Delay-jitter Bound and Statistical Loss Bound for Heterogeneous Correlated Traffic — Architecture and Equivalent Bandwidth*

A. Adas
Department of Electrical Engineering
Georgia Institute of Technology
Atlanta, GA 30332

Amarnath Mukherjee
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332

LAST REVISED: April 29, 1996

GIT-CC-96-05

Abstract

Network support for variable bit-rate video needs to consider (*i*) properties of workload induced (e.g., significant auto-correlations into far lags and different marginal distributions among connections) and (*ii*) application specific bounds on delay-jitter and statistical cell-loss probabilities. The objective of this paper is to present a quality-of-service solution for such traffic at each multiplexing point in the network. Heterogeneity in both offered workload and quality-of-service needs are addressed.

The paper has several parts: (*i*) motivation for the proposed architecture, including an empirical model for selected queue statistics when the input process is a fractionally-differenced ARIMA$(1, d, 0)$ process and the server is work conserving; (*ii*) a framing-strategy with active cell-discard to address the impact of inter-frame dependencies that would otherwise increase queue lengths, while simultaneously providing delay-jitter bounds; (*iii*) a pseudo earliest-due-date cell dispatcher for resolving competing deadlines, addressing cell-loss bounds and fairness across virtual circuits, and maximizing output-channel efficiency; (*iv*) upper bounds on equivalent bandwidth needed for heterogeneous delay-jitter bounds and heterogeneous cell-loss probabilities for traffic with heterogeneous arrival statistics.

This paper assimilates and builds on the results of a number of authors, notably those of Golestani (Stop-and-Go framing), Garrett and Willinger and Pancha and El Zarki (variable bit-rate video traffic models and related multiplexor performance studies), and Yang and Pan (optimal space-conserving loss schedulers).

*This research was supported by the National Science Foundation under grant NCR-9396299, and by Hitachi Telecom USA, Inc. (HITEL). Results in Sections 2 and 3.2 were presented at the IEEE INFOCOM Conference in April 1995.

Contents

1 Introduction **3**

2 Queue Simulations With Controlled F-ARIMA$(1, d, 0)$ Processes **5**
- 2.1 The Fractionally-Differenced ARIMA Process 5
- 2.2 Performance of work-conserving servers with infinite buffers 6
  - 2.2.1 Empirical model for queue statistics with infinite buffers 7
  - 2.2.2 Impact of coefficient of variation 10
- 2.3 Performance of work-conserving servers with finite buffers and lazy cell discard . . . 10

3 Proposed Architecture **11**
- 3.1 Need for Stop-and-Go framing with active cell discard 11
- 3.2 Implementation of framing and active cell discard 12
- 3.3 Cell Dispatcher 14

4 Equivalent Bandwidth **17**
- 4.1 Equivalent bandwidth for homogeneous delay-jitter bounds 17
  - 4.1.1 Scenario 1: equal cell-loss probability requirements 17
  - 4.1.2 Scenario 2: heterogeneous cell-loss probability requirements 18
  - 4.1.3 Numerical results and simulation 20
- 4.2 Equivalent bandwidth for heterogeneous frames 21

5 Summary and Conclusions **21**

A Background On Short and Long Memory Processes **23**
- A.1 Self-Similarity 24

1 Introduction

Objective

The objective of this paper is to present a solution for providing (i) heterogeneous delay-jitter bounds, and (ii) heterogeneous statistical cell-loss bounds to different classes of variable bit-rate real-time traffic at individual multiplexing points. The network is assumed to be cell-switched with virtual circuits similar to the Asynchronous Transfer Mode. Detailed specifications of the ATM protocol are, however, not necessary for functioning of the mechanisms proposed.

Correlated traffic and implications

Recent studies [3, 8, 9, 27] on a range of video applications indicate that there exists a slowly decaying auto-correlation structure in the underlying stochastic processes that constitute VBR video applications. It appears that different compression schemes can change this correlation structure somewhat; however, the dominant dependence structures imposed by applications are likely to persist. Studies in [5, 9] have suggested that VBR video can be modeled as a fractionally differenced autoregressive integrated moving average process.

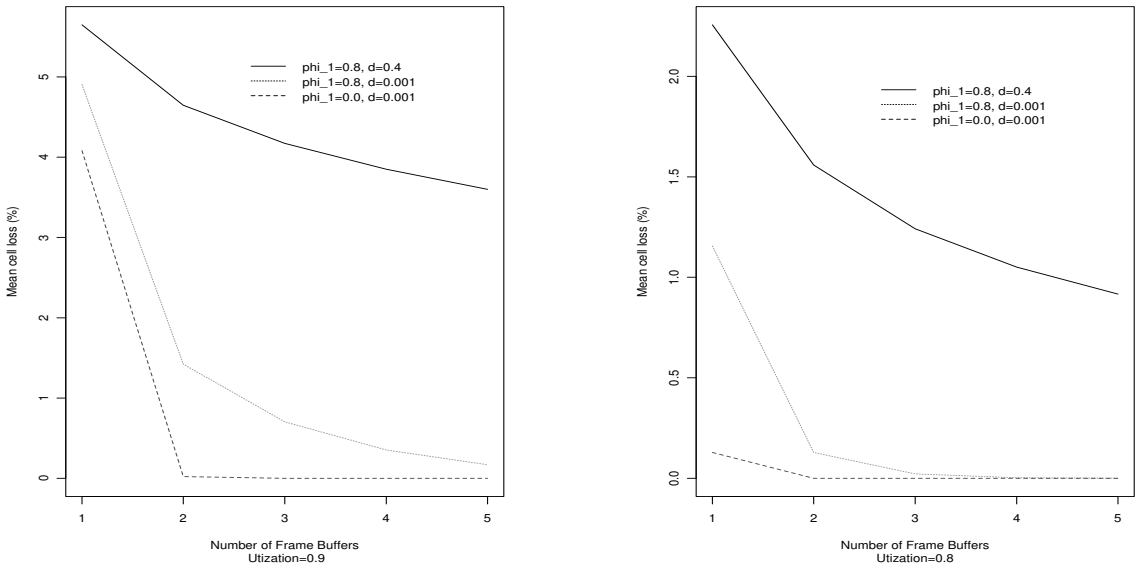
A significant fraction of VBR video applications are expected to require guaranteed maximum-delay and delay-jitter bound for cells that are delivered, and low cell-loss probabilities. Further, quality of service guarantees for different applications are likely to belong to one of several classes. In order to provide dependable quality of service, a multiplexing point in the network must have the capacity available when it is needed. Providing quality of service guarantees for real-time traffic is an active area of research, see for instance [7, 11, 18, 32]. However, providing guarantees on maximum-delay and delay-jitter performance in the presence of a slowly decaying auto-correlation structure in traffic is non-trivial, especially if the coefficient of variation of the marginal distribution (or the distribution tail) is large. This is because such traffic significantly increases queue length statistics at a multiplexor [1, 6, 9, 19, 20, 21, 26].

With the help of a fractionally differenced ARIMA$(1, d, 0)$ model for the input process, we study through simulations, the queue length process at a work-conserving server with infinite and finite buffers and varying levels of mean server utilization. A fractional-ARIMA model, with $0 < d < 1/2$, allows for the injection of controlled intensity of persistence in the arrival process. An empirical study of the mean queue length, its standard deviation, and quantiles at the tail of the queue length distribution with infinite buffers shows that the ratio of a queue statistic with dependence to that without dependence is proportional to a product of exponentials of the form $e^{c_1 \phi_1}\ e^{c_2 d}$, $(c_1, c_2 > 1)$, where ϕ_1 is an auto-regressive component (therefore, short-memory), and d is a long-memory component. In all our experiments, the value of c_2 was greater than c_1.

Issues in heterogeneous delay-jitter bounds and heterogeneous cell-loss bounds

Figure 1 shows mean cell loss versus number of frame-buffers for three different traffic correlation structures with approximately the same coefficient of variation (~ 0.24). A frame-buffer is the maximum number of cells that can be transmitted by the output channel in the corresponding time interval. The solid line which shows a slow decay is for a representative long-memory sequence. The dashed line in the middle is for traffic with short memory. The line with the smallest mean cell loss corresponds to a white noise (uncorrelated) input stream.

Since (i) for long-memory traffic with reasonable coefficient of variation (> 0.1), mean cell-loss decays slowly with increasing buffer size, and (ii) the corresponding applications also require strict delay-jitter bounds, we adopt Golestani's Stop-and-Go Queuing strategy [11, 12] with enhancements



(a) (b)

Figure 1: Mean number of cells dropped for different dependency structures in the workload. (a) Mean utilization = 0.9; (b) Mean utilization = 0.8. The model form used is an ARIMA$(1, d, 0)$ process: $(1 - \phi_1 B)\Delta^d X_t = \epsilon_t$, where the operator B is the backward shift operator: $BX_t = X_{t-1}$. ϕ_1 represents an exponentially decaying auto-correlation structure ($0 < \phi_1 < 1$). Super-imposed on it is d which represents a hyperbolic decay in the auto-correlation structure ($0 < d < 1/2$ to ensure stationarity, see [15]). $\{\epsilon_t\}$ is a Gaussian white noise process. A larger d corresponds to stronger persistence (long-range dependence).

to be mentioned shortly. Stop-and-Go Queuing decrees that cells from a given virtual circuit, arriving in a given time interval (frame) of length T, be buffered and not transmitted until the beginning of the next frame-time for that virtual circuit. If sufficient capacity is available to transmit these cells over the next frame, the cells have a delay-jitter bound equal to $|T|$ and maximum delay bound equal to $2HT + \tau$ [10, p. 11], where H is the number of hops in a connection's path, and τ is one-way end-to-end propagation delay.

Stop-and-Go Queuing was originally intended to guarantee that a traffic stream that was (r, T)-smooth[1] at a network entry point remained (r, T)-smooth over intermediate hops. This was achieved by ensuring that traffic over successive frames (time intervals) did not get bunched together; frame n cells of a virtual circuit were never to contend for bandwidth with frame $n + k$, $k > 0$, of the same virtual circuit. In [10], Golestani gave an algorithm for cell dispatching for heterogeneous frame sizes when no cells were to be dropped. Frame times needed to be integer multiples of a base frame-time, and they needed to be synchronized across virtual circuits. In [12], he gave an algorithm for transmitting hierarchically encoded video with two levels of hierarchy. Lower priority cells could be dropped if high priority cells were in queue. Frame times were assumed equal and synchronized across virtual circuits.

