. Enable the begsim intrpt and the endsim intrpt.
- Close the processor attribute menu.



(prb) Attributes

name	:	prb
process model	:	trf_idi_idc
intrpt interval	:	disabled
begsim intrpt	:	enabled
endsim intrpt	:	enabled
failure intrpts	:	disabled
recovery intrpts	:	disabled
priority	:	0

(src) Attributes

name	:	src
process model	:	trf_generator
intrpt interval	:	disabled
begsim intrpt	:	enabled
endsim intrpt	:	enabled
failure intrpts	:	disabled
recovery intrpts	:	disabled
priority	:	0

(sink) Attributes

name	:	sink
process model	:	sink
intrpt interval	:	disabled
begsim intrpt	:	disabled
endsim intrpt	:	disabled
failure intrpts	:	disabled
recovery intrpts	:	disabled
priority	:	0

Figure 10: The Traffic node model

3.2.2.3 Creating the sink model

- Create a processor module. Name the module sink.

- Open the processor attribute menu. Chose sink for the process model.

- Close the processor attribute menu.

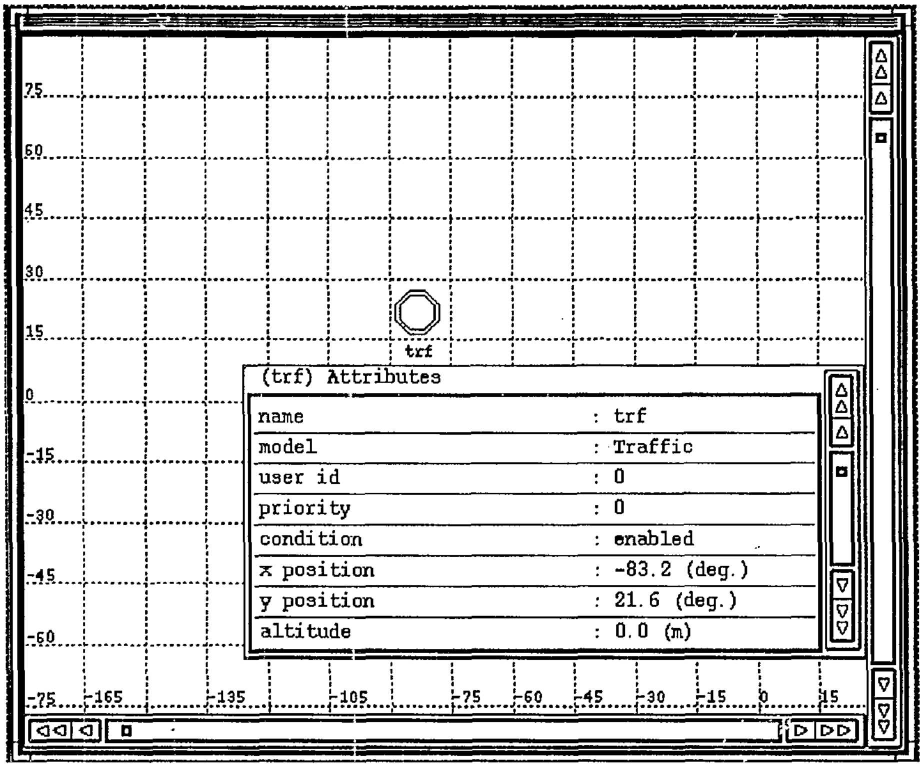
3.2.2.4 Connect the Modules with Packet Streams

- Activate the create packet stream action button.

- Connect a stream from src to prb.

- Connect another stream from prb to sink.

Save the created node under the name Traffic and close this Node Editor.



(trf) Attributes

name	: trf
model	: Traffic
user id	: 0
priority	: 0
condition	: enabled
x position	: -83.2 (deg.)
y position	: 21.6 (deg.)
altitude	: 0.0 (m)

Figure 11: Define the network model

3.2.3 CREATE THE NETWORK MODEL

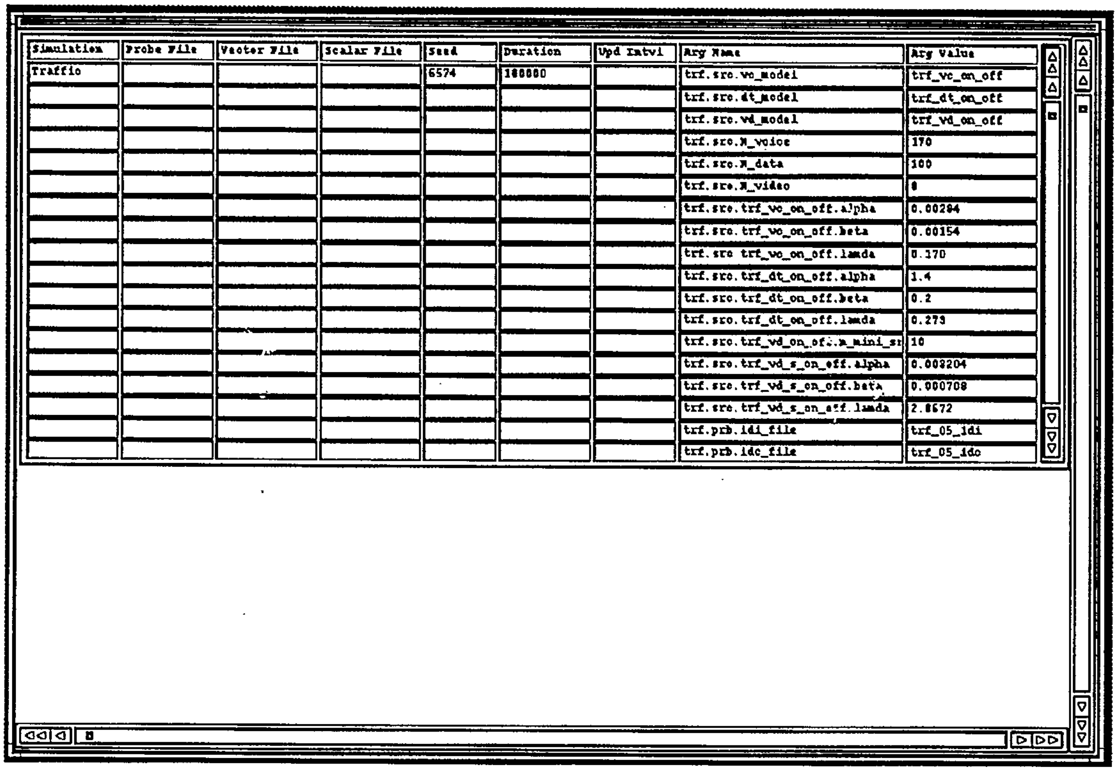
The traffic network model will consist of a single node object based on the Trafiic node model. The Network Editor is used both to define the network model and create the simulation executable.

3.2.3.1 Define the network model

- Open the Network Editor.
- Activate the create fixed comm. node action button.
- Select Traffic from the menu available node models, then name the node Traffic.

3.2.3.2 Create the simulation executable

- Activate the archive, bind simulation action button.
- Supply the filename Traffic.



Simulation	Probe File	Vector File	Scalar File	Seed	Duration	Upd Intvl	Arg Name	Arg Value
Traffic				6574	100000		trf.src.vc_model	trf_vc_on_off
							trf.src.dt_model	trf_dt_on_off
							trf.src.vd_model	trf_vd_on_off
							trf.src.N_voice	170
							trf.src.N_data	100
							trf.src.N_video	0
							trf.src.trf_vc_on_off.alpha	0.00294
							trf.src.trf_vc_on_off.beta	0.00154
							trf.src trf_vc_on_off.lamda	0.170
							trf.src.trf_dt_on_off.alpha	1.4
							trf.src.trf_dt_on_off.beta	0.2
							trf.src.trf_dt_on_off.lamda	0.273
							trf.src.trf_vd_on_off.n_mini_sr	10
							trf.src.trf_vd_s_on_off.alpha	0.008204
							trf.src.trf_vd_s_on_off.beta	0.000708
							trf.src.trf_vd_s_on_off.lamda	2.8672
							trf.prb.idi_file	trf_05_idi
							trf.prb.idc_file	trf_05_idc

Figure 12: Input all parameters need to run the simulation

Opnet builds the model archive and binds the separate simulation components, creating a simulation executable called Traffic.sim.

3.2.4 EXECUTING THE SIMULATION

The user will use the Simulation Tool to invoke the stand-alone simulation executable. Supply the model-independent simulation parameters to the Simulation Tool data table as shown below.

3.3 ANALYZING THE RESULTS

The user will use the provided programs to calculate the IDI and IDC curves. These programs are written in matlab. The code of the programs can be found in Appendix C.The IDI and IDC curves are obtained. One of the results, the IDC curve, is shown below.

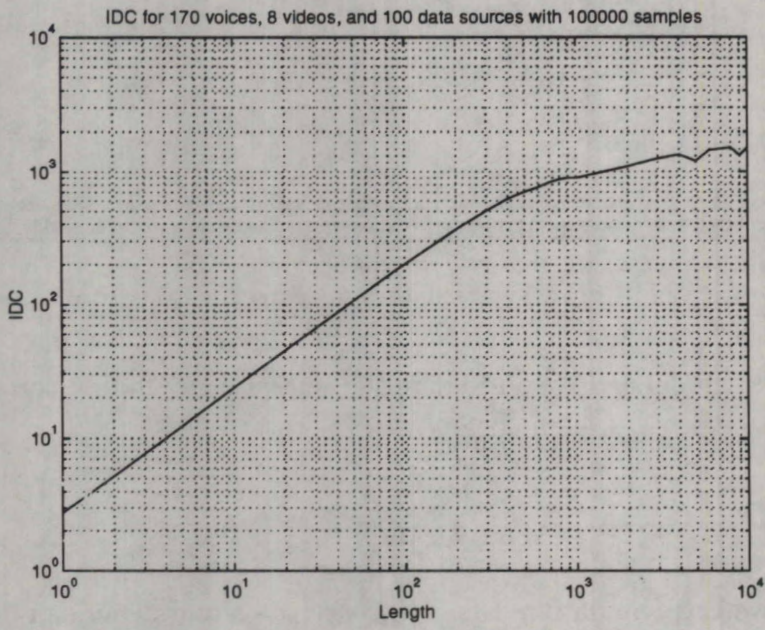


Figure 13: The IDC curve

A APPENDIX A: TIPS ON FORMATTED PACKET

This section has an example to show the user how to create a formatted packets.

```
/* create a formatted packet */
f_pkptr = op_pk_create_fmt( ”example_pkt”);

/* assign integer fields in the packet */
op_pk_nfd_set( f_pkptr, ”fd_int_1”, 3);
db_value = 5 * 200;
op_pk_nfd_set( f_pkptr,”fd_int_2”, int_value);

/* assign double fields in the packet */
op_pk_nfd_set( f_pkptr, ”fd_double_1”, 2511.67);
int_value = 5.4 * 200.55;
op_pk_nfd_set( f_pkptr,”fd_double_2”, int_value);

/* encapsulated a higher-level packet in the packet */
enc_pkptr = op_pk_create( 24 );
op_pk_nfd_set( f_pkptr,”fd_packet”,enc_pkptr);

/* send the defined packet out of the processor */
op_pk_send( f_pkptr, OUTSTRM);
```

B APPENDIX B: THEORY

B.1 INDEX OF DISPERSION FOR INTERVALS (IDI)

The index of dispersion of intervals is used to focus on the dependence among successive interarrival times in the aggregate packet arrival process.

