

# Modeling and Synthesis of On-chip Dynamic Multimedia Traffic

Girish Varatkar      Radu Marculescu  
Department of Electrical and Computer Engineering  
Carnegie Mellon University, Pittsburgh, PA 15213 USA

## Abstract

The objectives of this paper are twofold: First, to introduce the self-similarity as a fundamental property exhibited by the bursty traffic between on-chip modules in typical MPEG-2 video applications. Indeed, statistical tests performed on relevant traces extracted from common video clips establish unequivocally the existence of self-similarity in on-chip video traffic. Second, we describe a technique for synthetically generating traces having statistical properties similar to those obtained from real video clips. Our proposed technique speeds up buffer simulations, allows media system designers to explore architectures more rapidly and use large media data benchmarks more efficiently. We believe that our findings open up new directions of research with deep implications on some fundamental issues in on-chip network design for multimedia applications.

**Keywords:** system-level design, on-chip networks, communication analysis, self-similarity, long-range dependence.

## 1. Introduction and objectives

Nowadays, people see the need for portable embedded multimedia appliances capable of handling advanced algorithms required for all forms of communication (text, speech and video). As a consequence, it is important to determine a common design “platform”, consisting of both hardware and software resources, that could be shared across multiple multimedia applications.

The system-level view of such a generic design “platform” is shown in Fig. 1. As we can see, it consists of both *fixed* processing resources (e.g. ASIC) and *programmable* resources (e.g. processor 1) that co-operate to run the target application (e.g. MPEG-2 audio/video decoder, e-mail, web, etc.). The overall goal of the system level design is then to find the best mapping of the target application onto the set of architectural resources while satisfying the imposed design constraints (e.g. minimum area, minimum power dissipation, best performance, etc.). Most notably, the transition from desktop multimedia to portable multimedia based on heterogeneous design platforms brings *concurrency and communication* as prime candidates for system-level analysis and optimization.

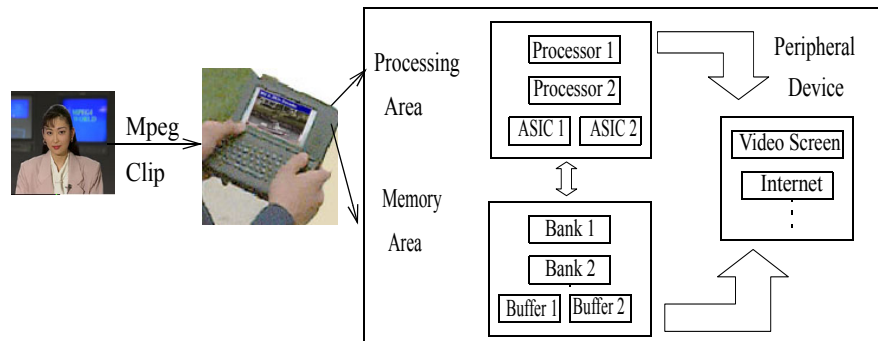


Fig. 1: A Generic Portable Embedded Multimedia System

In this paper we address the fundamental issue of selecting the optimal communication resources between different on-chip modules [15]. For complex systems composed of many heterogeneous components, the on-chip traffic produced among different modules has very diverse characteristics. Since the traffic patterns depend so much on the target application, it is necessary to judiciously allocate the on-chip communication resources, especially since the on-chip buffer space is usually very limited compared to real data networks.

Recently, Dally and Towles [1] proposed a novel on-chip interconnection network (Fig.2(a)) which can be used instead of the classical ad-hoc global wiring structure. It can offer well-controlled electrical parameters which enables high-performance circuits to reduce latency and increase bandwidth.

As shown in Fig.2(a), a chip employing such a communication architecture consists of several network *clients* (e.g. processors, memories, and custom logic) which are connected to a network that routes *packets* between them. Each client is placed on a tile and communicates with other clients (not only its neighbors) via the on-chip network. A

*router* is needed for each tile and it consists of several input-output controllers and their associated *buffers* (Fig.2(b)). From a practical point of view, the success of such an architecture depends on the ability to keep the overall area overhead to a *minimum*<sup>1</sup>. Since the area of the router is heavily dominated by the space occupied by the on-chip buffers, the problem of *optimal buffer sizing* becomes an issue of critical importance. Indeed, dropping or misrouting packets because of inappropriate buffer sizing reduce the overall performance and significantly increase the on-chip power dissipation. We also point out that this severe limitation of the on-chip buffer space comes in deep contrast with real data networks where there is ample room for very large buffers. This makes the on-chip network design problem quite unique and challenging.