[1] An (r, T)-smooth stream was defined as one where the average bit-rate over a time-interval T did not exceed r. Equivalently, the number of bits over $(nT, (n + 1)T]$ did not exceed rT, for all integer n.

We combine Stop-and-Go Framing with an active cell-discard unit and a new cell dispatcher. Mechanisms for providing heterogeneous delay-jitter bounds, and heterogeneous cell-loss probabilities are presented. Upper bounds on equivalent bandwidth needed are computed. An implementation is given where frames do not need to be synchronized, and frame-times are software set-table. Delay-jitter bounds can therefore be negotiated over a continuum, and be heterogeneous across virtual circuits. In order to achieve high utilizations for long-memory traffic, an active cell-discard mechanism is needed, and its implementation with associative matching of cell tags is given — to indicate that it is feasible with current hardware technology.

The advantages of using this method are:

- Cells of a given frame of a virtual circuit do not contend with cells of a previous frame (property of Stop-and-Go queuing). Old cells, if implemented, are always given lower priority.
- Equivalent bandwidth computations for meeting different cell-loss probabilities need not be concerned with correlation in input traffic. The cost of active-discard is small for long-memory traffic anyway (e.g., see Figure 1).
- Delay-jitter bounds are software set-table and are guaranteed to be met. Upper-bounds on cell-loss probability will also me met if a call is accepted (based on equivalent bandwidth computations – and assuming that traffic distributions are accurate).

Stop-and-Go Queueing's original requirement that a traffic stream declare its (r, T) parameters is dropped. This trades-off higher utilization for a loss-less network :– For a long-memory stream, the average rate over an (small) interval, T, can be significantly higher (or lower) than its overall average rate, so r would need to be close to peak rate for loss-less transmission, and would result in significantly low utilizations. In the current proposal, cell losses, while allowed, will be reduced through statistical multiplexing across virtual circuits and controlled through equivalent bandwidth computations.

Outline

The rest of the paper is organized as follows. Section 2 motivates the need for the proposed architecture with simulation results for a work-conserving server with long-memory input traffic and lazy (on-demand) cell-discard. Section 3 presents the architecture. It includes (i) framing with active cell-discard, and (ii) cell dispatching to meet heterogeneous delay-jitter and cell-loss guarantees for heterogeneous virtual circuits (with heterogeneous marginal distributions and dependence structures). Section 4 addresses upper-bounds on equivalent bandwidth needed to meet heterogeneous delay-jitter requirements and heterogeneous cell-loss probability bounds, and presents numerical examples. Section 5 presents our conclusions and directions for future work.

2 Queue Simulations With Controlled F-ARIMA$(1, d, 0)$ Processes

2.1 The Fractionally-Differenced ARIMA Process

A fractionally-differenced Auto-Regressive Integrated Moving Average process, denoted F-ARIMA(p, d, q), with $0 < d < 1/2$, is an example of a wide-sense stationary process with long memory, i.e., its mean, variance and co-variance functions are independent of time, and the sum of its auto-correlations diverge. In [9], Garrett and Willinger reported that VBR-video traffic has properties similar to a fractionally-differenced ARIMA $(0, d, 0)$ process with Gamma/Pareto marginals. Since its dependence structure is parameterized by d, $(d,\ 0 < d < 1/2)$, the F-ARIMA model provides a convenient

starting point for determining the impact of long-memory processes on queue length statistics. In later sections where the proposed architecture will be evaluated, we shall augment this model with three MPEG-I video traces (approximately one-half hour long each) made available in the public domain by Rose [29].

A brief overview of the F-ARIMA process follows. A background on long-memory processes appears in the Appendix.

ARIMA(p, d, q) models, with $d = 0, 1, 2, \ldots$ were popularized by Box and Jenkins [2] for purposes of modeling time series data. Models with a fractional d in the range $0 < d < 1/2$ was first introduced by Hosking [14] in an attempt to bridge the gap between fractional Gaussian Noise models and ARIMA models. Let $\{X_t\}$ denote observations from a stationary process with zero mean. An ARMA model of order (p, q) has the form

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + \epsilon_t - \theta_1 \epsilon_{t-1} - \theta_2 \epsilon_{t-2} - \cdots - \theta_q \epsilon_{t-q}, \tag{1}$$

where ϵ_t is white noise, and ϕ_j's and θ_j's are real. These models are also extended to model non-stationary behavior in data, and models that have proved useful in practice are Autoregressive Integrated Moving Average models or ARIMA(p, d, q) models for short.

Define a lag operator B as $X_{t-1} = BX_t$, and the difference operator ∇ as $(X_t - X_{t-1}) = \nabla X_t$. Notice that $\nabla X_t = (1 - B)X_t$. Let $\phi(B)$ and $\theta(B)$ be polynomials in the operator B, defined as follows:

$$\begin{aligned} \phi(B) &= (1 - \phi_1 B - \cdots - \phi_p B^p), \\ \theta(B) &= (1 - \theta_1 B - \cdots - \theta_q B^q). \end{aligned}$$

Then an ARIMA(p, d, q) process is defined as

$$\phi(B)\nabla^d X_t = \theta(B)\epsilon_t. \tag{2}$$

When d is equal to zero, (2) reduces to (1).

In [14], Hosking has shown that for $0 < d < 1/2$, an ARIMA$(0, d, 0)$ process is stationary, with a long memory, and an auto-correlation function

$$\rho_k = \frac{\Gamma(1-d)\Gamma(k+d)}{\Gamma(d)\Gamma(k+1-d)} \sim \frac{\Gamma(1-d)}{\Gamma(d)} k^{2d-1} \quad \text{as } k \to \infty. \tag{3}$$

The hyperbolic decay in (3) results in $\sum_k \rho_k \to \infty$ when $0 < d < 1/2$.

2.2 Performance of work-conserving servers with infinite buffers

An F-ARIMA$(1, d, 0)$ process was first used as the arrival process to a First-Come-First-Served queue with infinite buffers. The objective was to systematically vary the auto-correlation structure (superposition of an exponentially decaying component and a hyperbolically decaying component) and coefficient of variation of the input process and study the resulting Queue process. The process has the form

$$(1 - \phi_1 B)\nabla^d X_t = \epsilon_t; \;\; 0 \le \phi_1 < 1.0, \; 0 \le d < 0.5. \tag{4}$$

In (4), $\{X_t\}$, represents the number of cell arrivals in $(t, t+1]$. We used the algorithm due to Haslett and Raftery [13] for generating a sequence of $\{X_t's\}$ based on a fractionally-differenced ARIMA(p, d, q) model. The exponentially decaying component of the ACF is controlled by changing the auto-regressive coefficient, ϕ_1; the hyperbolically decaying component of ACF is controlled via

ϕ_1	d				
	0.001	0.1	0.2	0.3	0.4
0.0	1	3	4	18	158
0.2	2	3	9	21	244
0.4	3	5	19	78	285
0.6	6	12	24	168	563
0.8	16	33	105	615	2093

(a) Mean Queue Length

ϕ_1	d				
	0.001	0.1	0.2	0.3	0.4
0.0	5	20	77	1685	153189
0.2	11	37	375	2150	296727
0.4	32	84	2077	50711	466852
0.6	111	538	2437	167108	1417408
0.8	756	3685	55404	2043158	15102276

(b) Variance of Queue Length

ϕ_1	d				
	0.001	0.1	0.2	0.3	0.4
0.0	15	32	74	296	2310
0.2	21	46	191	354	2899
0.4	40	71	461	1469	4067
0.6	79	170	551	2773	8128
0.8	178	478	2070	9079	20847

(c) 0.999 Quantile of Queue Length

Table 1: Summary of queue statistics for mean utilization $u = 0.9$. (a) Mean queue length. (b) Variance of queue length. (c) 0.999 quantile of queue length.

---

d. (Note that a change in d also changes the ACF for small lags.) ϕ_1 was varied from 0.0 to 0.8 in steps of 0.2, and d was varied from 0.0 to 0.4 in steps of 0.1. The MPEG-I coded *Starwars* trace was reported to have an estimated d between 0.28 and 0.38 [9]. A white noise process is generated by setting $p = q = d = 0$.

Let C be the capacity of the server defined as the number of customers it can serve over a fixed time interval $(t, t+1]$, $\forall t$. Let u be the mean utilization of the server, i.e., $u = EX_t/C$. Queue simulations are parameterized by (ϕ_1, d, u). The queue lengths, $\{Q_t\}$, were recorded for each time interval using the standard relation $Q_{t+1} = \max(0,\ Q_t + X_t - C)$, from which different statistics of $\{Q_t\}$ were obtained. Tables 1(a)-(c) show a typical mean, variance, and 0.999-quantile of the queue length distribution for utilization, $u = 0.9$. We observe a significant impact of ϕ_1 and d, which qualitatively corroborate observations in [9, 19, 21].

2.2.1 Empirical model for queue statistics with infinite buffers

The effect of increasing ϕ_1 and d on mean, standard-deviation and several quantiles at the right tail of the queue length distribution show an exponential growth with both ϕ_1 and d for mean utilizations, $u \geq 0.7$. The relative impact of d is substantially larger. Figure 2(a) shows the 0.999-quantile of queue length distribution versus d (with ϕ_1 fixed) and Figure 2(b) shows the same versus ϕ_1 with d fixed. A logarithmic model may, therefore, be appropriate for the mean, standard-deviation and the tail-quantiles that were studied. Let Q refer to a given queue length statistic, and consider the following model for it:

$$\log Q = a_1 + a_2\phi_1 + a_3 d + \epsilon \tag{5}$$

Utilization (u)	Q-statistic	Coefficients			Diagnostics of Fit		
		a_1	a_2	a_3	$R^2(\%)$	F-statistic (on 2 and 22 deg. of freedom)	p-value
0.9	mean	-0.45^*	3.48	11.83	95.03	210.4	$4.552 * 10^{-15}$
	std-dev	0.12^*	3.48	12.60	95.89	256.8	$5.551 * 10^{-16}$
	0.999 quantile	2.18	3.48	12.17	97.53	431.1	0
0.8	mean	-2.23	3.35	8.72	90.49	104.7	$5.75 * 10^{-12}$
	std-dev	-1.11	3.38	9.48	90.93	110.3	$3.42 * 10^{-12}$
	0.999 quantile	1.11	3.50	10.04	91.99	126.4	$8.676 * 10^{-13}$
0.7	mean	-4.34	3.12	6.58	85.24	63.53	$7.238 * 10^{-10}$
	std-dev	-2.41	3.33	6.85	87.92	80.08	$7.974 * 10^{-11}$
	0.999 quantile	0.12^*	3.51	7.16	88.09	81.32	$6.871 * 10^{-11}$

$*$ These coefficients are not significantly different from 0 at 99% confidence level.