Let $\{X_k, k >= 1\}$ represent the sequence of packet interarrival times from the superposition process of integrated traffic sources. The IDI, also called the k interval squared coefficient of variation sequence, is the sequence $\{c_k^2, k = 1\}$ defined by

$$c_k^2 = \frac{kVar\{X_1 + X_2 + ... + X_k\}}{E[\{X_1 + X_2 + ... + X_k\}]^2}$$

Assuming that $X_k, k = 1$ is stationary we note that the sum $X_1 + X_2 + ... + X_k = X_{i+1} + X_{i+2} + ... + X_{i+k}$. Denoting this sum by S_k we have

$$c_k^2 = \frac{kVar(S_k)}{[E(S_k)]^2} = \frac{Var(S_k)}{k[E(X_1)]^2}$$

B.2 INDEX OF DISPERSION FOR COUNTS (IDC)

To define the Index of Dispersion for Counts (IDC), let N(t) denote the counting process associated with an arrival process. The index of dispersion for counts, I(t), is the function

$$I(t) = \frac{Var[N(t)]}{E[N(t)]}, t > 0$$

C APPENDIX C: PROGRAMS

C.1 PROGRAM FOR CALCULATING IDI

This program is provided by Hamid-Reza Mehrvar and is written in matlab. Here is an example to calculate the IDI with 100000 data from the file trf_05_idi.

```
------------
clear


%  This Program calculates Index of Dispersion for Interval
%  and also collect Statistics for Variance-time  decaying curve.
%  Index of dispersion and Variance-time values are calculated
%  in interval of length of L frame time, T. The results are
% in vectors I and V1 and are plotted versus L.


no_data=100000;
fpt1=fopen('trf_05_idi');


S=0;
for k=1:100000
X=fscanf(fpt1,'%f',1);
S=S+X;
end
avg=S/100000;


L=[1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500,
600, 700, 800, 900, 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000,9000, 10000 ];


b=length(L);


%diary output
```

```
for r=1:b
clear fpt1 sum Sum V E  no_batch X Y

fpt1=fopen('trf_05_idi');

if (L(r)==1) no_batch=5000;
else no_batch=fix(no_data/ L(r));
end
for i=1:no_batch

   Y(i)=0;
   for j=1:L(r)
   X=fscanf(fpt1,'%f',1);
   Y(i)=Y(i)+ X;
   end

end



n=length(Y);

E=sum(Y)/n;

sum=0;
for j=1:n
sum=sum+ (Y(j)-E)^2;
end

V=sum/(n-1);
I(r) =V/(L(r)*E^2);
V1(r)=V/(L(r)^2);
```

```
save  trf_05_idi  L I V1
loglog(L,I)

end
------------
```

C.2 PROGRAM FOR CALCULATING IDC

This program is provided by Hamid-Reza Mehrvar and is written in matlab. Here is an example to calculate the IDC with 100000 data from the file trf_05_idc.

```
------------
clear

%  This Program calculates Index of Dispersion for Count Process
%  and also collect Statistics for Variance-time  decaying curve
%  Index of dispersion and Variance-time values are calculated
%  in interval of length of L frame time, T. The results are
% in vectors I and V1 and are plotted versus L.

no_data=100000;
L=[1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500, 600,

700, 800, 900, 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000];
b=length(L);
%diary output
for r=1:b
clear fpt1 sum Sum V E  no_batch X Y
```

```
fpt1=fopen('trf_05_idc');
no_batch=fix(no_data/ L(r));
for i=1:no_batch
   Y(i)=0;
   for j=1:L(r)
   X=fscanf(fpt1,'%f',1);
   Y(i)=Y(i)+ X;
   end
 end
n=length(Y);
E=sum(Y)/n;
sum=0;
for j=1:n
sum=sum+ (Y(j)-E)^2;
end

V=sum/(n-1);
I(r) =V/E;
V1(r)=V/(L(r)^2);

save  trf_05_idc  L I V1
loglog(L,I)
end

%diary off
------------
```

Technical Report:

Performance of CFDAMA in a Multimedia SAT-COM System using MF-TDMA

by
R. Di Girolamo and T. Le-Ngoc

Report Summary

The main purpose of this work is to evaluate the performance of CFDAMA in a multimedia SATCOM system, using an MF-TDMA frame. This report presents the progress made in the development of the OPNET Network Simulator for the above multiple access scheme. Originally, models of the CFDAMA protocol were available only for Poisson type data traffic, and constant bit rate (CBR) voice. These initial simulations clearly showed the usefulness of CFDAMA in a satellite environment. However, many questions regarding CFDAMA remained unanswered. Of primary concern is the performance of CFDAMA in a multimedia environment, when users generate both real-time traffic (with variable bit rates (VBR)) and long-range dependent data traffic. To answer this, the OPNET models were modified to include the newly developed traffic generator, which produces the desired traffic profiles.

The simulation results presented were carried out for MF-TDMA with CFDAMA-PA. The traffic considered is multimedia (VBR voice and video and long-range dependent data). For the real-time traffic we investigate the loss probability, whereas delay is the figure of merit for the jitter-tolerant traffic. The MMPP model is used to represent the VBR voice and video traffic, while the PMPP model is used for the data traffic sources. Two slot assignment (scheduler) strategies are investigated. Scheduler 1 treats all users independently, while scheduler 2 treats all users collectively. Results show that the CFDAMA protocol still performs well under the conditions tested. The overall conclusions are:

1. Treating the users collectively, for slot assignment, reduces the loss probability

considerably, since we can make better use of each users random traffic levels.

2. Although the loss probability is reduced, the data delay is increased for scheduler 2. This comes about since the lower loss implies that more real-time packets are being transmitted, leaving less slots for the data traffic, and therefore a larger delay.
3. Scheduler 1 results in large loss probabilities which are independent of load levels.

1. Introduction

This report presents a review of the current state of the OPNET network simulator, for the proposed fixed satellite system. The purpose is to give an overview of the basic operation of the system, the type of traffic generated by the earth terminals, and the scheduling algorithms included in the on-board processing satellite.

The basic system considered, uses CFDAMA (Combined free/demand assignment multiple access), with a multi-frequency time-division multiple access (MF-TDMA) frame - a secondary access scheme resting on top of a primary access scheme. In order to support real-time traffic, an underlying frame structure must be present in the primary access scheme. The recurring frame allows the user requirements to be met for constant and variable bit rate (CBR and VBR) sources. The basic scheme is outlined in Section 2. Section 3 then presents a description of the models. The different traffic sources, requesting strategies, and schedulers are discussed. Flowcharts are also given showing the progression of the simulations. Section 4 highlights the parameters for the system under study. Results and conclusions are included in Section 5. The figures of merit include the data cell delay experienced by the data packets as they wait for a slot (when a packet is given a slot, we mean that this packet is transmitted), and the real-time loss probability resulting from a lack of capacity on the uplink.

2. Basic Operation of CFDAMA with MF-TDMA

For any fixed bandwidth, the number of slots available per frame is limited to some N_{max}. For MF-TDMA, the slots are divided amongst a number of carriers (TDMA is a special case of MF-TDMA with a single carrier). Since the satellite network envisioned is to serve a large number of low load earth terminals (direct to homes, small businesses,...) it would be unwise to assign the available slots to the users on a fixed basis, since many of these would go idle, thereby reducing overall channel utilization. For such a case, it is not hard to imagine a user having to wait many frames for his preassigned slot while the slots of users who have no information to transmit, go empty. Therefore, it seems natural to assign slots based on user requirements, by having each of the users make requests or reservations for these. This is a dynamic capacity allocation scheme. There are a host of techniques which aim to achieve this goal. These include TDMA-reservation, Aloha-reservation, combined random-reservation multiple access (CRRMA)[1,2], and combined fixed/demand assignment multiple access. For our purposes, we propose to investigate the performance of CFDAMA (Combined free/demand assignment multiple access). Early results with Poisson type data traffic suggest that this technique performs very well for a range of user sizes [3].

The main concept of CFDAMA is to interleave channels which are assigned based on requests, with those which are assigned freely. The concept of free assignment will be explained later. Every frame is divided into two sections: a reservation section and a traffic section (See Figure 1). The users make requests in the reservation section, and when they are given a slot, they transmit their packet in the traffic section.

The basic operation can be explained by considering Figure 2, which shows how requests are transmitted over time, and how the scheduler deals with these requests. For the moment, consider only jitter-tolerant traffic. As user *k* receives packets, it stores these in its queue. Depending on the reservation strategy employed (see Section 2.1), user *k* sends requests to the on board processor (OBP) scheduler. These requests arrive at the satellite after the appropriate delay and are placed in a scheduler queue along with the requests from all other users. These requests are then serviced at the start of every frame. When the scheduler services a request, we mean that it assigns a slot in the current frame to the user that initiated the request. The scheduler assigns all the slots in the current frame, and notifications are sent to the users so that they may transmit at the specified time. This notification arrives at the user terminals, again after a suitable transmission delay.

At the scheduler, the requests are served on a first-come first served basis. Owing to the randomness of user transmissions, it is possible that after assigning a certain number of slots, the scheduler queue becomes empty. The scheduler knows that there are slots available, but it has no more requests to service. However, by the time the scheduler notification arrives at the user earth terminals, additional packets may have arrived. Therefore, the scheduler can **free** assign any unused slots to earth terminals. When the notification for the free assigned slots arrives at the earth terminals, and these terminals have packets to transmit, these can be transmitted without having made a request. The delay experienced by these packets is therefore very small.

The above discussion strictly applies only to jitter-tolerant traffic which can be queued. For real-time traffic, the situation is slightly different. Basically, the earth terminal makes a request every frame, depending on the number of slots it requires. Therefore, once a user is given notification that his real-time call is accepted, he makes a request for the number slots he requires for the first frame. After one round trip delay, the user transmits in the slots which have been assigned by the scheduler. In the meantime, he has continued to generate real-time packets, and has continued to make requests, once every frame. The scheduler has serviced these, and has sent notification back to the earth terminal. In essence, the real-time traffic has a guaranteed number of slots every frame. The only problem is that the earth terminal must queue all the real-time packets for an entire round trip time (which is the time taken before acknowledgment for the first request is received). Fortunately this is an absolute delay, and can be included as part of the call set-up phase.