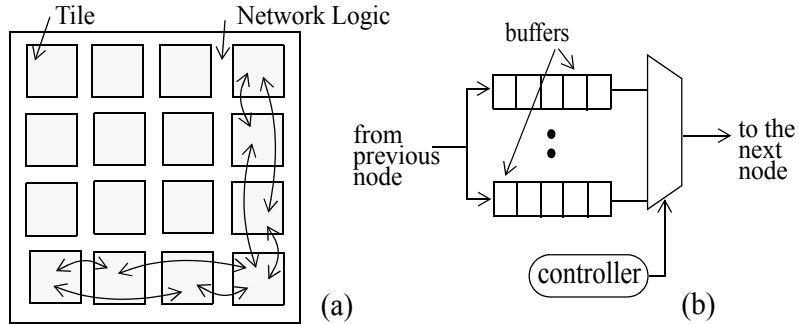


Fig. 2(a) Die module tiles and network logic, Fig.2(b) A generic input controller and its buffers.

This paper has two major objectives:

(a) First, to propose a novel traffic analysis approach as a precise way to characterize the on-chip communication pattern of different multimedia applications. More precisely, we propose a technique for traffic modeling based on *self-similar* or *Long-Range Dependent* (denoted as LRD<sup>2</sup>) stochastic processes. By analyzing the statistical properties of the arrival process at different points in a generic architecture like the one in Fig.2, for a standard MPEG-2 application, we characterize *quantitatively* the degree of self-similarity of the on-chip traffic using standard techniques based on Hurst parameter [3]. Knowing the Hurst parameter helps the designer to choose the *minimal* buffer size for the router at each tile in Fig.2(a) which will guarantee a certain *Quality of Service* (QoS).

(b) Second, we describe an algorithm for synthetically generating self-similar sample paths. We can control the length of the synthetic trace generated. If we choose to generate synthetic traces smaller than the original full length trace, then these synthetic traces can be used for simulating video traffic and estimating the buffer loss probabilities and the delay experienced by a macroblock at the buffer in a very short time compared to the full length simulation. This way, we can dramatically speed up the simulation of buffer and estimation of buffer loss probability.

The analysis we propose is especially relevant to the large class of *portable embedded multimedia systems* where the QoS requirements vary considerably from one media to another (e.g. video connections require consistently high throughput, but tolerate reasonable levels of jitter or packet errors) and buffer space is very limited. Consequently, the ability to explore several communication schemes while trying to satisfy QoS requirements is of crucial importance. As we show later in the paper, making use of the knowledge of traffic pattern for achieving a certain QoS with optimal resources proves to be extremely helpful.

## 1.1. Contributions of the paper

The contributions of this paper are threefold:

- First, we provide evidence about the presence of self-similar phenomena in on-chip *macroblock level* traffic generated by multimedia applications. This has very important consequences since self-similar processes have properties which are completely different from traditional *short-range dependent* autoregressive (ARMA) or Markovian processes which have been mostly used in system-level analysis [16-18]. We also point out that the traditional video traffic modeling has concentrated at the *frame level* traffic analysis while we look at the *macroblock level* on-chip traffic.
- Second, knowing the Hurst parameter which characterizes the traffic pattern for a particular application helps in finding the optimal buffer length distribution which turns out to be the critical issue for the routers at each node in the on-chip communication network.

1. The authors in [1] suggest about 6% area overhead of network logic for each tile.

2. We use interchangeably Self-Similarity and Long-Range Dependence.

- Third, we describe a method of generating *synthetic traces* with statistical properties similar to the original ones. We assess the performance of this synthetic trace generation method by comparing the bit loss probability at a buffer obtained by simulating a synthetic trace and comparing it with that of real trace. These synthetic traces can be used to dramatically speed up the simulation process for multimedia applications where tens of hours of simulation are typically required to gather useful information for on-chip network design.

Taken together, our proposed technique allows media systems designers implementing on-chip communication networks to choose the appropriate on-chip buffer sizes and use large multimedia data benchmarks more effectively. Ultimately, this will enable systems designers to optimally trade-off performance metrics and media quality.

## 1.2 Related work

In recent years, due to the advent of SoCs, the issue of efficient communication schemes - at chip level - received increased attention [19-21]. In particular, people have explored alternative solutions to the standard bus-based communication, especially for portable devices where tight power and performance constraints make the bus-based communication solution less attractive [22][25].