Table 2: Regression coefficients and diagnostics of fit. Let SST be the total variation in a given response data, SSE be the sum of squared errors after regression, and SSR = SST- SSE. R^2 is $SSR * 100/SST$, and shows the percentage of variation explained by regression. Let ν_R and ν_E be the degrees of freedom of SSR and SSE, respectively. The F-statistic is the ratio of Mean Square Regression (MSR = SSR/ν_R) to Mean Square Error (MSE = SSE/ν_E). In this specific case $\nu_R = 2$ and $\nu_E = 22$, respectively. The p-value represents the area under the tail of an F-density function from the F-statistic value to infinity; $(1 - p)$ is the probability that MSR is greater than MSE under the assumption that MSR and MSE are χ^2-distributed with ν_R and ν_E degrees of freedom, respectively. Since the residuals are normally distributed, this is to be expected.

In (5), (a_1, a_2, a_3) are constants and ϵ is white noise. Figures 3-4 show the diagnostics for a specific example (0.999-quantile of queue length with $u = 0.8$). In Figure 3, the left panel shows the response data, $\log Q$, versus corresponding fitted values $a_1 + a_2\phi_1 + a_3 d$. The right panel does not show any obvious trend in the absolute value of residuals with respect to the fitted values. In Figure 4(a), there does not appear to be a trend in the residuals with respect to the experiment numbers, and in Figure 4(b), the residual quantiles plotted against Standard Normal quantiles show an approximate straight line trend, implying that they are roughly Normally distributed. Table 2 shows the analysis of variation, and analysis of variance results for different queue statistics at different utilization levels, and the corresponding coefficients of fit. The regression model (5) explains $85 - 97\%$ of the variation in Q, and in all cases, the p-values are small. We believe the model is fairly accurate — however, it is upto the reader to make that determination for himself or herself.

From (5), the queue length statistic, Q, is given by

$$Q = e^{a_1}\ e^{a_2\phi_1}\ e^{a_3 d}\ e^{\epsilon}. \tag{6}$$

From Table 2, a_2 and a_3 are positive for all utilization levels, (i.e., the exponential trend is an increasing trend). Further, a_3 is greater than a_2 for all utilization levels, and their difference is larger for larger utilizations. Also, the range of variation of a_2 is small — across utilizations and across Q-statistic measures. The key conclusions from a performance perspective are that (i) the effect of d is substantially larger than the effect of ϕ_1 for all utilizations, and (ii) it increases with increasing utilizations.

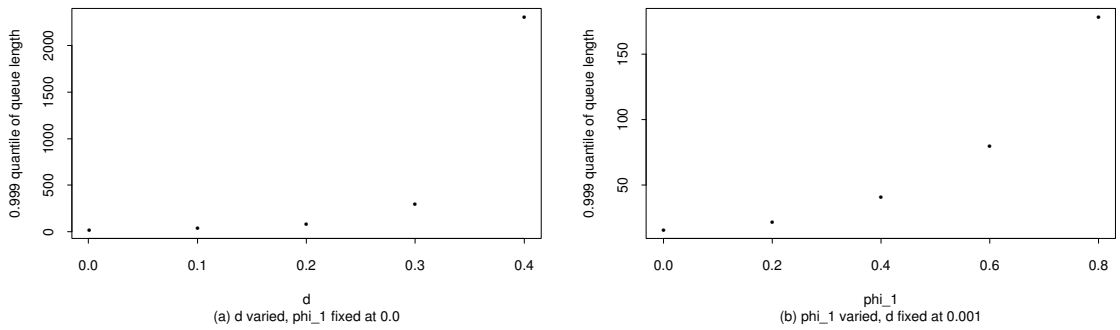


Figure 2: 0.999 quantile of queue length as a function of (a) d, with ϕ_1 fixed, and (b) ϕ_1, with d fixed. $u = 0.8$.

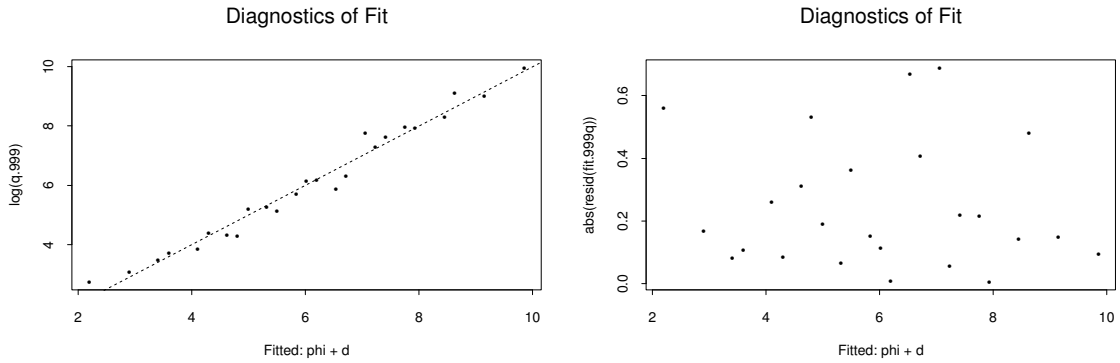


Figure 3: Quality of fit. (a) y-axis is log(0.999 quantile of queue length distribution); x-axis is the fitted value of the right hand side of (5). (b) y-axis is the absolute value of ϵ's, x-axis is the same as in (a). These plots show a fairly reasonable fit and no obvious trend in residuals.

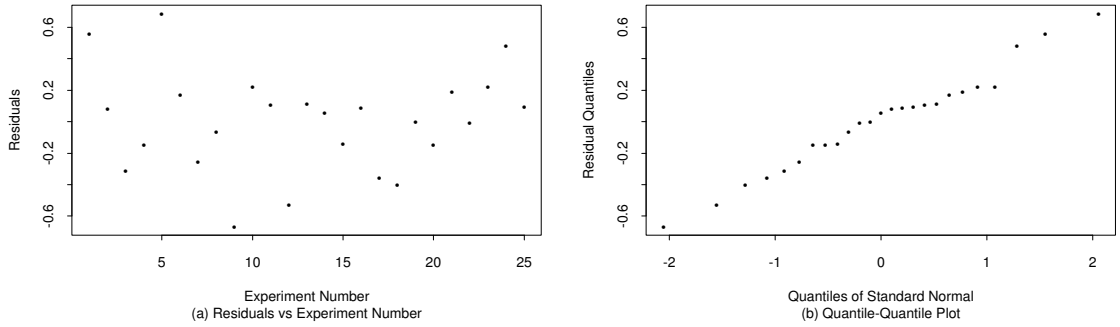


Figure 4: Additional diagnostics of residuals of regression.

2.2.2 Impact of coefficient of variation

The correlation structure and coefficient of variation was varied independently of one another as follows. Let

$$Y_N(t) = \frac{1}{N}\Big(X_t{}^{(1)} + \cdots + X_t{}^{(N)}\Big), \quad N = 1, 2, \cdots \tag{7}$$

where $\{X_t{}^{(i)}\}$, $i = 1, 2, \cdots$ are wide-sense stationary, mutually independent F-ARIMA$(1, d, 0)$ processes with identical values for (ϕ_1, d), and variance ${\sigma_X}^2$, all independent of i. Then $Y_N(t)$ is also an F-ARIMA$(1, d, 0)$ process with the same (ϕ_1, d), and variance ${\sigma_{Y_N}}^2 = {\sigma_X}^2/N$. To see the former, we have

$$(1 - \phi_1 B)\nabla^d X_t{}^{(i)} = \epsilon_t{}^{(i)}, \text{ for each } i.$$

Summing over all i,

$$\begin{aligned} \sum_i (1 - \phi_1 B)\nabla^d X_t{}^{(i)} &= \sum_i \epsilon_t{}^{(i)}\ , \\ \Rightarrow (1 - \phi_1 B)\nabla^d \left(\sum_i X_t{}^{(i)}\right) &= \sum_i \epsilon_t{}^{(i)}. \end{aligned} \tag{8}$$

The $\sum$ can be migrated inwards only if ϕ_1 and d are identical for all sequences. The resulting sequence will have the same (ϕ_1, d) in accordance with (8).

The coefficient of variation of $Y_N(t)$, denoted as η_{Y_N} is equal to $\eta_X/\sqrt{N}$. Therefore, the number of sequences, N, provides an independent degree of freedom with which to control η_{Y_N}, while keeping (ϕ_1, d) constant. An alternative method is to transform the Gaussian marginals to Gamma marginals after generating an F-ARIMA correlation sequence [9]. While, this method preserves d [16], the magnitude of auto-correlations were found to change slightly. We generated fifty independent F-ARIMA$(1, d, 0)$ sequences with $\phi_1 = 0.6$ and $d = 0.4$. Queue length simulations were performed, varying N. For a fixed desired mean cell-loss, as N was increased, the mean utilization achievable increased as well, corroborating the evidence in [9] — that for a smaller coefficient of variation, (or potentially, a tighter distribution tail), larger utilizations are achievable for long-memory traffic. For example, for a single frame buffer and a desired mean cell loss of less than 0.01, a single sequence ($N = 1$) achieved a utilization of 0.6 while for $N = 50$, a utilization of 0.9 was achieved.

2.3 Performance of work-conserving servers with finite buffers and lazy cell discard

An important question is does long memory in the arrival process matter when one has a finite number of buffers. We simulated a work-conserving, FCFS server with a finite number of buffers and lazy discard. (With lazy discard, cells are not dropped until buffers are full.) Queue lengths were simulated using:

$$Q_{t+1} = \min\Big(B, \max\left(0,\ Q_t + X_t - C\right)\Big).$$

with a finite number of frame buffers. Let C be the maximum number of cells that can be transmitted by the output channel in a time interval $(t, t+1]$. The buffer size, B, was chosen (somewhat arbitrarily) as integer multiples of C. In the sequel, we refer to C as a frame-buffer. Figures 1(a)-(b) show the mean number of cells dropped as a function of (i) dependence structure in the workload, (ii) number of frame-buffers allocated, and (iii) mean channel utilization, with standard deviation of number of arrivals set to unity. We observe that increasing buffer-size does not significantly

reduce mean cell loss for the long-memory sequence. As pointed out by Garrett and Willinger [9], a possible reason for this is that, for large d, the input sequence is persistent, so when it exceeds the service rate, it may do so for an extended period of time with a significant probability. Additional buffers may, however, reduce cell loss if coefficient of variation of the number of arrivals is small.