The scheduling for real-time traffic is also slightly different. Since the real-time traffic should have priority over the jitter-tolerant traffic, the scheduler first services requests for the former (this is denoted in Figure 2 by having two queues - one for the real-time and the other for the jitter-tolerant traffic). After servicing these requests, the scheduler services the jitter-tolerant requests, as discussed above.

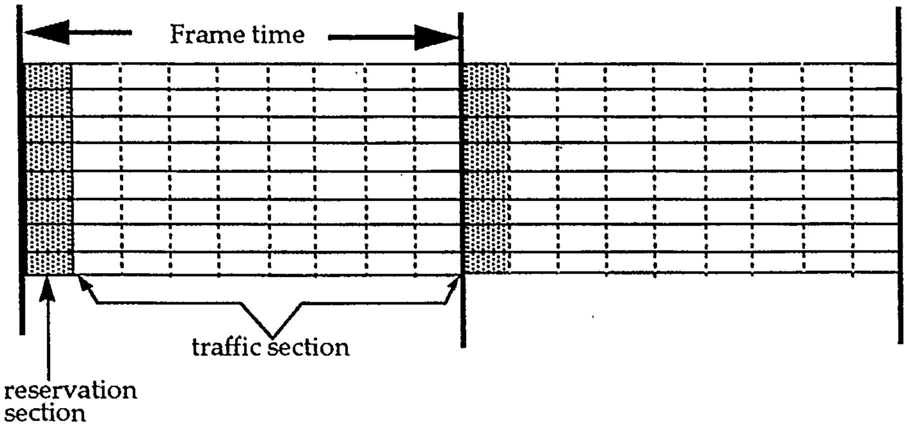


Figure 1: MF-TDMA Frame Format

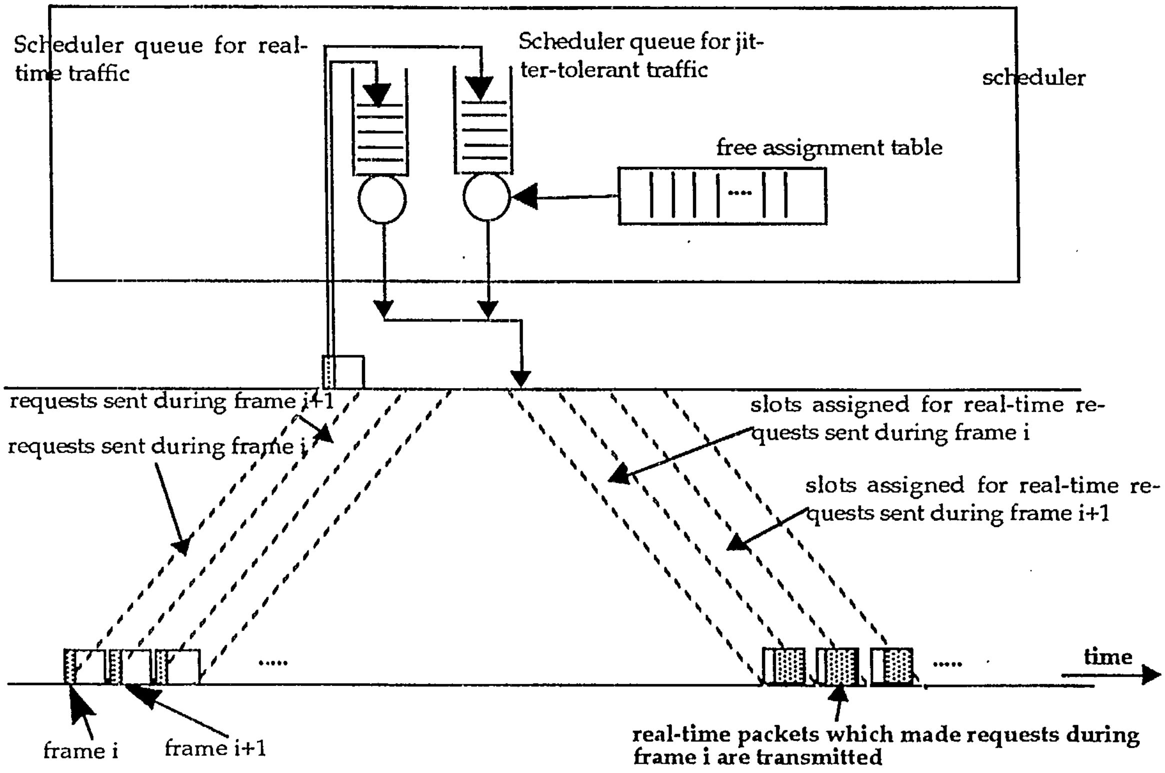

Scheduler queue for real-time traffic
Scheduler queue for jitter-tolerant traffic
scheduler
free assignment table
requests sent during frame i+1
requests sent during frame i
slots assigned for real-time requests sent during frame i
slots assigned for real-time requests sent during frame i+1
.....
.....
time
frame i
frame i+1
real-time packets which made requests during frame i are transmitted

Figure 2: Satellite and Earth Terminal Description

2.1. Possible Requesting Strategies

There are three ways in which a user may transmit its requests. These include preassigned (PA), piggybacking (PB), and random access (RA). In the PA case, the reservation slots are assigned to each of the users in a fixed manner. A user simply waits for his slot and transmits the requests at the appropriate time. In the PB scheme, the frame has no explicit reservation section. Rather, each packet has a field which it can use to make requests for additional slots. At the satellite, the scheduler examines this field for every packet, and if necessary, places the request in its scheduler queue. One of the problems with this technique deals with the initial access. How does a user continue to make requests if it has not sent a packet? For our purposes, we will assume that a user is given initial access by means of free assigned slots. For the final technique mentioned above, namely RA, all the reservation slots are available to all users. Each user which must send a request, does so by selecting one of these reservation slots randomly, and transmitting his request. Obviously, more than one user can send a reservation in the same reservation slot. This collision is resolved by allowing those packets for which the requests have collided to get through via free assigned slots. Unfortunately, the packets for real-time users can not be queued while waiting for a free assigned slot, and these will have to be blocked.

3. Model Description

The overall model for the system is shown in Figure 3. In the end, each of the individual components will be modeled in OPNET, and combined to produce a complete network simulator. In this report, only the shaded regions will be discussed, namely the traffic generator, the multiple access scheme, and the scheduler.

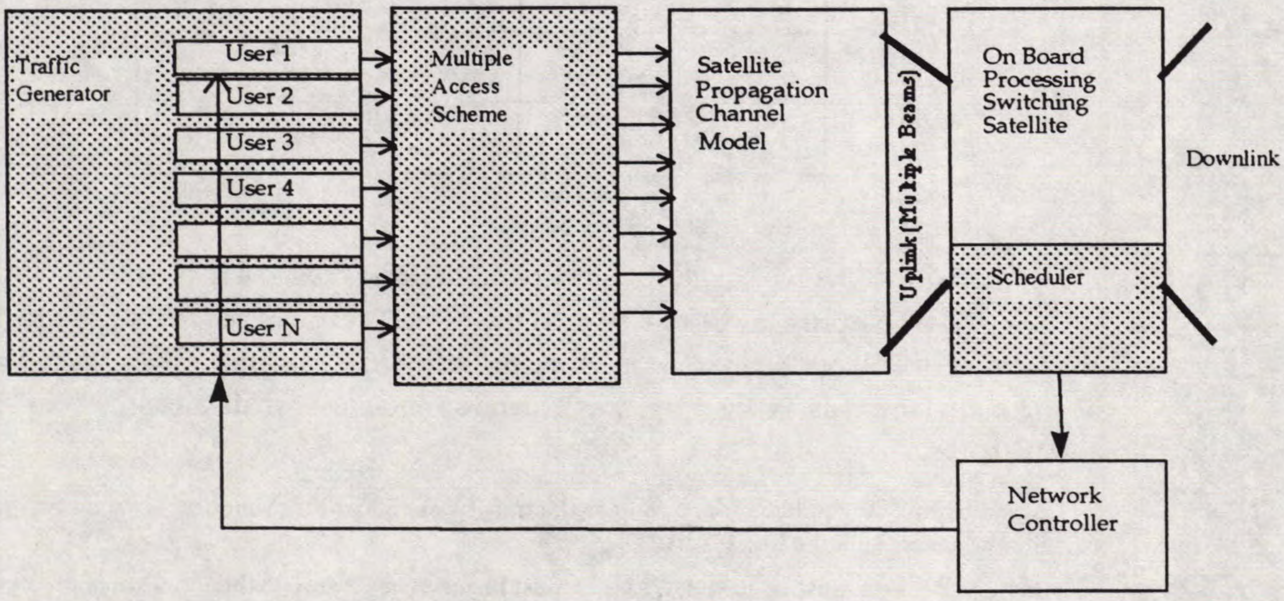


Figure 3: Network Simulator

OPNET is an event driven simulation tool which allows a communication network to be described based on the principal of hierarchical layers. At the top layer, we define the overall structure of the network we wish to simulate - this is known as the network model. For a communication system with *N* users, the network layout contains *N* earth terminals, each supporting multimedia (or ATM) traffic, and one on-board processing satellite. The case for 10 users is shown in Figure 4. For large networks, the cumbersome procedure of entering the layout manually is avoided, by using an external `C' program to generate the desired layout. This program accepts as input the number of user earth terminals (ET's), and produces the corresponding network model.