One problem with the approaches proposed so far for on-chip network exploration is that they rely entirely on explicit simulation. Consequently, due to the huge amount of data contained in multimedia applications, the simulation-based techniques tend to become prohibitively expensive in practice [14][19][24]. Typically, tens of hours are needed to simulate just a few minutes of video data. Moreover, randomly simulating video data, without a precise (quantitative) measure of traffic characteristics, is dangerous since the actual implications of traffic on the overall system performance may be completely obscured by using inappropriate data. These major issues prompted our attention towards developing a more *formal* approach for on-chip communication analysis with emphasis on precise characterization of multimedia traffic and using robust technique for simulation of synthetically generated video sequences so as to reduce the simulation time still maintaining the accuracy of the estimates of bit loss probability.

Our initial effort has concentrated on the literature available for real data networks trying to see what from that sizable body of knowledge can be ported to on-chip network design. Another question was to determine what is specific to on-chip network design as opposed to real networks design. Trying to answer these questions, we realized several things: First, the landscape of networking research has changed dramatically over the last decade. Since the seminal study of Leland et al. [2], which set the groundwork for considering self-similarity an important concept in understanding of data network traffic including modeling and analysis of network performance, an explosion of work has ensued investigating the multifaceted nature of this phenomenon [23][5]. Second, the long-held paradigm in the communication and performance communities that voice and data traffic can be adequately described by Markovian models has been seriously reconsidered [26]. Third, the most widely studied non-markovian LRD process is Fractional Gaussian Noise (denoted as FGN) process. There are several existing methods for synthesizing FGN processes[27] but each method has some advantages and some drawbacks.

Our paper is an attempt to bridge conceptually two very different worlds: data networks and on-chip networks. To this end, we first identify, at chip-level, a phenomenon discussed so far only in the context of traffic for real data networks. Second, we analyze the traffic of a multimedia application which targets a novel packet-based SoC implementation and illustrate the impact of our analysis on on-chip network design. We hope that beyond its practical implications, the connection that we create between these apparently so different domains will stimulate further research on formal methods for on-chip network design.

## 1.3 Organization of the paper

Section 2 defines the self-similarity and describes the statistical method used to establish the presence of LRD. Section 3 describes the method for generating synthetic LRD traces. In Section 4, we present a detailed analysis of traffic for the MPEG-2 video decoder and show the results for different video clips. In Section 5, we illustrate the implications of LRD on the on-chip network design and also show an example of how to speed up the buffer simulation using synthetic traces to estimate buffer loss probability. Finally, we conclude by summarizing our main contribution.

## 2. Long-Range Dependence: Definition and Properties

*Self-similarity* and fractals are concepts pioneered by Mandelbrot [11][12]. They describe the phenomenon where a certain property of a natural image or a time series is preserved with respect to *scaling* in space and/or time. If an object is self-similar then its parts, when magnified, resemble - in a suitable sense - the shape of the whole. In this section we give a brief description of the concept of self-similarity, discuss its most important mathematical properties and outline some statistical methods for analyzing LRD data.

*Stochastic* self-similarity admits the infusion of probabilistic behavior in the deterministic fractals. Here the objects do not possess the *exact* resemblance of their parts at finer levels of detail. If we think, for instance, in terms of time series which may characterize some real data traces and relax a little bit the measure of resemblance, say, by focusing on certain statistics of rescaled time series, then it may be possible to expect an *approximate* similarity with respect to these relaxed measures. *Second-order* (or temporal) statistics are the statistical properties that capture burstiness (or variability) in time series which characterize, for instance, traffic patterns in real networks [2]. In particular, the *autocorrelation function*, as a function of the time lag, decreases *polynomially* rather than exponentially. The existence of such non-trivial correlation “at a distance” is referred to as LRD and it is formally defined as follows.

Let  $X = (X_t : t = 0, 1, \dots)$  be a wide-sense stationary stochastic process with mean  $m$ , variance  $\sigma^2$  and autocorrelation function  $r(k)$ ,  $k \geq 0$ . According to [2]  $X$  is said to exhibit *long-range dependence* if

$$r(k) \sim k^{-\beta} L_1(t) \text{ as } k \rightarrow \infty, \quad (1)$$

where  $0 < \beta < 1$ ,  $L_1(t)$  is a slowly varying function and  $\sim$  denotes the “asymptotically close” condition; that is,  $\lim_{t \rightarrow \infty} L_1(tx)/L_1(t) = 1$ , for all  $x > 0$ .