One solution is to assign a higher service rate; however this would result in a low utilization if the rate is assigned statically. For example, for a mean loss less than 0.005% and workload parameters ($\phi_1 = 0.4,\ d = 0.001$), a utilization of 0.9 and four frame-buffers was sufficient. However, for a workload ($\phi_1 = 0.4,\ d = 0.4$), and the same mean cell loss, an example allocation was $u = 0.7$ and two frame-buffers. In [28], Pancha and El Zarki report a similar finding for empirical MPEG-II traffic at a Leaky-Bucket input — where the token utilization is found to be approximately 0.5-0.6 for acceptably low cell-loss.

3 Proposed Architecture

3.1 Need for Stop-and-Go framing with active cell discard

The conclusions from the simulations in Section 2 are that work-conserving disciplines and on-demand, lazy cell-discard are likely to be inadequate for meeting quality of service guarantees for correlated traffic. Specifically,

(i) For long-memory traffic, an increase in buffer size will not decrease cell-loss probabilities appreciably, unless buffer allocation is large or the coefficient of variation is small.

(ii) Increased buffer-allocations will increase the maximum-delay bound and delay-jitter bound for applications.

Further, increased buffer allocations will increase complexity associated with computing and guaranteeing cell-loss bounds, not to mention maximum-delay and delay-jitter bounds.

Next consider the benefits of active cell-discard at each switch after an application specified frame time, T.

(i) In conjunction with Stop-and-Go Framing [11], where cells belonging to frame i are not eligible for service until frame-time $i+1$, all delivered cells are guaranteed a delay jitter bound equal to $|T|$, and a maximum delay bound through the network equal to $2HT + \tau$, where H is the number of hops in the application's path and τ is a constant representing the propagation delay;

(ii) If there are excess cells queued at a switch after a frame time, it is likely that (a) this is due to persistence in the arrival process, (b) these cells are likely to cause increased delay for cells in successive frames; and

(iii) Computation of equivalent bandwidth for multiplexed long-memory processes is non-trivial. However, it is possible to compute upper-bounds on equivalent bandwidth with active cell-discard (see Section 4).

Hardware support for efficient and flexible implementation of heterogeneous frames and active-cell discard is discussed next.

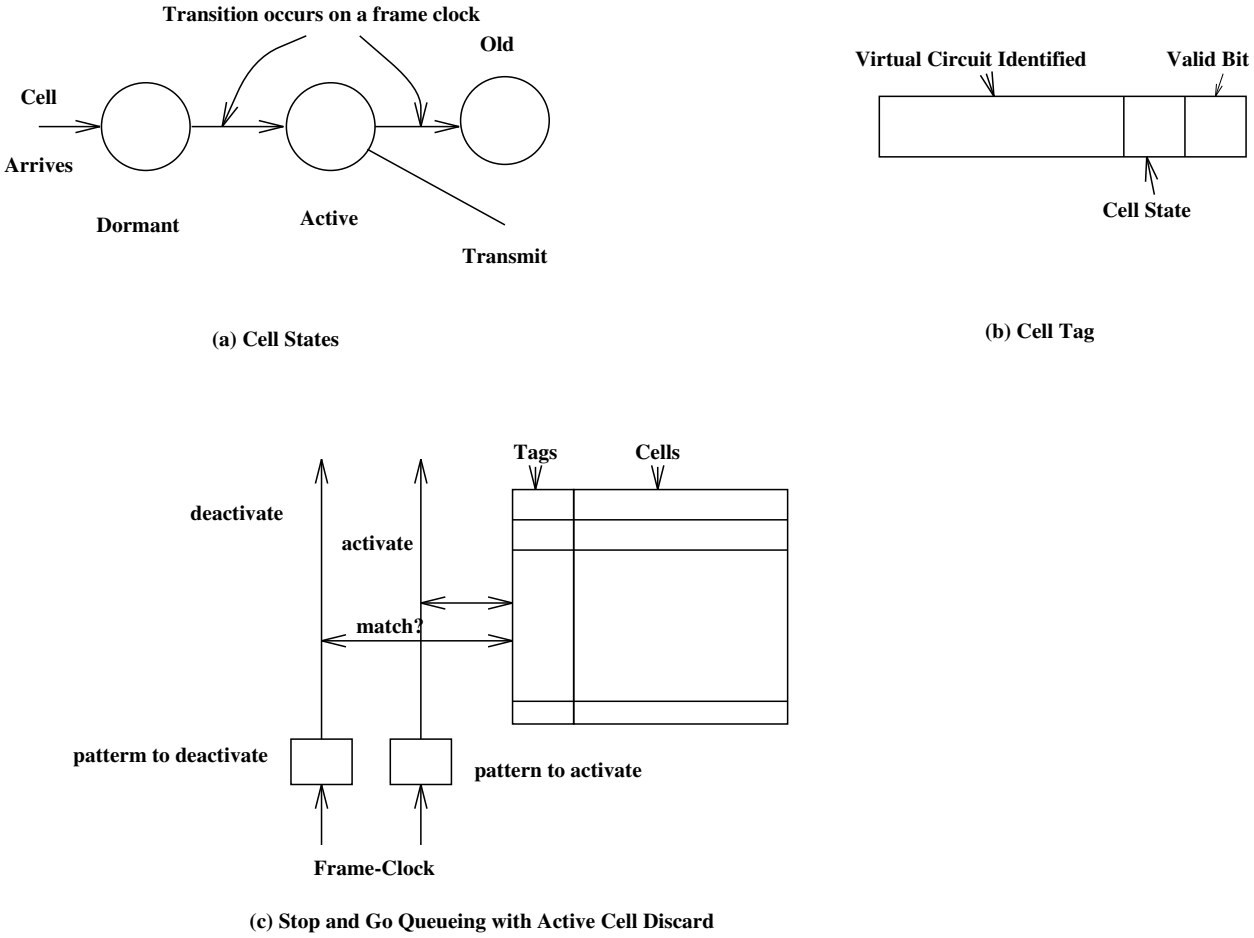


Figure 5: Implementation architecture for active cell-discard with Stop-and-Go queuing.

3.2 Implementation of framing and active cell discard

The objective is to induce a framing strategy on top of cells of a given virtual circuit, and for the switch to actively discard (or mark as *old*), cells that are not serviced during their assigned frame-time. In order to allow for flexibility of application-specified jitter-bounds, the frame-time should be software set-able (e.g., it may be negotiated during connection-open). It should then be set to the connection's delay-jitter tolerance. This also allows for flexibility of adjusting the frame-time during the lifetime of a virtual circuit, if necessary.

Issues that need to be addressed for frame synchronization, Stop-and-Go framing and active cell discard are as follows.

(a) *Frame synchronization between adjacent nodes (where nodes refer to switches and end-points):* Cells transmitted during the t^{th} frame by a node must be recognized as belonging to frame t by the next down-stream node. As shown below, an alternating bit sequence number distinguishing cells in adjacent frames is sufficient — if the sequence number is generated at the transmitter.

(b) *Frame-clock generation:* For the i^{th} virtual circuit, one needs a step-down counter, initialized under software control to the maximum number of cells that constitute its frame-time. Let

this number be M_i. The counter is to be fed with a clock that runs at the speed of cell transmission at the output link. On each clock cycle (at cell granularity), the counter must count down one tick until it hits zero. At this point, it will need to generate a *frame-clock* signal and reset itself to M_i.

(c) *Cell tagging:* A cell arriving during frame t for virtual circuit i, will not be eligible for service until frame $t+1$ for the same virtual circuit. It is, therefore, assigned a state, *dormant,* on arrival. See Figure 5(a). When the next *frame-clock* signal arrives, the cell is ready to be transmitted, so its state needs to be changed to *active.* If it still remains in the queue when the following *frame-clock* signal arrives, it is old, and now there are two possibilities. One strategy is to simply discard the cell and re-claim its buffer. A second strategy is to change its state to *old* and keep it eligible for transmission, but at the lowest priority level. Should a higher priority cell become *active* when an *old* cell is still enqueued and buffers are depleted, these buffers may be reclaimed and the *old* cell dropped at that time. In either case, the *active* and *old* cells are passed to the control of the dispatcher, discussed in Section 3.3.

To simplify the discussion for what happens next, let us assume that *active* cells not transmitted in their frame-time are discarded. Then, at any given time, cells belonging to frames t and $t+2$ will never be simultaneously present[2] at a switch output queue, and all that is necessary is to distinguish between cells in frames t and $t+1$. A single bit, therefore, suffices to distinguish between *active* and *dormant* cells.

Assume that during frame t, *dormant* cells are represented by a 0 and *active* cells have been marked 1 in the previous cycle. On a new cell arrival, the switch needs to attach to it a tag, identifying its virtual circuit and its frame number (in this case 0), set its valid bit to 1 and forward it to the output queue. See Figure 5(b). (As mentioned earlier, a convenient place to generate and attach the frame sequence number is at the transmitter; if this is done, synchronization steps across hops is simplified.) The valid bit's function is to help discard cells, similar to the action of flushing a cache memory on a context switch. In a fast cell-switched, virtual circuit network, a switch would implement a tagging scheme for virtual circuit identifiers anyway, so additional circuitry needed is small.

On the next *frame-clock,* the entire output queue would be fed with two logical signals, one to deactivate the *active* cells that did not get transmitted during their allotted frame time (due to lack of available capacity), and one to activate the *dormant* cells. See Figure 5(c). Both of these can be achieved by associatively matching the cell-tags with an identifier representing the appropriate virtual circuit and its state. The primary difference between standard content-addressable memories available commercially today and this is that more than one match is likely, especially for *dormant* cells). On a match, *active* cells mark themselves invalid by setting their valid bits to 0; the *dormant* cells move to the *active* state and are ready to be transmitted. At this point, they move under the control of the cell dispatcher, which must decide on a strategy that is consistent with the overall goals of delay-jitter and statistical cell-loss bounds.