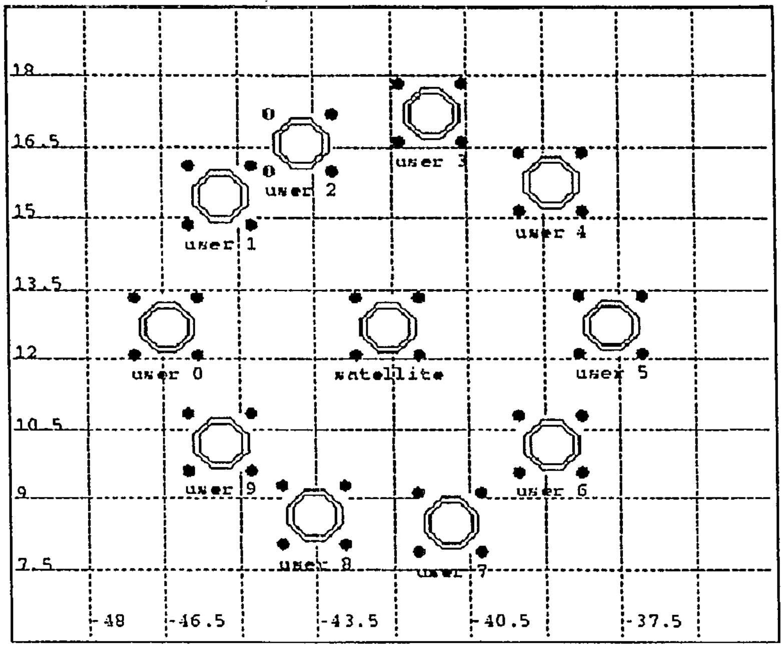


Figure 4: Network Layout

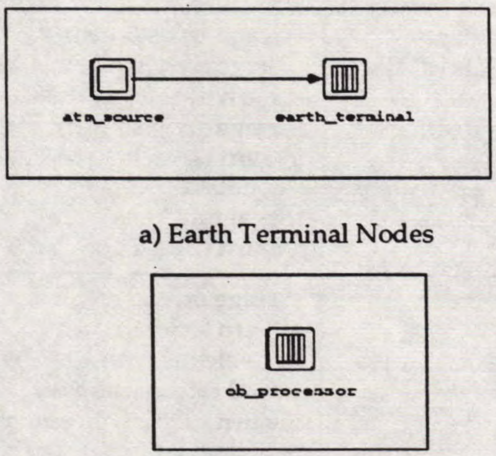
Each of the *N+1* elements of the network model is called a node. The definitions of these nodes comprise the second layer of the OPNET model. Each of these nodes is made up of processors, which perform specific functions (processes) within the network. The *N* earth terminals, for instance, are all identical and have two distinct functions to perform. That is

- packets must be generated according to the traffic generator. This function is accomplished by the processor called **atm_source.**
- arriving packets must be stored, requests must be made according to the requesting strategy, and packets must be transmitted in response to a slot assignment from the on-board satellite scheduler. These three functions are all part of the multiple access scheme, and are lumped together and performed by the processor called **earth_terminal**.

These processors are shown in Figure 5a. When the packets are generated by the **atm_source**, they immediately go to the **earth_terminal**, via the link shown.

At the satellite, the main operations to be performed include the storing of the incoming packets, and the assigning or scheduling of the slots for the upcoming frame. These functions are performed in the processor block **ob_processor** (See Figure 5b).

The next level of the OPNET hierarchy is concerned with the processes which are performed in each of the processors shown in Figure 5.



a) Earth Terminal Nodes

b) Satellite Node

Figure 5: Processors within each Node

3.1 Process Descriptions

In OPNET, the processes are defined by means of state transition diagrams. Each of the states contains the C-code to perform the desired operation. The function of all the above processors is explained below:

atm_source Process

As stated earlier, the basic function to be performed in this processor is to generate the multimedia traffic. The state transition diagram is shown in Figure 6. The process starts in state init, at the beginning of the simulation (this is denoted by the large solid arrow beside the state). The objective of this state is to initialize variables, and to prompt the user to enter the traffic models to be used for each traffic type, the number of such models, and the parameters for these models. The available models, and the parameters required are shown in Table 1.

Table 1 Traffic Models Available

Traffic Type	Models Available	Model Parameters
Data	Poisson 2 state MMPP 2 state PMPP on-off sources (Pareto on times)	average arrival rate λ average arrival rate in each state λ_1, λ_2 sojourn times in each state α_1, α_2 average arrival rate in each state λ_1, λ_2 sojourn times in each state α # of data sources arrival rate in on state sojourn times in on and off states β, α
Voice	2 state MMPP on-off sources (Exponential on times)	average arrival rate in each state λ_1, λ_2 sojourn times in each state α_1, α_2 # of voice sources arrival rate in on state sojourn times in on and off states β, α
Video	2 state MMPP on-off mini-sources	average arrival rate in each state λ_1, λ_2 sojourn times in each state α_1, α_2 # of mini-sources/ video source # of video sources arrival rate in on state sojourn times in on and off states β, α

Once this is done, the process moves to state create, which invokes the suitable number and type of processes. Essentially, this state calls other state transition diagrams, much like a computer program calls a subroutine. Once these processes have been invoked, the **atm_source** processor goes to a wait state, and stays there until the end of the simulation.

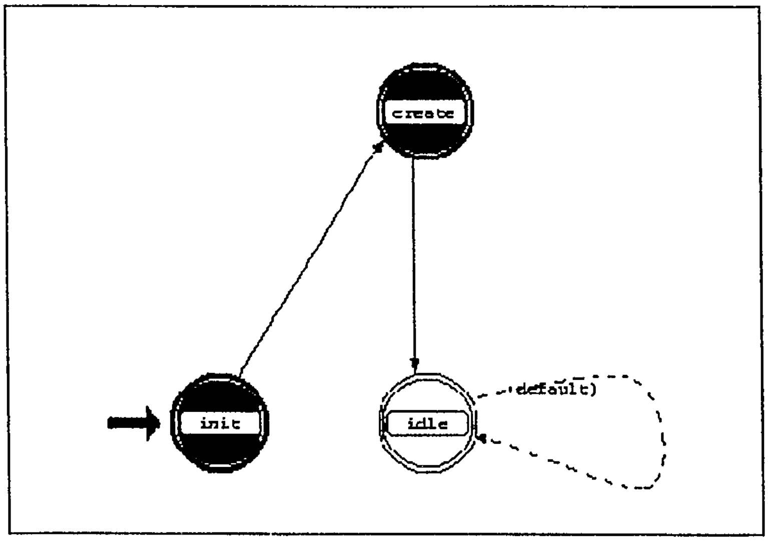


Figure 6: State Transition Diagram for **atm_source** Processor

The processes invoked by **atm_source**, are any combination of data, voice, and video sources listed in Table 1. In this report, simulation results are obtained for multimedia type traffic with voice and video generated by a Markov Modulated Poisson process (MMPP) and data by a Pareto Modulated Poisson Process (PMPP). The state transition diagrams for all of these is shown in Figure 7. Again, variable initialization is performed in state init. The 2 state process is then initially set to state 1. The program then waits in state st_1 (no action is performed in st_1). Essentially the program stalls here, until an arrival occurs, the sojourn time in the state ends, or the simulation time expires. The basic events are shown in Figure 8. The only differences between the MMPP and PMPP models is that the sojourn times in each of the states is Pareto distributed for the latter, and exponentially distributed for the former.

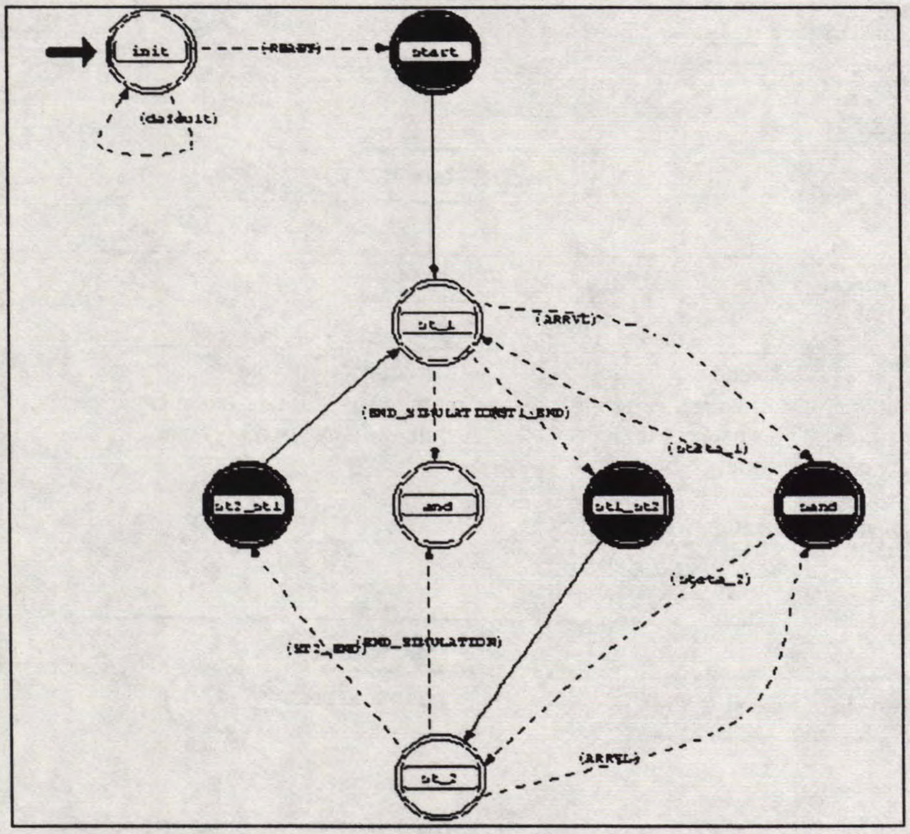


Figure 7: State Transition diagram for MMPP (and PMPP)