From equation (1) we see that long-range dependence is characterized by an autocorrelation function that decays *hyperbolically* rather than exponentially fast. It also implies that the spectral density obeys a *power-law* function near the origin (also called  $1/f$ -noise). This captures the intuition behind long-range dependence, namely that while high-lag correlations are individually small, their *cumulative effect* counts and gives rise to features which are very different from those of short-range dependent processes. In what follows, we describe a method for testing long-range dependence[4] in any time series  $X$ .

## 2.1 Variance-Time Analysis

Let  $X$  be a wide-sense stationary time series. For each  $m = 1, 2, 3, \dots$  let  $X^{(m)} = X_k^m : k = 1, 2, 3, \dots$  denote the new wide sense stationary time series obtained by averaging the original time series  $X$  over non-overlapping blocks of size  $m$ .

That is, for each  $m = 1, 2, 3, \dots$ ;  $X^m$  is given by  $X_k^m = \frac{1}{m}(X_{km-m+1} + \dots + X_{km})$ ,  $k > 0$ .

The variances of  $X^m$ ,  $m = 1, 2, 3, \dots$  for *short-range dependent* (SRD) processes (e.g. Markov processes) will eventually decrease *linearly* in log-log plots against  $m$  with a slope equal to -1. On the other hand, for processes with long-range dependence, the variances of the aggregated processes  $X^m$ , decrease linearly (for large  $m$ ) in log-log plots against  $m$  with slopes arbitrarily flatter than -1. For a constant  $c$ , we have

$$\text{var } X^{(m)} \sim cm^{-\beta} \text{ as } m \rightarrow \infty, \quad (2)$$

with  $0 < \beta < 1$ . Actually, this value of  $\beta$  is related to the rate at which autocorrelations decay for large values of the lag. From equation (1), we can see that the autocorrelations decay *hyperbolically* with decay constant  $\beta$ . The relation between Hurst parameter  $H$  and the rate at which the autocorrelation decays is given by  $H = 1 - \beta/2$ . The Hurst parameter gets its name from the empirical law called as *Hurst effect*, which is observed in many naturally occurring time series [3].

## 3. Synthetic trace generation

Our main objective is to compact long input video sequences into much shorter sequences so as to reduce the simulation time by orders of magnitude and still maintain the accuracy of the buffer-loss and delay estimates. We use the concept of long-range dependence as a fundamental property of typical MPEG-2 video clips which should be preserved during the compaction process.

As a first step towards the synthetic MPEG generation, we have to identify the exact statistical properties of the original clip viewed at macroblock level which we need to preserve. We believe that the first and the second order statistics are the most relevant properties of the video clip that should be preserved in order to preserve the pattern of buffer storage in a mpeg-2 decoder. The first order statistics are captured by the distribution function. But the buffer occupancy pattern will depend not only on the distribution of sizes of macroblocks but also upon the *order* in which macroblocks arrive. The buffer loss probability will be drastically different in a clip in which many big macroblocks

arrive in a burst compared to another clip in which the macroblocks of all sizes arrive uniformly distributed over time. This information is captured by the autocorrelation function.

To capture the long-range slowly decaying autocorrelation function, we need to preserve the Hurst parameter of the original data. This will preserve the *temporal structure* of the arrival process of the macroblocks. We describe here a method based on the Discrete Time Fourier Transform (denoted as DTFT), for synthesizing a trace with desired Hurst parameter.

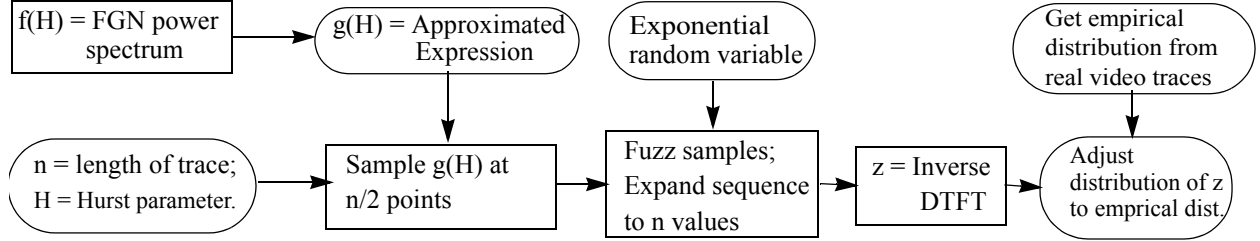


Fig. 3 Synthetic Trace Generation Procedure

The strategy can be summarized as follows. The power spectrum of a *fractional gaussian noise* (FGN) process, which is one kind of self-similar process, has a closed form expression given by

$$f(\lambda;H) = A(\lambda;H)[|\lambda|^{-2H-1} + B(\lambda;H)] \quad (3)$$

for  $0 < H < 1$  and  $-\pi \leq \lambda \leq \pi$ , where

$$A(\lambda;H) = 2 \sin(\pi H) \Gamma(2H + 1) (1 - \cos \lambda) \quad (4)$$

$$B(\lambda;H) = \sum_{j=1}^{\infty} [(2\pi j + \lambda)^{-2H-1} + (2\pi j - \lambda)^{-2H-1}] \quad (5)$$