A convenient model for the buffer memory organization is to view it as a set of logical queues, one per virtual circuit, with a sequence number distinguishing *active* and *dormant* cells. All *old* cells may potentially be grouped into one logical queue, as discussed below.

[2]One additional frame would be needed if perfect frame inter-node synchronization is achieved through extra delay circuitry proposed in [11]. The implementation would then require two bits to identify a frame instead of one. However, a simpler alternative is to transmit a one-bit frame sequence number with cells to downstream switches.

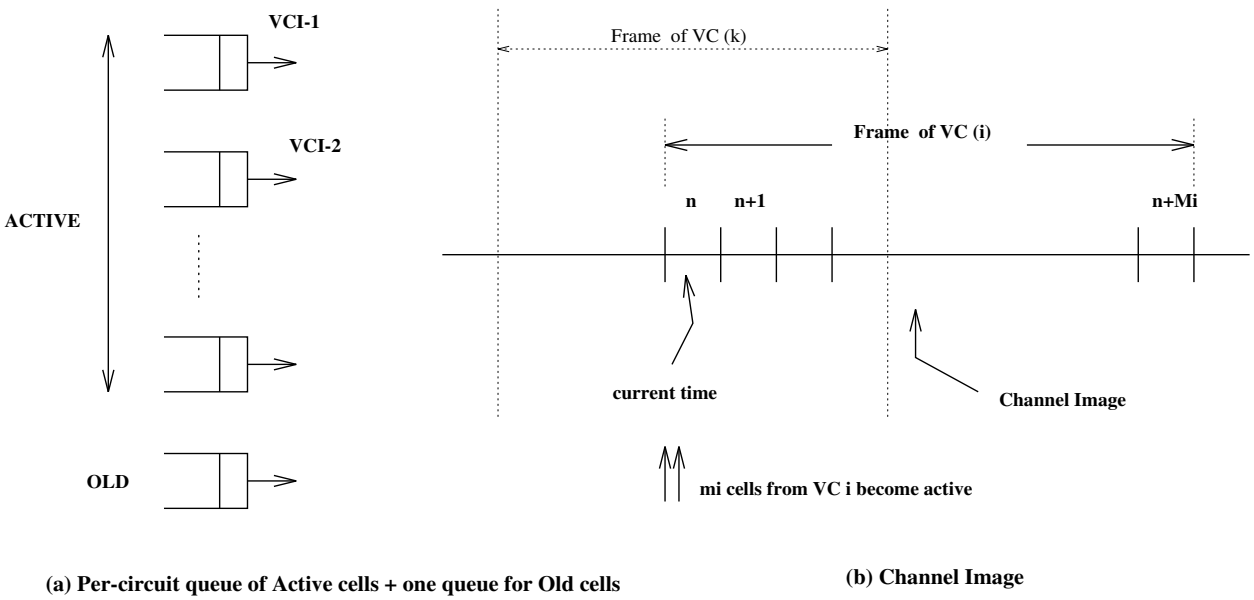


Figure 6: Cell-dispatcher's view. (a) Queue of *active* cells for each virtual circuit plus a queue of *old* cells. This is used by the transmitter unit. (b) `Channel_image`. This is shared by the scheduler and the transmitter units.

3.3 Cell Dispatcher

The cell dispatcher is responsible for (*i*) scheduling and (*ii*) transmitting *active* and (potentially) *old* cells. *Dormant* cells are not within its purview. From the dispatchers perspective, the *active* cells for each virtual circuit are assumed to be logically organized as a queue (see Figure 6(a)). The *old* cells (implemented optionally) are organized either as separate queues or as a single queue.

The dispatcher consists of two concurrent units, a scheduler and a transmitter. The scheduler allocates cell times to *active* cells of individual virtual circuits and decides which cells are to be dropped if contentions for capacity arise. The transmitter transmits them (and *old* cells if all *active* queues are empty and *old* cells are waiting). The scheduler will guarantee transmission of at least C_i cells for connection i, $(i = 1, \cdots, K)$, where K is the number of active virtual circuits at the multiplexor. The computation of C_i's is based on fairness and cell-loss requirement considerations, and is presented in Section 4. The scheduler and the transmitter share a circular buffer that represents channel allocations in the future. (This circular buffer is presented below as a linear array for convenience of exposition.) Let this data-structure be called `channel_image`. See Figure 6(b). `Channel_image[`n`]` records the ID of one virtual circuit. If `Channel_image[`n`]` equals i, the transmitter will transmit from the head of the *active* queue corresponding to virtual circuit i at time n. This will be modified below after the basic algorithm is presented.

The scheduler is activated on every new-frame activation (i.e., on a *frame-clock*). Let the new frame activation be at time n. (See Figure 6(b).) Let the corresponding virtual circuit be i, the frame-length (jitter-bound) be M_i, number of cells in the current *active* frame be m_i, and the minimum number guaranteed to be transmitted from this virtual circuit in any frame be C_i. `Channel_image` has future slots either marked with a virtual circuit identifier, or is marked empty. The scheduler's task is as follows.

- The time window in the future over which the m_i *active* cells need to be transmitted is $[n + 1, n + M_i]$. (The nth slot is kept aside for the transmitter to begin transmitting.)

- Beginning with $n + M_i$, going down to $n + 1$, it attempts to find the largest $k_i \leq m_i$ slots that are empty in `channel_image[`$n + M_i$`]` through `channel_image[`$n + 1$`]`, and mark each with the current virtual circuit, i.

- `If (`$k_i \leq C_i$`) {`
  `guaranteed_cells[i] =` k_i;
  `not_guaranteed[i] =` 0;
  `} else {`
  `guaranteed_cells[i] =` C_i;
  `not_guaranteed[i] =` $k_i - C_i$;
  `}`

- If $k_i = m_i$, then the scheduler's allocation task for this frame is done. Else, it needs to pick at most $m_i - k_i$ slots in `channel_image[`$n + 1$`]` through `channel_image[`$n + M_i$`]` that are marked, but not necessarily guaranteed, and over-write them with i. Each slot overwritten represents a cell loss for the corresponding virtual circuit. Since this would give priority to some virtual circuits over others when the number of *active* cells exceed the capacity available (for meeting deadlines), the policy used must balance fairness and cell-loss commitments. The dropping policy adopted is described below.

Dropping Policy

Let the negotiated mean cell-loss ratio of virtual circuit i be ϵ_i, and the estimated cell-loss ratio at time n be $\hat{\epsilon}_i[n]$. Let $S_i[n] = \epsilon_i / \hat{\epsilon}_i[n]$. Yang and Pan has proposed a dropping policy [31] where if an incoming cell arrives to a full buffer (in our case, full `channel_image`), the scheduler will search the buffer for the virtual circuit (j) that has the largest $S_j[n]$ and discard one of its cells. If the arriving cell belongs to virtual circuit j itself, then that cell will be dropped. The authors show that using the largest $S_j[n]$, is optimal in bandwidth utilization among all all stationary, space conserving loss scheduling schemes (see [31]).

We augment this strategy with (i) a minimum capacity C_i in order to provide fairness/firewall among connections, and (ii) relaxing the constraint that the candidate for dropping be necessarily the largest $S_j[n]$, see below. The objective is to select a j for which $S_j[n] > 1$ and `not_guaranteed[j] > 0`. The proposed dropping policy is as follows.

- If $k_i < C_i$, then the scheduler needs to overwrite at least $C_i - k_i$ cells belonging to other virtual circuits in `channel_image`. These cell-losses will be distributed among virtual circuits with `(not_guaranteed[j] > 0) and` $(S_j[n] > 1)$. If C_i cells have still not been marked, the scheduler will drop from `(not_guaranteed[j] > 0) and` $(S_j[n] \leq 1)$ and update $S_j[n]$ similar to [31]. In all cases, the number of erased cells is decremented from `not_guaranteed[j],` and `guaranteed_cells[i]` is incremented by the corresponding amount. The latter should be equal to C_i at the end of this step. (The C_i's are chosen such that $\sum_i C_i$ is always less than or equal to the capacity, so it is always true that `not_guaranteed[j] > 0` for some j, see Section 4. This does not, however, mean that there is no capacity sharing or multiplexing gain. See Section 4.)

- If $m_i > C_i$, then from the previous step, C_i cells have been scheduled. An additional ℓ_i cells ($0 \leq \ell_i \leq m_i - C_i$), may be scheduled, in which case `not_guaranteed[i]` will be set to ℓ_i. The scheduler schedules these cells only if there exists a j such that `(not_guaranteed[j] > 0) and` $(S_j[n] > 1 > S_i[n])$.

- $S_i[n]$ is updated with the number of dropped cells in this time frame, if any, i.e., with $(m_i - C_i - \ell_i)$.

The above cell dispatcher guarantees a minimum capacity C_i in each frame for virtual circuit i. Extra capacity, if available, will be used by any virtual circuit without interference from the dropping policy above — until the sum of arrivals fills up `channel_image` (i.e., exceeds the total output link capacity until the next due date). Only then, the dropping policy is used to give priorities among different virtual circuits. With upper-bound computations of equivalent bandwidth given in Section 4, and typical low cell-loss ratio requirements expected, the dropping policy is not expected to be called upon too frequently. Its effectiveness, therefore, needs to be evaluated, and simulation results for it are given in Section 4.1.3.

Transmitter

The transmitter works as follows. At time n,

1. If `channel_image[`n`]` is not empty, let $i =$ `channel_image[`n`]`. The transmitter transmits an *active* cell from the queue corresponding to virtual circuit i, and goes to Step 3.

2. If, `channel_image[`n`]` is empty, however, the transmitter proceeds to the next non-empty slot $n' > n$. If no such n' exists, it proceeds to Step 4. Else it transmits an *active* cell from the queue corresponding to the virtual circuit recorded in `channel_image[`n'`]`, and marks this slot empty. Let this virtual circuit identifier be i.