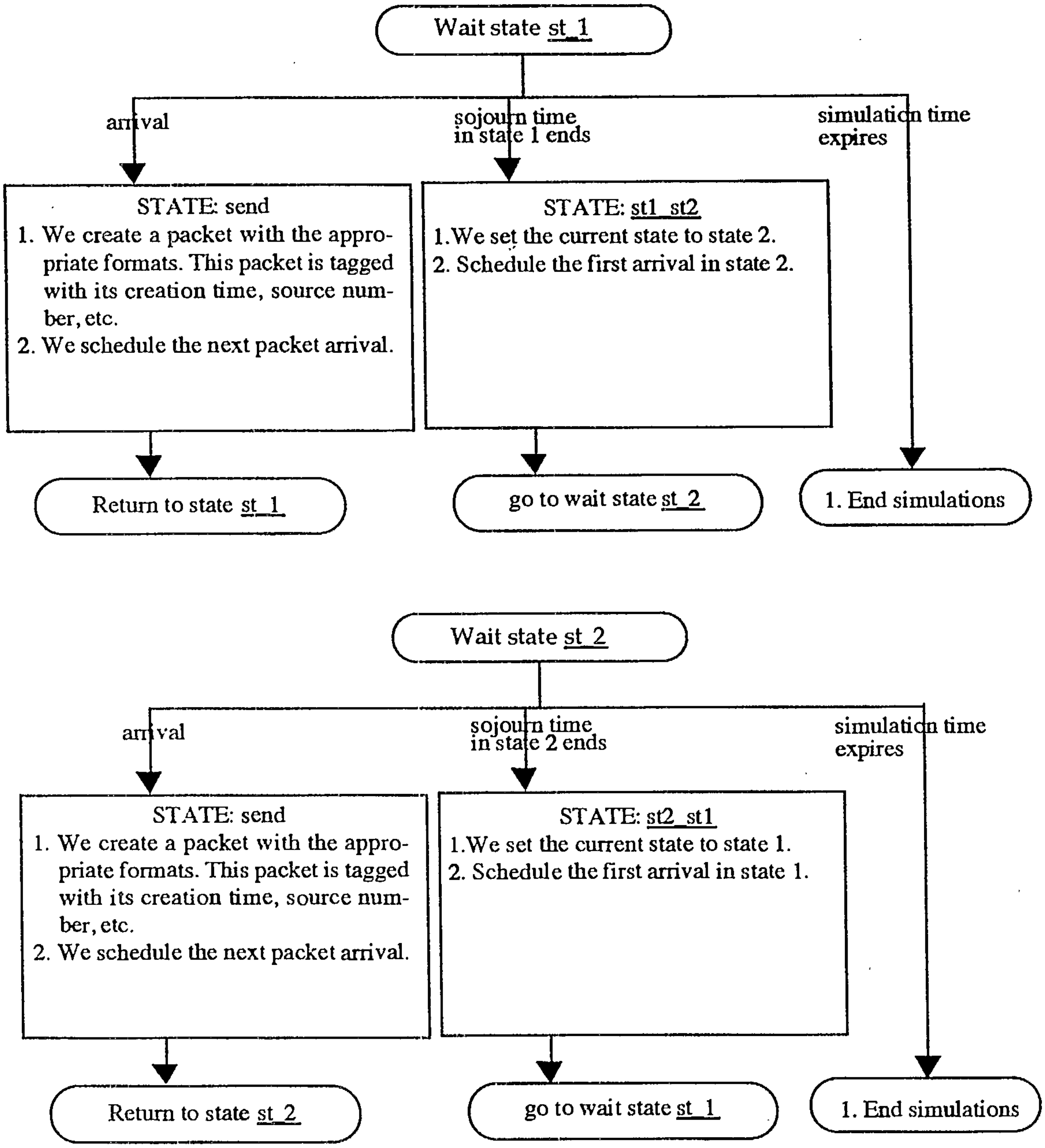

Wait state st_1
arrival
sojourn time in state 1 ends
simulation time expires
STATE: send
1. We create a packet with the appropriate formats. This packet is tagged with its creation time, source number, etc.
2. We schedule the next packet arrival.
STATE: st1_st2
1.We set the current state to state 2.
2. Schedule the first arrival in state 2.
Return to state st_1
go to wait state st_2
1. End simulations
Wait state st_2
arrival
sojourn time in state 2 ends
simulation time expires
STATE: send
1. We create a packet with the appropriate formats. This packet is tagged with its creation time, source number, etc.
2. We schedule the next packet arrival.
STATE: st2_st1
1.We set the current state to state 1.
2. Schedule the first arrival in state 1.
Return to state st_2
go to wait state st_1
1. End simulations

Figure 8: Flowchart for MMPP (and PMPP)

The reader can find a description of the on-off model in [4].

earth_terminal Process

As the packets are created in the **atm_source** processor they are immediately transferred to the **earth_terminal** processor. As discussed earlier, this processor has three main functions:

1. queue arriving packets
2. send requests when a reservation slot is available to the earth terminal, and
3. transmit packets when the scheduler has assigned a slot to the earth terminal.

The state transition diagram is shown in Figure 9.

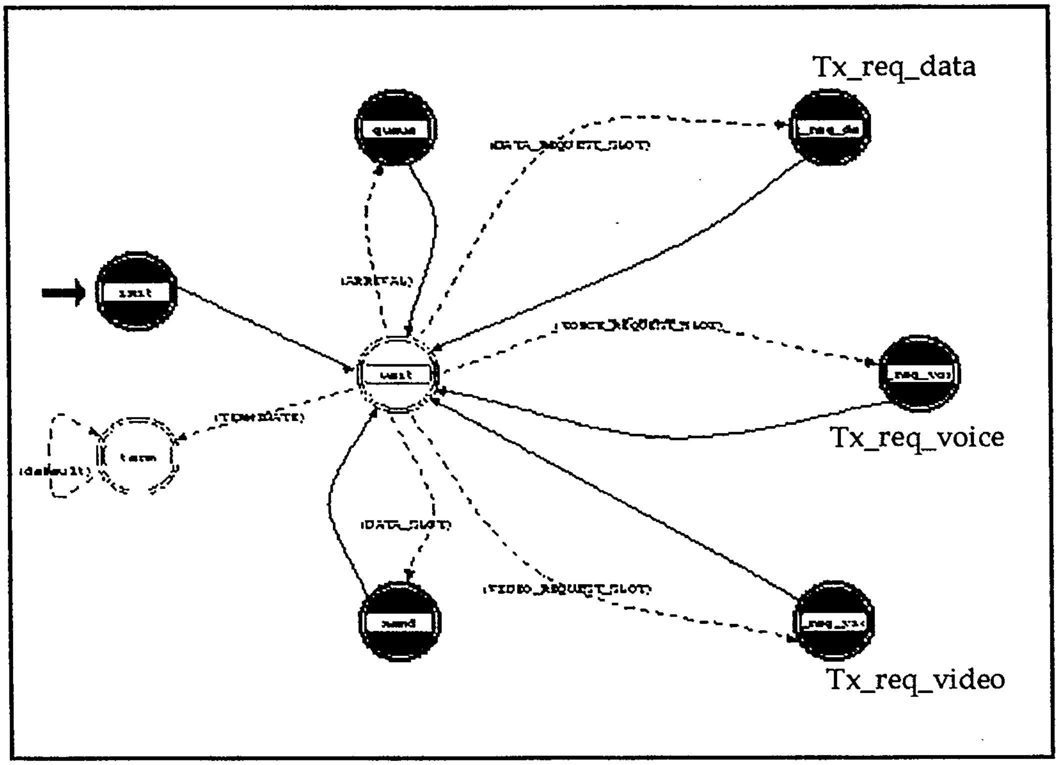


Figure 9: State Transition diagram for earth_terminal Process

As before, the process starts in state <u>init</u>, where the various variables are initialized, and the user is prompted to enter the number of slots available per frame (no_slots_per_frame), the number of frequency carriers (no_carriers), and the reservation channel capacity (overhead used for the reservation slots). From the number of slots available, the program calculates the number of slots per carrier (no_slots_per_carrier).

no_slots_per_carrier = no_slots_per_frame/no_carriers.

Since a single earth terminal cannot transmit on two carriers simultaneously (in order to maximize transmitter output power), the number of slots any earth terminal can be given per frame is limited to no_slots_per_carrier. If an earth terminal has requested more than this number, the packets making the request are lost if they are voice or video, or delayed if they are jitter-tolerant.

After the init state, the process moves on to the wait state, where it pauses until one of three events occurs. For a general description, see Figure 10.

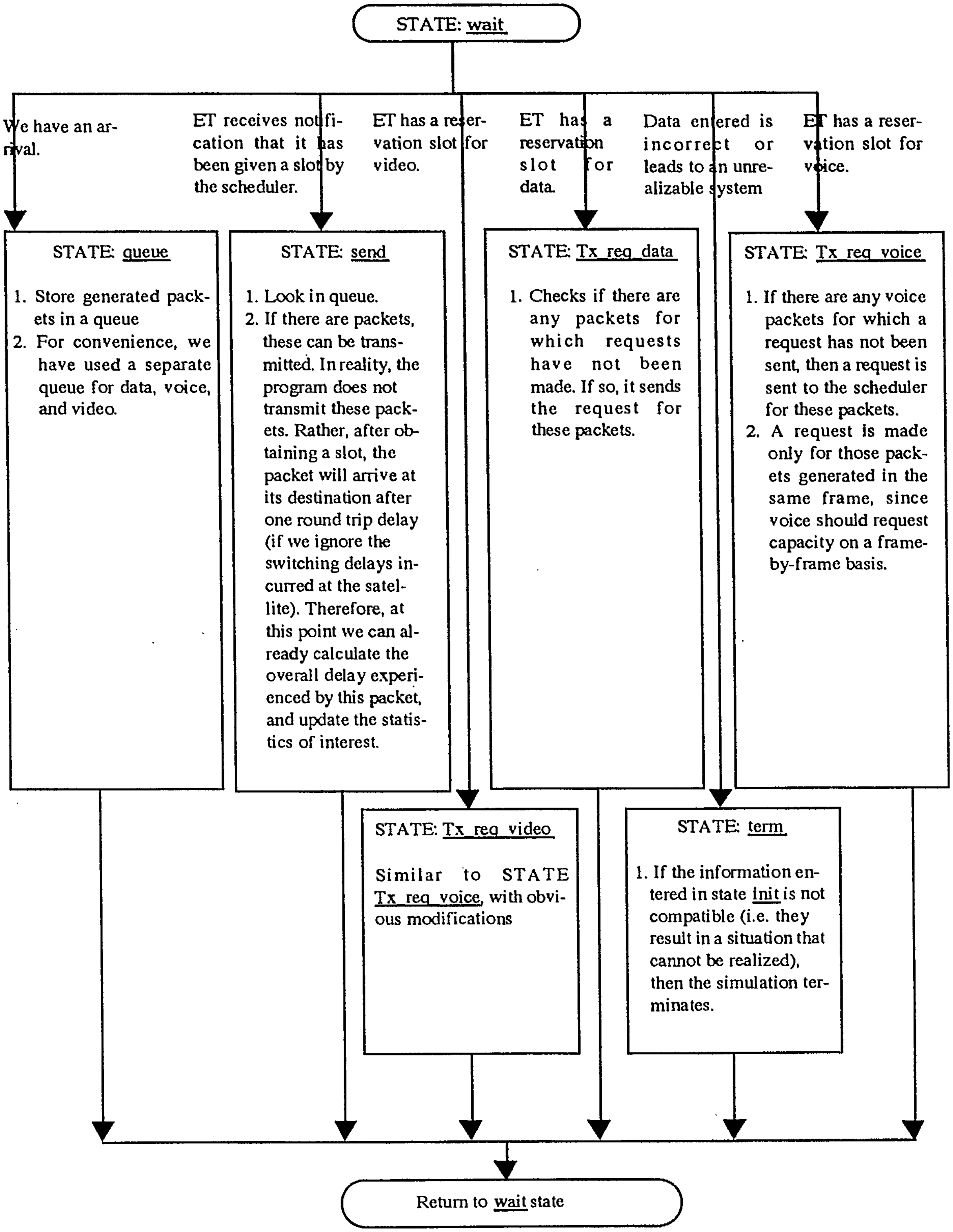

STATE: wait
We have an arrival.
ET receives notification that it has been given a slot by the scheduler.
ET has a reservation slot for video.
ET has a reservation slot for data.
Data entered is incorrect or leads to an unrealizable system
ET has a reservation slot for voice.
STATE: queue
1. Store generated packets in a queue
2. For convenience, we have used a separate queue for data, voice, and video.
STATE: send
1. Look in queue.
2. If there are packets, these can be transmitted. In reality, the program does not transmit these packets. Rather, after obtaining a slot, the packet will arrive at its destination after one round trip delay (if we ignore the switching delays incurred at the satellite). Therefore, at this point we can already calculate the overall delay experienced by this packet, and update the statistics of interest.
STATE: Tx_req_data
1. Checks if there are any packets for which requests have not been made. If so, it sends the request for these packets.
STATE: Tx_req_voice
1. If there are any voice packets for which a request has not been sent, then a request is sent to the scheduler for these packets.
2. A request is made only for those packets generated in the same frame, since voice should request capacity on a frame-by-frame basis.
STATE: Tx_req_video
Similar to STATE Tx_req_voice, with obvious modifications
STATE: term
1. If the information entered in state init is not compatible (i.e. they result in a situation that cannot be realized), then the simulation terminates.
Return to wait state

Figure 10: Flowchart for **earth_terminal** Process

scheduler Process

The final processor to consider is the scheduler processor in the **ob_processor** node. Generally, the scheduler has two functions:

1. To place requests in a scheduler queue.
2. To assign slots based on these requests.

However, we have augmented the function of this process to include:

a. inform the users when they have a reservation slot. In reality, for PA, all users know exactly when their reservation slots occur. Nonetheless, it proves simpler to allow the scheduler to inform the earth terminals that they have an upcoming reservation slot, since this is done once every frame, similar to the slot assignment. Therefore, at the beginning of every frame, the scheduler tells the appropriate earth terminals that they have a reservation slot.

b. write out the statistics observed to a file at the end of the simulation (after the duration of the simulation has elapsed).