3. ```
   If (guaranteed_cells[i] > 0)
       guaranteed_cells[i] = guaranteed_cells[i] - 1;
   else
       not_guaranteed[i] = not_guaranteed[i] - 1;
   ```

4. If all future slots are empty, it may schedule an *old* cell, if this option is implemented.

Rationale

The reason for allocating slots beginning with time $n + M_i$ down to $n + 1$ is to keep slots near time n available for virtual circuits with potentially lower delay-jitter bounds that may be activated at some time in the future. Since the constraint/objective is to ensure that *active* cells of VC i are transmitted at or before time $n + M_i$, this strategy would meet more deadlines than if cells were allocated from time $n + 1$ upwards. If the transmitter finds a slot empty, however, it is important to transmit *active* cells if they are waiting in queue, so potential contentions for future slots are reduced. The scheduler and the transmitter, therefore, implements a pseudo earliest-due--date schedule, with a minor difference: if the scheduler finds a slot n' marked with VC j when it is allocating slots for VC i, it may allocate a cell $n'' < n'$ for VC i, even though $i's$ deadline is potentially greater than that of j. This does not violate the deadlines for either of the two VCs, but does increase the likelihood of dropping a cell from VC j. The alternative is to shift j's slot to the left and allocate `channel_image[`n'`]` to i at a higher implementation cost.

The reason for keeping slot n out of reach of the scheduler's view is to enable the transmitter to schedule timely transmission during the current slot. Also, the transmitter needs to have priority over the scheduler in its access/writing of the `channel_image` data structure.
   ```

4 Equivalent Bandwidth

In this section, we compute upper-bounds on the capacity needed for guaranteeing desired overflow probabilities and mean cell loss in the presence of Stop-and-Go framing with active cell-discard. We also compute the values $\{C_i\}$ to be used by the cell dispatcher in Section 3.3.

Lemma 1 *Under Stop-and-Go framing with active cell-discard, the capacity needed to meet a desired overflow probability, ϵ_o is greater than or equal to the capacity needed to meet a mean cell-loss $\epsilon_m \geq \epsilon_o$.*

Proof: Without loss of generality, consider a single virtual circuit operating under Stop-and-Go framing with active cell-discard. Let X be the random variable indicating number of cell arrivals in a frame. Let C be the capacity needed to meet a desired mean cell-loss ϵ_m. Let $f(x)$ be the probability density function of X. Then,

$$\begin{aligned}
\epsilon_m &= \int_C^\infty \frac{x-C}{x} f(x)dx && (9)\\
&= \int_C^\infty f(x)dx - C\int_C^\infty \frac{1}{x} f(x)dx && (10)\\
&\leq \int_C^\infty f(x)dx && (11)\\
&= \epsilon_o && (12)
\end{aligned}$$

(9) is true because all cells not served within its corresponding frame time are discarded. (11) follows from (10) because for $C \geq 0$, the second term in (10) is always non-negative. ∎

The point of this (trivial) lemma is that we will be using overflow probability estimates in our equivalent bandwidth computations, and these in turn will provide an upper-bound on the capacity needed to meet an equal or greater mean cell-loss.

4.1 Equivalent bandwidth for homogeneous delay-jitter bounds

In the presence of active cell-discard, homogeneous delay-jitter bounds implies that all virtual circuits have the same frame-size. We consider the following two cases separately: (i) homogeneous cell-loss guarantees and, (ii) heterogeneous cell-loss guarantees. For the latter case, all virtual circuits with the same cell loss requirement are grouped into the same class.

4.1.1 Scenario 1: equal cell-loss probability requirements

This is the trivial case, and is included here for completeness. Let the number of connections be N, and let the desired upper-bound on the required cell-loss probability per connection be ϵ. Let us assume for the moment that all frames are synchronized which is a worst-case scenario for cell-losses. If frames are not synchronized the number of cells lost may be smaller. Let $X_1, \cdots, X_N$ be the number of cell arrivals from connections 1 through N, in a given frame. Let

$$S_N = X_1 + \cdots + X_N.$$

Let $f_k(x)$ be the probability density function of X_k, $(k = 1, \cdots, N)$. Let $f_{S_N}(x)$ be the probability density function of S_N. Assuming that traffic across connections are mutually independent, $f_{S_N}(x)$ is the convolution of the of the $f_k(x)$'s:

$$f_{S_N}(x) = f_1(x) \otimes f_2(x) \otimes \cdots \otimes f_N(x). \quad (13)$$

The minimum bandwidth required is the $(1-\epsilon)$-th quantile of the distribution of S_N. For large N, using the central limit theorem, $f_{S_N}(x)$ is approximated by

$$f_{S_N}(x) \approx \frac{1}{\sqrt{2\pi}\sigma}\, e^{\frac{-(x-\mu)^2}{2\sigma^2}},$$

where $\mu = \sum_{k=1}^{N} \mu_k$ and $\sigma^2 = \sum_{k=1}^{N} \sigma_k^2$. μ_k and σ_k are the mean and standard deviation of X_k, respectively. In this case (large N), since C is the $(1-\epsilon)$-th quantile of a Normal Distribution with mean μ and standard deviation σ, we may equivalently set $Z = (S_N - \mu)/\sigma$, and if $z_{1-\epsilon}$ is the $(1-\epsilon)$-th quantile of a Standard Normal Distribution, C is given by

$$C = \mu + z_{1-\epsilon}\, \sigma. \tag{14}$$

For small N, (13) needs to be used instead.

Computing C_i's

In order to guarantee fairness, the cell-dispatcher needs a capacity allocation C_i for each virtual circuit i. Recall from Section 3.3 that if less than C_i is needed by VC i in a given frame, the rest is made available to the other VCs. The objective of setting a C_i is to provide firewall/fairness to each connection. The problem, therefore, is to determine $C_i \geq 0$, $(i = 1, \cdots, N)$, such that $\sum_{i=1}^{N} C_i = C$.

Heuristics alternatives when all connections require the same maximum overflow probability ϵ, are:

(i) $C_i = \frac{\mu_i}{\sum_{i=1}^{N} \mu_i} C$, where μ_i is the mean number of cell arrivals per frame for VC i.

(ii) $C_i = \frac{q_i(\epsilon)}{\sum_{i=1}^{N} q_i(\epsilon)} C$, where $q_i(\epsilon)$ is the $(1-\epsilon)$-th quantile of the number of cell arrivals in a frame for VC i.

We studied (ii) to address heterogeneous arrival distributions. Experiments were conducted with Gamma distribution for number of arrivals/frame, equal means and a range of different tails. When the tails were substantially different, Method (ii) was more robust in distributing cell-losses than Method (i).

4.1.2 Scenario 2: heterogeneous cell-loss probability requirements

Consider two different classes of traffic with desired upper-bounds on cell-loss probabilities, ϵ_1, and ϵ_2, respectively. Multiple virtual circuits with same delay-jitter bound (frame-size) and cell-loss probability requirement are considered part of the same class. Let $f_{X_i}(x_i)$, $(i = 1, 2)$, be the density function for the number of bits/frame transmitted by class i. The following two methods are upper-bound estimates, with progressively tighter bounds – and increased computational complexity. The algorithm below generalizes to more than two classes.

Alternative 1.

The simplest upper-bound on equivalent bandwidth is the sum of equivalent bandwidths for each ϵ_i, $(i = 1, 2)$ computed in isolation. Hence, the minimum bandwidth required for connection i is $(1-\epsilon)$-th quantile of the distribution of $f_{X_i}(x_i)$, i.e, the smallest C_i that satisfies

$$\int_{C_i}^{\infty} f_{X_i}(x_i) dx_i \leq \epsilon, \tag{15}$$

and the equivalent bandwidth is $C = \sum_i C_i$. This approach does not consider the statistical multiplexing gains across different loss classes and over-estimates the true equivalent bandwidth needed. The following method gives a tighter upper-bound.

Alternative 2.

Let $f_{X_1,X_2}(x_1, x_2)$ be the joint-density of $f_{X_1}(x_1)$ and $f_{X_2}(x_2)$. We assume that traffic classes are spatially independent, i.e., $f_{X_1,X_2}(x_1, x_2) = f_{X_1}(x_1)\ f_{X_2}(x_2)$. The algorithm for computing a tighter equivalent bandwidth estimate is as follows.

Algorithm implementing Alternative 2

1. Initialize C_i in accordance with (15). Let Δ be a (small) quantum of capacity that may be subtracted from C_i, $(i = 1, 2)$. Let ${C_i}^0 = C_i$, $(i = 1, 2)$, and let the iteration step $n = 1$.

2. Set ${C_i}^n = {C_i}^{n-1} - \Delta$, $(i = 1, 2)$. Set boolean variable `tryagain` to TRUE.

3. The probability that no cells will be dropped from class 1 is the probability that the arrivals from Class 1, X_1, is less than C_1 OR X_1 is greater than C_1 but the total arrivals in the frame, $X_1 + X_2 \leq C_1 + C_2$. Therefore, we need to check if

$$P(X_1 \leq {C_1}^n) + P({C_1}^n < X_1 \leq {C_1}^n + {C_2}^n - X_2) \quad \geq \quad 1 - \epsilon_1 \tag{16}$$

$$\Rightarrow\ F_{X_1}({C_1}^n) + \int_0^{{C_2}^n} \int_{{C_1}^n}^{{C_1}^n + {C_2}^n - x_2} f_{X_1}(x_1)\ f_{X_2}(x_2)\ dx_1\ dx_2 \quad \geq \quad 1 - \epsilon_1 \tag{17}$$

$$\Rightarrow \int_0^{{C_2}^n} [F_{X_1}({C_1}^n + {C_2}^n - x_2) - F_{X_1}({C_1}^n)]\ f_{X_2}(x_2)\ dx_2 \quad \geq \quad 1 - \epsilon_1 - F_{X_1}({C_1}^n)$$

$$\Rightarrow \int_0^{{C_2}^n} F_{X_1}({C_1}^n + {C_2}^n - x_2)\ f_{X_2}(x_2)\ dx_2 \geq 1 - \epsilon_1 - F_{X_1}({C_1}^n)[1 - F_{X_2}({C_2}^n)] \tag{18}$$

   If inequality (18) is not satisfied, mark `tryagain` FALSE.

4. Analogous to (18), check if the following is satisfied:

$$\int_0^{{C_1}^n} F_{X_2}({C_1}^n + {C_2}^n - x_1)\ f_{X_1}(x_1)\ dx_1 \quad \geq \quad 1 - \epsilon_2 - F_{X_2}({C_2}^n)[1 - F_{X_1}({C_1}^n)] \tag{19}$$

   If (19) is not satisfied, mark `tryagain` to FALSE.

5. If `tryagain` is TRUE goto Step 2. Else goto Step 6.

6. A tight upper-bound on the equivalent bandwidth is ${C_1}^{n-1} + {C_2}^{n-1}$.

If their are K classes, the bandwidth needed for class k at iteration n is computed as follows [compare this to (16)].

$$P(X_k \leq {C_k}^n) + P({C_k}^n < X_k \leq \sum_{i \neq k} C_i - \sum_{i \neq k} X_i)\ \geq 1 - \epsilon_k \tag{20}$$

Cell loss ratio (ϵ_1, ϵ_2, ϵ_3)	Alternative 1 Capacity (Kbits/frame)	Alternative 2 Capacity (Kbits/frame)
$(1.5 * 10^{-2}, 2 * 10^{-2}, 10^{-2})$	256	150
$(2 * 10^{-3}, 3 * 10^{-}3, 1 * 10^{-3})$	388	220
$(10^{-2}, 10^{-3}, 10^{-4})$	460	280

Table 3: Result for equivalent bandwidth calculation. Where, ϵ_1, ϵ_2, and ϵ_3 represent the cell-loss ratio for Terminator, Goldfinger, and Soccer respectively

The second term in (20) considers only the case when the total arrivals of all connections in a frame is less than the total capacity. It is possible to not drop any cell from connection k, even if the total number of arrivals is greater than the total capacity. Therefore, the equivalent bandwidth computation above is an upper bound. As an example, for three ϵ-classes ($K = 3$), the integral (20) for $k = 1$ evaluates to the following.

$$F_{X_1}({C_1}^n) + \int_0^{{C_2}^n + {C_3}^n} \int_0^{{C_1}^n + {C_2}^n + {C_3}^n - x_3} \int_{{C_1}^n}^{{C_1}^n + {C_2}^n + {C_3}^n - x_2 - x_3} f_{X_1}(x_1)dx_1 \, f_{X_2}(x_2)dx_2 \, f_{X_3}(x_3)dx_3 \geq 1 - \epsilon_1 \tag{21}$$

For small number of jitter-classes, K, (e.g., for $K \leq 3$), these integrals can be computed quickly. For larger K, approximations need to be found. For real-time computations, they would need to run at call-arrival time-scales. With Alternative 2, this would be feasible only if the number of jitter classes is kept small. The other possibility is to explore implementing them off-line and using table look-up for call-admission.

4.1.3 Numerical results and simulation

Equivalent Bandwidth calculation

We applied Alternatives 1 and 2 to MPEG-I video traces made available in the public domain by Rose [29]. Table 3 shows the equivalent bandwidths needed for three of the traces (Terminator, Goldfinger, and Soccer) when all of them belong to the same jitter-class (same frame-size), and need different cell-loss probabilities. The marginal distribution of each trace appeared to be a Gamma distribution, so a fitted Gamma was used in the equivalent bandwidth computations. Alternative 2 resulted in approximately 40% savings for these experiments. The fact that it results in bandwidth savings over Alternative 1 is of course obvious. The key is that Alternative 2 can solve a heterogeneous $(\epsilon_1, \epsilon_2, \epsilon_3)$ specification.

Scheduler with dropping policy

In the following experiments, each class had only one connection. Using the capacity calculated using Alternative 2 we next evaluate the effectiveness of guaranteeing a minimum capacity C_i per frame for class i at the cell dispatcher. Recall from Section 3.3, that when `channel_image` was full, the cell dispatcher gave preference to classes j, that had not yet received at least C_j cell allocations in the current frame and to those with smaller current estimates of the ratio $S_j[\cdot]$.

In experiments where I-frames of different connections did not overlap in time, the cell-loss ratios of all connections were less than that requested, so clearly an approach that ensures such a load scheduling across connections is desirable — if it is feasible to implement.

Expt #	Video trace	CLR negotiated	CLR delivered (LIFO)	CLR delivered (proposed dispatcher)
I.	Terminator	0.01	0.0030	0.0012
	Goldfinger	0.001	0.0045	0.0046
	Soccer	0.0001	0.00037	0.00002
II.	Terminator	0.002	0.0014	0.0019
	Goldfinger	0.003	0.0020	0.0029
	Soccer	0.001	0.0018	0.0007

Table 4: Comparison of the proposed dispatcher and a simple Last In First Out cell dispatcher. The numbers indicate that additional controls over a simple LIFO dispatcher is necessary.

In experiments where I-frames did overlap, we compared the dropping policy of two schedulers: (i) the dispatcher mentioned in Section 3.3, and (ii) a simple Last in First Out dropping policy (LIFO), i.e., newly *active* cells that found the channel image full were dropped. Table 4 shows the results. As is expected, the experiment shows that LIFO may not be able to guarantee the desired cell loss ratio although in both cases the overall cell losses were the same.

4.2 Equivalent bandwidth for heterogeneous frames

Let there be L delay-jitter classes, each with a different delay-jitter bound, and a corresponding frame-size. Let n_ℓ, $(\ell = 1, \cdots, L)$, be the number of connections belonging to Class ℓ. For Class ℓ in isolation, the aggregate capacity needed to meet prescribed cell-loss probabilities may be computed in accordance with Section 4.1. Let C_ℓ be the capacity needed for Class ℓ traffic. Then an upper-bound on capacity needed for all classes is $\sum_\ell C_\ell$. Tighter bounds are non-trivial, and are currently under study.

5 Summary and Conclusions

A positively correlated arrival process has been shown to significantly increase queue-length statistics at a multiplexor. The relationship between this increase and a parameterized correlation structure was studied for a work-conserving server with lazy (on-demand) cell-discard, through a fractionally-differenced ARIMA$(1, d, 0)$ process in Section 2. The fractionally-differenced ARIMA process enabled the study of a queue with a controlled correlation structure and controlled coefficient of variation. Simulation results showed that:

- For infinite buffers, selected queue length statistics (mean, standard-deviation and quantiles at the right tail of queue length distribution) significantly increased with both d and ϕ_1. They were proportional to $e^{a_2\phi_1}$ and $e^{a_3 d}$, $0 \leq \phi_1 < 1.0$, $0 < d < 1/2$. Further, a_3 was significantly larger than a_2.

- The coefficient of variation of number of arrivals in a time-interval played a role in determining utilizations achievable for a given value of expected cell-loss, corroborating results of Garrett and Willinger [9].

- Increased buffer size did not significantly decrease cell-loss probabilities, especially for a long-memory process, and less so if its coefficient of variation was large. In comparison, the cell loss probability for a white noise process or one with short memory decreased rapidly with increased buffer size. Therefore, for long-memory traffic, if coefficient of variation was

large, a higher service rate was needed. This is in agreement with [28] where higher rates were necessary at a Leaky-Bucket for individual MPEG-II video streams. The ACF for these streams were significant into far lags, and the coefficient of variation of each stream (individually) was large.

Increasing buffer size for long memory traffic makes guaranteeing maximum-delay and delay-jitter bounds non-trivial. The active cell-discard mechanism, in conjunction with individual virtual circuit frame-clocking, and the priority cell-dispatcher in Section 3, has the following properties.

- It supports heterogeneous delay-jitter and simplifies computing upper-bounds on equivalent bandwidth needed for guaranteeing cell-loss probabilities;
- It is able to exploit statistical multiplexing across different connections, and yet guard against large queue lengths caused by correlated traffic. This is because (i) the tail of the aggregate distribution of a sum of sources is smaller than the tail of an individual source (standard advantage of packet switching), and (ii) the framing structure protects packets in future frames from being delayed by packets in previous ones even if the arrival process is correlated;
- The framing structure guarantees maximum delay and delay-jitter bounds for any network topology [11]. Unless loss-free service is desired, Stop-and-Go framing does not require a tight, deterministic, rate specification from each source;
- Its implementation with associative tags adds only a moderate cost; and
- Frames from different virtual circuits need not be synchronized to meet desired quality of service objectives.

The cell-dispatcher implements:

(a) a pseudo earliest-due-date priority discipline for scheduling *active* cells on the future `channel_image`, and

(b) a priority cell-drop mechanism that determines which *active* cells to drop in case of a full `channel_image`. It does so based on (i) the ratio of desired cell-loss to estimated cell-loss ($S_j[\cdot] = \epsilon_j/\hat{\epsilon}_j$), and (ii) the minimum capacities C_i to be provided to each virtual circuit per frame.

Cell-drop scheduling based on maximum $S_j[\cdot]$ has been shown to be optimal in bandwidth utilization among all stationary, space-conserving loss-scheduling mechanisms by Yang and Pan [31]. In conjunction with the pseudo earliest-due-date component, it attempts to also provide a minimum number of lost deadlines. The minimal capacity component, C_i, has two functions:

- It provides a firewall for virtual circuits from connections j, that have a low $S_j[\cdot]$ — potentially resulting from long periods of small transmissions — followed by a large surge of arrivals. Note that this is possible for long-memory traffic.
- It enables computation of tighter equivalent bandwidths — in Alternative 2 in Section 4.1.2, Eq. (20), the tighter upper-bound computation assumes that an amount C_i will be available for virtual circuit i if it needs it.

We are currently investigating tighter upper-bounds for equivalent bandwidth with multiple jitter classes.

Acknowledgment

We thank Biswanath Mukherjee for his feedback on earlier drafts of this paper.

A Background On Short and Long Memory Processes

An overview of long-memory and short-memory processes is presented in this Appendix. The interested reader is referred to Cox [4] for an excellent survey of this topic. Readers familiar with this topic may wish to skip this Section.

Let $\{X_t\}$, $t = 0, 1, 2, \cdots$ be a wide-sense stationary stochastic process, i.e., a process with a stationary mean $\mu = E[X_t]$, a stationary and finite variance $v = E[(X_t - \mu)^2]$, and a stationary auto-covariance function $\gamma_k = E[(X_t - \mu)(X_{t+k} - \mu)]$, $(k = 0, 1, 2, \cdots)$ that depends only on k, and not on t. Observe that $v = \gamma_0$. Let the auto-correlation of $\{X_t\}$ at lag k be denoted as ρ_k, where by definition, $\rho_k = \gamma_k/\gamma_0$.