The state transition diagram is shown in Figure 11. Again, the process starts in state init, in order to initialize variables. Then it immediately goes to states res_slot_as and data_slot_as. In the former, we notify the earth terminals that they will have a reservation slot in 0.135 sec. (round trip delay/2). In the latter state, the available slots are assigned by the scheduler. Currently, two different techniques have been employed for slot assignment. In the first scheduler, known as Scheduler 1, a hard decision is made as to the number of slots to assign to an earth terminal. A threshold for video and for voice is agreed upon during the call set-up phase. This threshold is based on the average number of arrivals expected. For example, if an earth terminal expects to receive, on average, 0.35 voice cells per frame and 0.05 video cells per frame, a suitable threshold may be 1 voice and video cell per frame. For this case, every earth terminal is given up to a maximum of 1 cell per frame for both voice and video (a maximum of 2 real-time cells per frame). This is done to ensure that no one user can overload the system by transmitting much more than his average. Figure 12 shows an example of how the slot assignment is done. The threshold is set to 1, as discussed above, and the number of users is 973.

Scheduler 1 is somewhat pessimistic, as is shown by the simulation results in section 5. Notice that an earth terminal's real time packets may be blocked even if there is unused capacity in the MF-TDMA frame. This is because the slot assignment for each earth terminal is done independently of the status of the other earth terminals. Basically, if one earth terminal requires no slots for real-time packets, then his unused capacity can be given to another user. For the case we have been considering, this other user would then be able to send 4 real-time packets per frame (2 each for voice and video). This is the main reason for considering the modified scheduler, which we refer to as Scheduler 2.

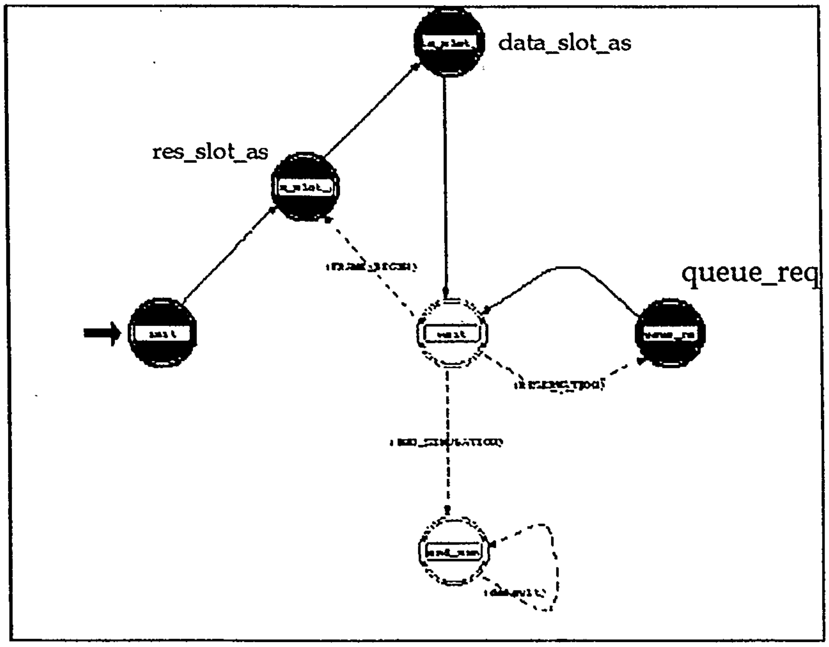


Figure 11: State Transition Diagram for scheduler Process

In this modified scheduler, slots are assigned based on the global state of all user requests. Essentially, four passes are performed over the scheduler queue, for every frame.

- Pass1: The scheduler determines how many real-time channels are required for all earth terminals.(i.e. no_vid_cells_required, no_voice_cells_required).
- Pass2: If no_vid_cells_required+no_voice_cells_required <= no_slots_per_frame, then all real-time requests are satisfied. Otherwise, only no_slots_per_frame slots are assigned (the maximum), in a round robin fashion.
- Pass3: If there are still empty channels after Pass2, then these are assigned to the data packets whose requests are in the scheduler queue. This assignment is on a first-come first-served basis.
- Pass4: If empty slots remain after Pass3, then these are freely assigned to earth terminals. By the time these slots 'arrive' at the earth terminals, it is possible that the ET's have a data packet in their queue which can use this slot. This packet would then be transmitted without having to wait for its request to be honored.

This scheduling strategy is expected to reduce the loss probability substantially, but as a result of more real-time packets getting through per frame, there will be less slots for data, and the data delay is expected to go up somewhat.

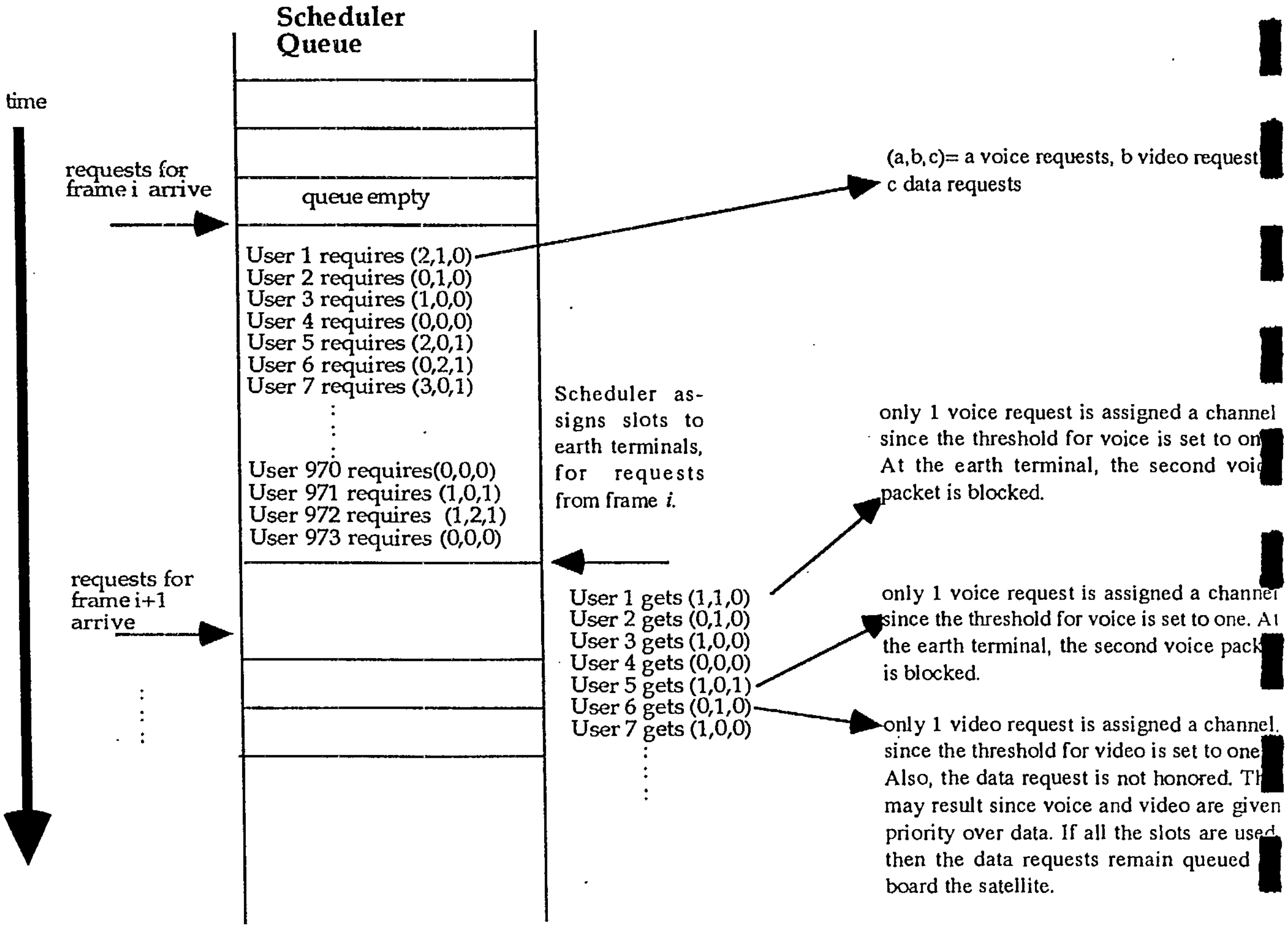
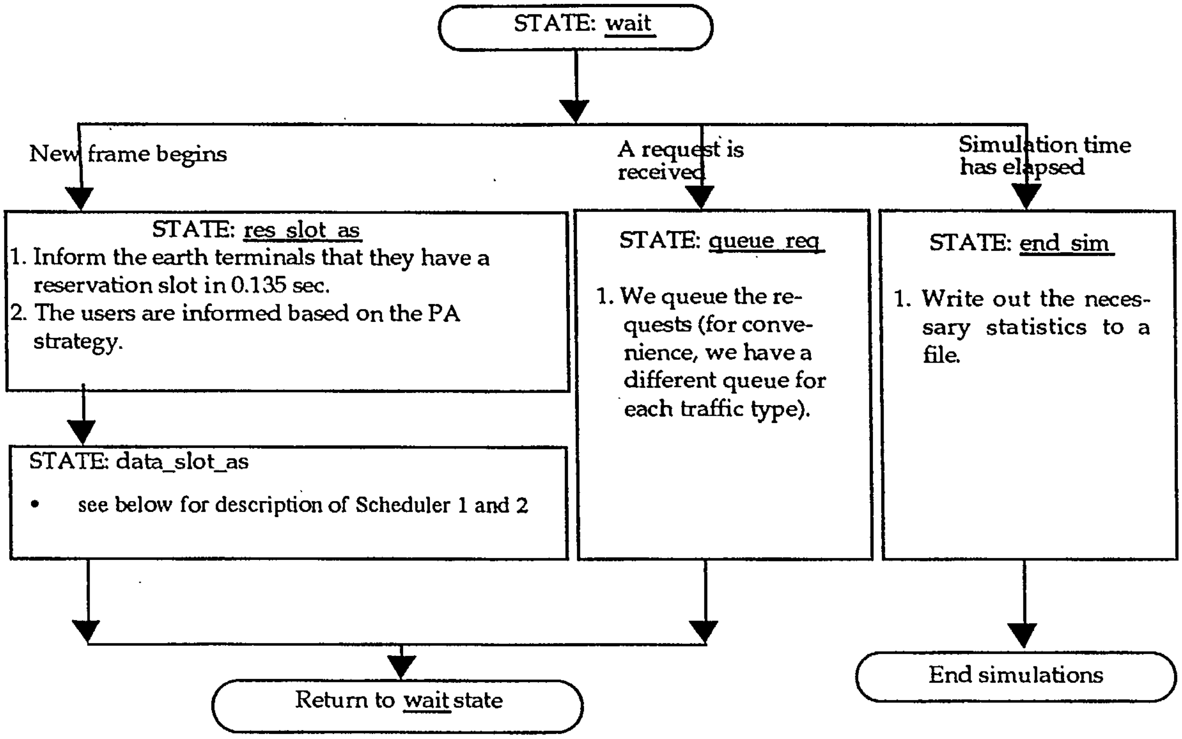