For each m, let $\{X_j{}^{(m)}\}$ denote the sample mean of $X_{jm-m+1} \cdots X_{jm}$, i.e.,

$$X_j{}^{(m)} = \frac{1}{m}\left(X_{jm-m+1} + \cdots + X_{jm}\right). \tag{22}$$

Let v_m, $\{\gamma_k{}^{(m)}\}$ and $\{\rho_k{}^{(m)}\}$ denote the variance, the auto-covariance and the auto-correlation functions of $\{X_j{}^{(m)}\}$. Then $v_1 = v$ and v_m is given by [4]

$$\begin{aligned} v_m &= E\left[\frac{1}{m}\left(X_{jm-m+1} + \cdots + X_{jm}\right)\right]^2 - \left[E\,\frac{1}{m}\left(X_{jm-m+1} + \cdots + X_{jm}\right)\right]^2, && (23)\\ &= \frac{v}{m} + \frac{2}{m^2}\sum_{k=1}^{m}(m-k)\gamma_k, && (24)\\ &= \frac{v}{m} + \frac{2}{m^2}\sum_{s=1}^{m-1}\sum_{k=1}^{s}\gamma_k. && (25) \end{aligned}$$

(24) follows from (23) by expanding the first term, applying the definition of γ_k, and simplifying. It follows from (25) that

$$\gamma_k = \frac{1}{2}\nabla^2(k^2 v_k), \tag{26}$$

where ∇^2 denotes the central second-difference operator. The reader may verify this by expanding the right hand side of (26) and substituting for v_k with the right hand side of (25).

The auto-covariance function for $\{X_j{}^{(m)}\}$ is given by

$$\begin{aligned} \gamma_k{}^{(m)} &= \frac{1}{2}\nabla^2(k^2 v_k{}^{(m)}), && (27)\\ &= \frac{1}{2}\nabla^2(k^2 v_{km}). && (28) \end{aligned}$$

The process $\{X_t\}$ is said to have short-range dependence if $\sum_k \gamma_k < \infty$. Equivalently, from (25), v_m, for large m, is asymptotically of the form v'/m, with v' finite.

The process $\{X_t\}$ is said to have long-range dependence if $\sum_k \gamma_k \to \infty$. Equivalently, from (24), $mv_m \to \infty$ as $m \to \infty$.

An example of a correlation structure that leads to long-range dependence is when, for $0 < \beta < 1$,

$$\begin{aligned} \gamma_k &\sim \gamma' k^{-\beta}, \text{ for large } k, \text{ or equivalently,} && (29)\\ v_m &\sim v' m^{-\beta}, \text{ for large } m. && (30) \end{aligned}$$

Simple relations based on (25) connect γ' and v'. It follows from (28) that

$$\rho_k^{(m)} \rightarrow \frac{1}{2}\nabla^2(k^{2-\beta}). \tag{31}$$

$$\doteq \frac{1}{2}(2-\beta)(1-\beta)k^{-\beta}. \tag{32}$$

(32) following for large k from the asymptotic equivalence of differencing and differentiation.

A hyperbolically decaying auto-correlation function shown in (29) is one example of a stationary long-range dependent process. Other forms are possible, at least in theory, as long as $\sum_k \gamma_k \rightarrow \infty$.

In comparison, a process with an exponentially decaying auto-covariance function, $\gamma_k \sim \gamma' e^{-\beta}$ for large k leads to a short-range dependent process because $\sum \gamma_k$ converges. An example process that exhibits this property is the auto-regressive moving average (ARMA) process where γ_k is a sum of exponentials.

Studies by Beran et al, Garrett and Willinger and Pancha and El Zarki [3, 9, 27] indicate that long-range dependence is to be expected in variable-bit-rate video traffic. The Hurst coefficient, H, defined as $(1 - \beta/2)$, was estimated in [9] as between 0.75 and 0.88. An implication is that, even if traffic were aggregated over long time periods, the net aggregate will not appear as white noise. Instead, the aggregate will have bursty sub-periods and less bursty sub-periods for small as well as large time-scales.

Delivering delay, loss and delay-jitter bounds for such traffic types on Broadband-ISDN networks is the focus of this study.

A.1 Self-Similarity

The process $\{X_t\}$ is said to be exactly (second-order) self-similar if ${\rho_k}^{(m)} = \rho_k$ for all m and k, i.e., the correlation structure is preserved across different time scales. Such a process is characterized by the Hurst parameter, H, where $H = 1 - \beta/2$. Notice that $1/2 < H < 1$, when $0 < \beta < 1$. A key property of long-range dependence is that the variance of $\{{X_j}^{(m)}\}$ decays in proportion to m^H, instead of $m^{1/2}$ even for large m. The Central Limit Theorem for independent random variables leads to $H = 1/2$. Mandelbrot and Van Ness [23] has shown that $H = 1/2$, even for short memory processes.

It was observed by Hurst [17] and by Mandelbrot and Wallis [24, 25] that many naturally occurring time series show $H > 1/2$. Examples include annual water flows of the Nile and many other rivers, sun-spot numbers, etc. (One river with H *equal* to $1/2$ is the Rhine.) It is also known from Leland et al's study [22] that Ethernet traffic traces show second-order self-similarity with $H > 1/2$, in contrast to Poisson and other Markovian models where the auto-correlations vanish at larger time scales.

References

[1] Adas, A. and A. Mukherjee, "On Resource Management and QoS Guarantees For Long Range Dependent Traffic," *Proc. IEEE INFOCOM,* pp 779-787, Boston, April 1995. Extended version: http://www.cc.gatech.edu/tech_reports/index.94.html/GIT-CC-94-60.

[2] Box, G.E.P, G.M. Jenkins, "Time series analysis — forecasting and control," Prentice Hall, 1976.

[3] Beran, J., R. Sherman, M.S. Taqqu and W. Willinger, "Variable-bit-rate video traffic and long-range dependence," accepted for publication in *IEEE Trans. Networking,* 1993.

[4] Cox, D. R., "Long-range dependence: A review," *Statistics: An Appraisal,* Ames, Iowa, 1984, pp. 55-74, Iowa State Univ. Press. 1984.

[5] Huang, C., M. Devetsikiotis, I. Lambadaris, and A. Kayes, "Self-similar modeling of variable bit-rate compressed video: a unified approach," *Proc. of ACM Sigcomm,* London, 1995.

[6] A. Erramilli, O. Narayan and W. Willinger, "Experimental queuing analysis with long-range dependent traffic," Preprint, September 28, 1994.

[7] Ferrari, D., and D. Verma, "A scheme for real-time channel establishment in wide-area networks," *IEEE Journal on Selected Areas in Communications,* pp 368-379, April 1990.

[8] Garrett, M., "Contributions toward real-time services on packet-switched networks," Ph.D. Thesis, Columbia University, 1993.

[9] Garrett, M., and W. Willinger, "Analysis, modeling and generation of self-similar VBR video traffic," *Proc. ACM Sigcomm,* pp 269-280, London, 1994.

[10] Golestani, S. J., "A Stop-and-Go queueing framework for congestion management," *Proc of ACM Sigcomm,* Philadelphia, pp 8-18, September 1990.

[11] Golestani, S. J., "A framing strategy for congestion management," *IEEE Journal on Selected Areas in Communications,* pp 1064-1077, September 1991.

[12] Golestani, S. J., "Duration-limited statistical multiplexing of delay-sensitive traffic in packet networks," *Proc. IEEE Infocom,* pp 0323-0332, 1991.

[13] Haslett, J. and Raftery, A. E.(1989). Space-time modeling with long-memory dependence: assessing Ireland's wind power resource (with discussion). Applied Statistics 38, 1-50.

[14] Hosking, J. R. M. "Fractional differencing," *Biometrica 68*, pp. 165-176, 1981.

[15] Hosking, J. R. M. "Modeling persistence in hydrological time series using fractional differencing," *Water Resources Research*, Vol.20, No.12, pp. 1898-1908, 1984.

[16] Huang, C., M. Devetsikiotis, I. Lambadaris, and A. R. Kaye, "Self-similar modeling of variable bit-rate compressed video: a unified approach," *Proc. of ACM Sigcomm,* London, 1994.

[17] Hurst, H.E., "Long-term storage capacity of reservoirs," *Transactions of the American Society of Civil Engineers,* 116, pp 770-779, 1951.

[18] Kurose, J., "On computing per-session performance bounds in high-speed multi-hop computer networks," *Proc. of ACM Sigmetrics,* pp 128-139, Newport, June, 1992.

[19] S. Q. Li and C. L. Hwang., "Queue Response to Input Correlation Functions: Discrete Spectral Analysis," *IEEE/ACM Trans. on Networking,* Vol. 1, No. 5, Oct. 1993, pp. 522-533.

[20] S. Q. Li and C. L. Hwang., "Queue Response to Input Correlation Functions: Continuous Spectral Analysis," *IEEE/ACM Trans. on Networking,* Vol. 2, No. 6, Dec. 1994, pp. 678-692.

[21] Livny, M., B. Melamed and A.K. Tsiolis, "The impact of auto-correlation on queueing systems," *Management Science,* pp 322-339, March 1993.

[22] Leland, W.E., M.S. Taqqu, W. Willinger and D.V. Wilson, "On the self-similar nature of Ethernet traffic (extended version)," *IEEE Trans. Networking,* **2,** (1), 1-15, February, 1994.

[23] Mandelbrot, B.B., and J.W. Van Ness, "Fractional brownian motions, fractional noises and applications," *SIAM Review,* 10, pp 422-437, 1968.

[24] Mandelbrot, B.B., and J.R. Wallis, "Computer experiments with fractional gaussian noises, *Water Resources Research,* 5, pp 228-267, 1969.

[25] Mandelbrot, B.B., and J.R. Wallis, "Some long-run properties of geophysical records" *Water Resources Research,* 5, pp. 321-40, 1969.

[26] Norros, I., "On the use of fractional Brownian motion in the theory of connectionless networks," Preprint, September 24, 1994.

[27] Pancha P., and M. El Zarki, "Variable bit rate video transmission" *IEEE Commun. Mag.*, Vol.32, No.5, pp. 54-66, May 1994.

[28] Pancha P., and M. El Zarki, "Leaky bucket access control for VBR MPEG video," *Proc. IEEE INFOCOM*, pp 796-803, Boston, April 1995.

[29] Rose, O. "Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems," University of Wuerzburg. Institute of Computer Science Research Report Series. Report No. 101. February 1995.

[30] The S+ package, Version 3.0, Statistical Sciences, Inc., September 1991.

[31] Yang, T. and J. Pan, " A measurement-based loss scheduling scheme," *Proc. IEEE INFOCOM,* pp 1062-1071, San Francisco, March 1996.

[32] Zhang, H., and E.W. Knightly, "Providing end-to-end statistical performance guarantees with bounding interval dependent stochastic models," *Proc of ACM Sigmetrics,* pp 211-220, Nashville, May 1994.