Scheduler
Queue
time
requests for
frame i arrive
queue empty
User 1 requires (2,1,0)
User 2 requires (0,1,0)
User 3 requires (1,0,0)
User 4 requires (0,0,0)
User 5 requires (2,0,1)
User 6 requires (0,2,1)
User 7 requires (3,0,1)
User 970 requires(0,0,0)
User 971 requires (1,0,1)
User 972 requires (1,2,1)
User 973 requires (0,0,0)
requests for
frame i+1
arrive
(a,b,c)= a voice requests, b video request
c data requests
Scheduler assigns slots to earth terminals, for requests from frame i.
User 1 gets (1,1,0)
User 2 gets (0,1,0)
User 3 gets (1,0,0)
User 4 gets (0,0,0)
User 5 gets (1,0,1)
User 6 gets (0,1,0)
User 7 gets (1,0,0)
only 1 voice request is assigned a channel since the threshold for voice is set to on At the earth terminal, the second voi packet is blocked.
only 1 voice request is assigned a channel since the threshold for voice is set to one. At the earth terminal, the second voice pack is blocked.
only 1 video request is assigned a channel. since the threshold for video is set to one Also, the data request is not honored. Th may result since voice and video are given priority over data. If all the slots are used then the data requests remain queued board the satellite.

Figure 12: Example of Scheduler 1

Naturally there are no requests in the scheduler queue at the start of the simulation, and so all channels are free assigned to the earth terminals, in a round robin manner. The process then pauses in state wait, and stays there until one of the following events occurs: the next frame begins; a request is received; or the simulation is over. These are explained in Figure 13. Both methods of slot assignment are considered



STATE: res slot as

1. Inform the earth terminals that they have a reservation slot in 0.135 sec.
2. The users are informed based on the PA strategy.

STATE: queue req

1. We queue the requests (for convenience, we have a different queue for each traffic type).

STATE: end sim

1. Write out the necessary statistics to a file.

STATE: data_slot_as

- see below for description of Scheduler 1 and 2

STATE: data slot as Scheduler 1

1. Initially, we check if there are any requests from video packets. If so, these are assigned, up to the maximum allowed (threshold for video).
2. The same is done for the requests coming from the voice packets. Slots are assigned up to the maximum allowed (threshold for voice).
3. If any requests from video or voice cannot be honored, this implies that the packets at the earth terminals will be blocked. Consequently, we update the statistic monitoring the number of cells (or packets) lost.
4. We also keep track of how many slots are still available, and how many slots have been assigned to each earth terminal (respectively no_slots_left and open_cells[earth_terminal]).
5. We check the queue for requests from data packets and service these on a first-come first-served basis, decrementing no_slots_left and open_cells[earth_terminal] as we go along.
6. If a request at the head of this queue cannot be served because an earth terminal has already been assigned the maximum number of slots per frame (no_slots_per_carrier), then this request is ignored, and we look at requests following this one.
7. If after serving all requests (video, voice, and data) there are still slots available, these are freely assigned to earth terminals in a round robin manner. Again the restriction that no earth terminal transmit more than no_slots_per_carrier is observed.

STATE: data slot as Scheduler 2

1. Add up all the requests from the video and voice packets. If this total is less than no_slots_per_frame, then all requests are assigned a slot.
2. If the total is larger than no_slots_per_frame then only the first no_slots_per_frame requests are assigned a slot. The assignment of slots to the first no_slots_per_frame is done in a round robin fashion, so that no ET is favored from frame to frame.
3. If any requests from video and voice packets cannot be serviced, those packets will be blocked at the earth terminals. The statistic monitoring the number of cells lost is updated.
4. We also keep track of how many slots are still available, and how many slots have been assigned to each earth terminal (respectively no_slots_left and open_cells[earth_terminal]).
5. We check the queue for requests from data packets and service these on a first-come first-served basis, decrementing no_slots_left and open_cells[earth_terminal] as we go along.
6. If a request at the head of this queue cannot be served because an earth terminal has already been assigned the maximum number of slots per frame (no_slots_per_carrier), then this request is ignored, and we look at requests following this one.
7. If after serving all requests (video, voice, and data) there are still slots available, these are freely assigned to earth terminals in a round robin manner. Again the restriction that no earth terminal transmit more than no_slots_per_carrier is observed.

Figure 13: Flowchart for **scheduler** Process

4. Illustrative Example

The example considered to illustrate the performance of MF_TDMA with CF-DAMA_PA is based on the scenarios of the advanced SATCOM system. The parameters are listed below:

1. MF-TDMA frame capacity = 8.192 Mb/s (512 slots of 48 bytes each).
2. Up and downlink frame duration = 24 ms.
3. MF-TDMA frame composition:

- 8 carriers of 1.024 Mb/s.

Figure 14 below shows a typical MF-TDMA frame with no overhead. In these results, overhead will not be considered, so that maximum channel utility is 1.

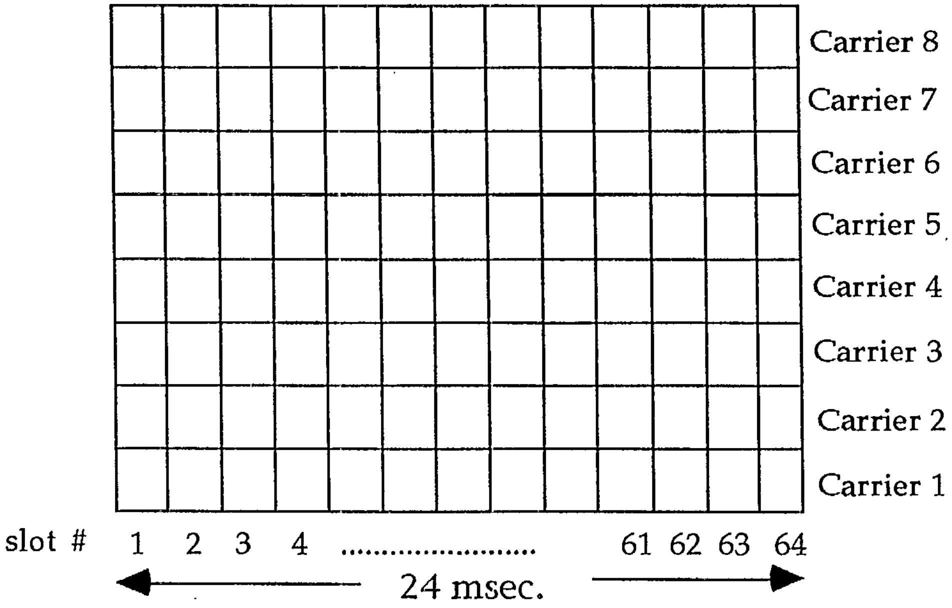


Figure 14: MF-TDMA frame

5. Multi-frame duration = 16 frames or 384 ms.
6. Switch port capacity = 4 x MF-TDMA frame capacity or 32.768 Mb/s.
7. Buffer memory size / port = 0.9 Mbytes or 18750 cells of 48 bytes each.
8. Maximum number of terminals (for 100% load) per MF-TDMA frame = 1024.
9. Load per user is fixed, and total system load is varied by increasing the number of terminals. The load points considered are highlighted in Table 2.

Table 2 Load Points Considered

Load	Number of Terminals	Maximum number of cells/ terminal/frame
1.0	1024	0.5
0.95	973	0.53
0.8	820	0.62
0.5	512	1.0
0.2	205	2.5

10. Four cases of aggregated source traffic are considered in the simulations. These are shown in Table 3.

Table 3 Parameters of Aggregated Sources

	Case 1 Voice Dominant	Case 2 Video Dominant	Case 3 Equal Load	Case 4 Data Dominant
% of voice	70	10	33.3	20
% of video	10	70	33.3	10
% of data	20	20	33.3	70

Table 4 below shows the characteristics of the individual traffic sources.

Table 4 Parameters of Individual Sources

	Peak to Average ratio	Peak Rate (Kbps)	Average rate (Kbps)	Average rate (cells/frame)
Voice	2.5	64	25.6	16
Video	5	384	76.8	4.8
Data	200	128	0.64	0.04

11. Data is generated by a PMPP source with a Hurst parameter 0.8. Voice and video are generated by MMPP sources, whose parameters are determined by mapping the aggregate voice and video into 2-state MMPP's [5]. The results of the mapping produces a sojourn time and an average arrival rate in each of the states $(\alpha_1, \lambda_2, \alpha_1, \lambda_2)$.

5. Simulation Results

The simulation results are shown in Figures 15to 18. Figure 15 shows the data cell delay for scheduler 1, for the four traffic mixes. As expected, as the percentage of data increases, the data delay also increases. Note also that the results for the voice and video dominant cases yield similar data delay performance (curves are overlapping). Figure 16 shows the actual number of real-time packets lost, as a result of blocking, for Scheduler 1. This figure of merit is used, as opposed to the more traditional loss probability, since for Scheduler 1, the loss probability is constant for all loads. As the load increases, the number of packets lost increases linearly (as seen in Figure 16), but the number of packets transmitted also increases linearly, resulting in a constant ratio. This phenomenon is attributed directly to the scheduler. With increasing load, the only parameter that changes is the number of terminals. Since this scheduler honors requests for each of these terminals independently, it makes no difference whether there are 205 terminals (0.2 load) or 973 terminals (0.95 load). The actual values of the loss probabilities for the four traffic mixes, are shown in Table 5. Note that this loss probability, can also be understood as a blocking rate.

Figure 17 shows the data delay results for Scheduler 2. Again the data delay is the same for both the video and voice dominant cases. For the simulation times observed (order of 10^6 packets), no packets were lost while using this scheduler. As a result, we can conclude that the loss probability is less than 10^{-4}, for all cases considered.

Figure 18 shows a comparison of the data cell delay, for the voice dominant case for the two schedulers. As expected, Scheduler 1 results in lower delay since the threshold imposed by the scheduler allows more data requests to be honored, and more channels to be free assigned. However, the price paid is the very high loss probability. In fact, the gain is so small that we conclude Scheduler 2 is more advantageous.

Table 5 Loss Probability for the four Traffic Mixes for Scheduler 1

Traffic Mix	Loss Probability
Voice Dominant	0.135
Video Dominant	0.135
Equal Mix	0.077
Data Dominant	0.040

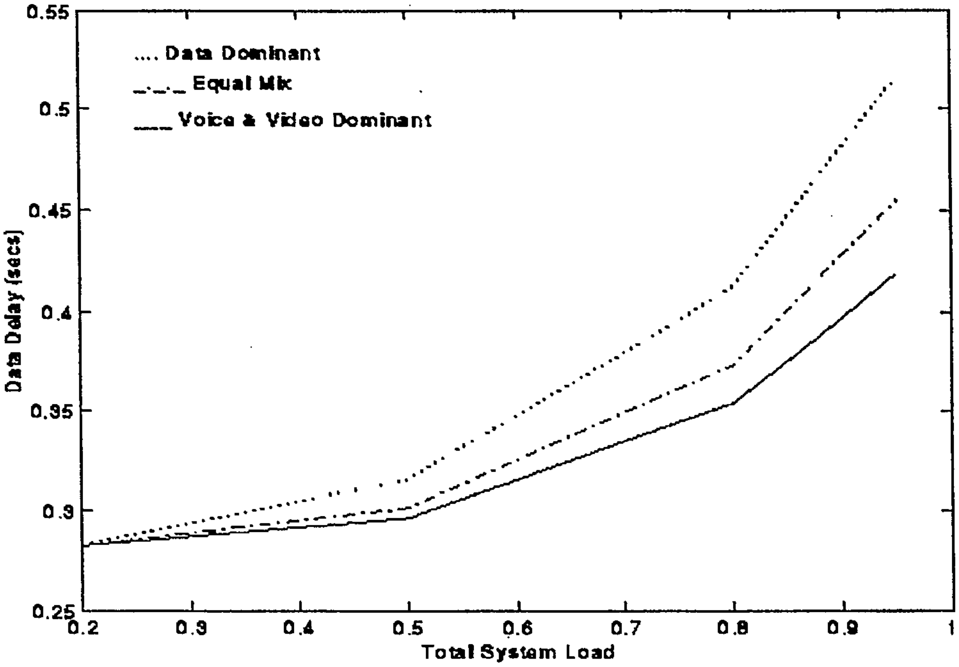


Figure 15: Data Cell Delay for Scheduler 1

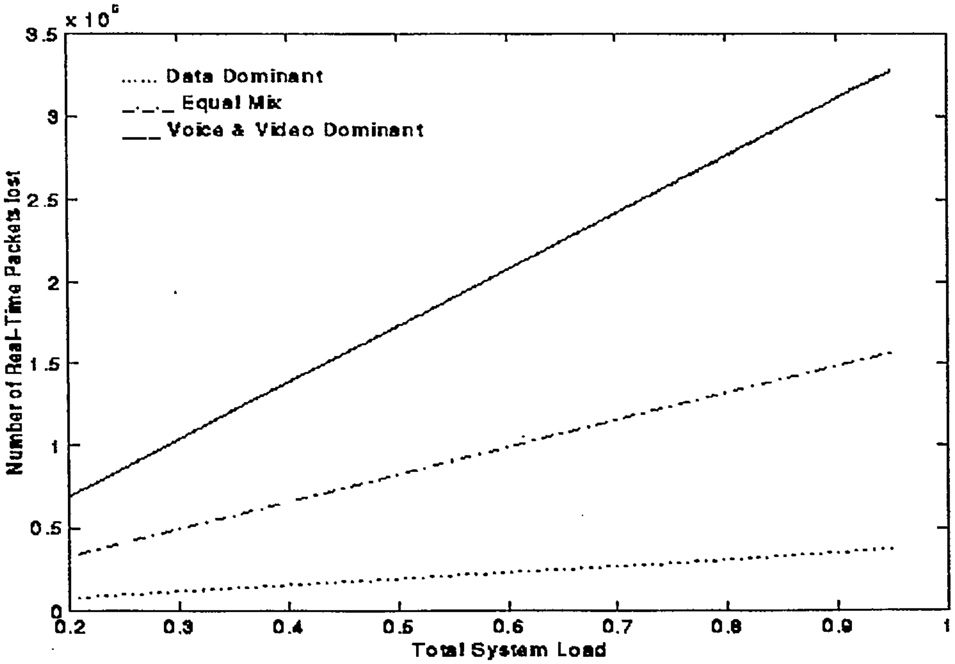


Figure 16: Number of Real-Time cells Lost for Scheduler 1

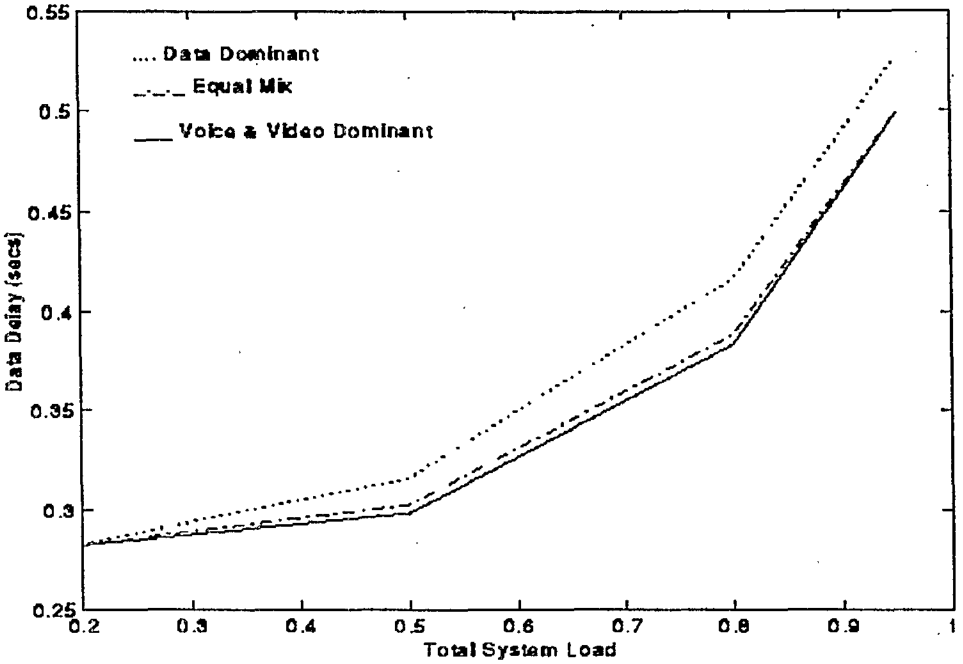


Figure 17: Data Cell Delay for Scheduler 2

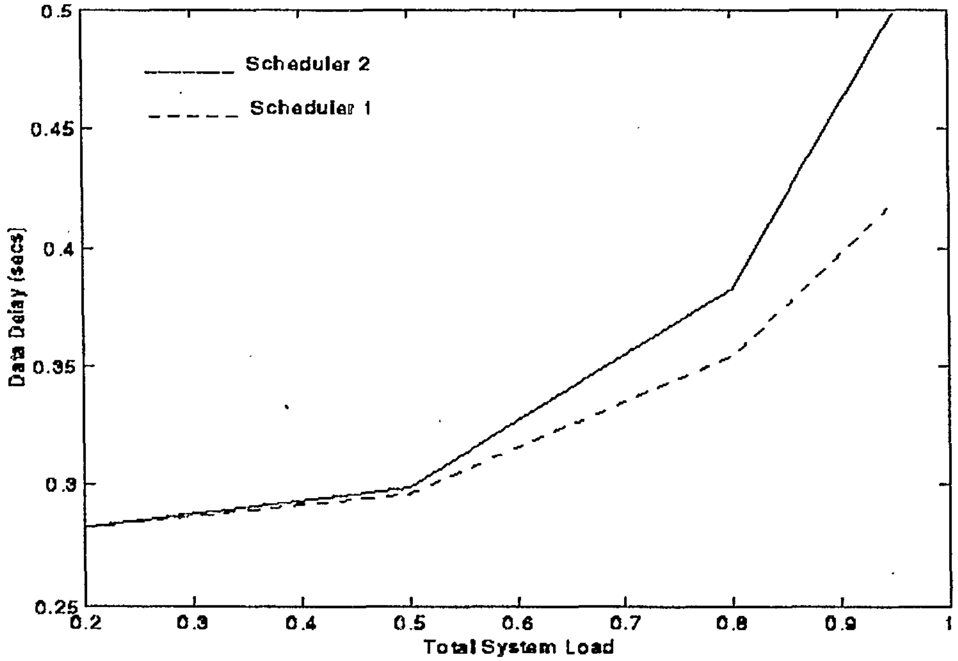


Figure 18: Comparison of Data Cell Delay for Voice Dominant Case

6. References

[1] H.W. Lee and J.W. Mark, "Combined Random/Reservation Access for Packet Switched Transmission over a satellite with On-Board Processing: Part I-Global Beam Satellite," *IEEE Transactions on Communications*, Vol. 31, No. 10, pp. 1161-1171, October 1983.

[2] H. Ahmadi and T.E. Stern, "A New Satellite Multiple Access Technique for Packet Switching Using Combined Fixed and Demand Assignment," *NTC Conference Proceedings*, p. 70.4.1-70.4.3, 1980.

[3] T. Le-Ngoc and S.V, Krishnamurthy, "Performance of Combined Free/Demand Assignment Multiple-Access Schemes in Satellite Communications," *International Journal of Satellite Communications*, 1996.

[4] S. Subramanian, *Traffic Generator User Manual*, Report: Concordia University, 1995.

[5] H. Heffes and D.M Lucantoni, "A Markov Modulated Characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer Performance," *IEEE Journal on Selected areas in Communications*, Vol. 4, No. 6, pp. 856-868, September 1986.