The FGN process is widely studied and the closed form expression  $f(\lambda, H)$  is approximated by a simpler expression [27] in which the infinite summation in equation (5) is approximated by a simpler expression. So now we have a simple closed form expression for the power spectrum of FGN process. To generate a synthetic sequence of length  $n$ , the power spectrum expression is sampled at  $n/2$  equidistant points in the frequency domain lying in  $(0, \pi)$ . Then a sequence of complex numbers corresponding to this power spectrum is generated. It is in some sense a frequency-domain sample path. The complex conjugates of the frequency domain sample path complete the sequence to make it a full length sequence. The inverse-Discrete Time Fourier Transform of this sample will then be the time-domain counterpart with real numbers and having the autocorrelation function implied by the original expression of the power spectrum. Since autocorrelation and power spectrum form a Fourier transform pair, the time domain sample path will then have the long-range-dependent autocorrelation with Hurst parameter  $H$ . The details of this method can be found in [27]. The main steps in the procedure are:

1. Construct  $\{f_1, f_2, \dots, f_{n/2}\}$  where  $f_i$  represent the sampled values of the approximated expression of the power spectrum and  $f_i$  lie equally spaced between  $(0, \pi)$ .
2. Multiply each  $f_i$  by an independent exponential random variable with mean 1.
3. Construct a sequence of complex values  $\{z_1, z_2, \dots, z_{n/2}\}$  such that  $|z_i| = \sqrt{f_i}$  and the phase is uniformly distributed in the range  $(0, 2\pi)$ .
4. Expand the sequence from  $n/2$  values to  $n$  values by taking complex conjugates of the  $z$  sequence. This now corresponds to the Fourier transform of a real-valued signal.
5. Inverse Fourier transform of the full length  $z$  sequence now gives the FGN sample path

The problem of synthesizing statistically similar trace is still partly solved because the intermediate synthetic data has LRD autocorrelation but still it has gaussian distribution around mean zero while the real data from video clips need not have gaussian distribution. So we need to transform the distribution function. Let  $F_X(x)$  be the gaussian probability distribution function of the time domain inter-mediate synthetic trace  $X_k$ . Let  $F_Y(y)$  be the

probability distribution function of the original video clip data obtained empirically. Then we can generate the process  $Y_k$  using the following transformation [28]

$$Y_k = h(X_k) = F_Y^{-1}(F_X)(X_k) \quad (6)$$

An important issue regarding this transformation is that if the process  $X_k$  is a self-similar gaussian process with Hurst parameter  $H$ , then the nature of the process  $Y$  will also be self-similar with the same Hurst parameter  $H$  [28].

We have thus synthesized the artificial traces which mimic the original video clip in terms of the first and second order statistical properties. The number of macroblocks to be generated is now under our control and thus we can effectively achieve wide range of compression of the original clip.

## 4. Case Study: The MPEG-2 video decoder

Our main observation is that, the traffic between different modules for a MPEG-2 decoder exhibits long-range dependence. This is explained through the example of an MPEG-2 video decoder (Fig.4a) [9]. The decoder consists of the VLD (Variable Length Decoder), IQ (Inverse Quantization), the IDCT (Inverse Discrete Cosine Transform), the Motion Compensator (MC), and the associated buffers.

### 4.1 Modelling and measurement setup

We model the MPEG-2 Video decoder using the Stateflow component of Matlab which uses the semantics of Statecharts, formally proposed by Harel [10]. Statecharts can be used to describe the behavior of complex concurrent systems characterized by event-driven approach.

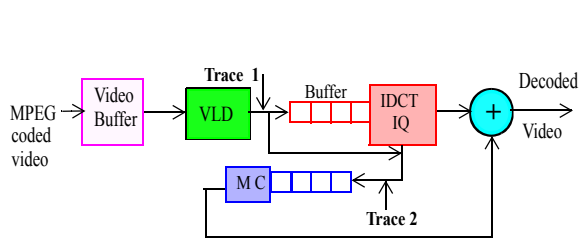


Fig.4a The block diagram of the MPEG-2 decoder

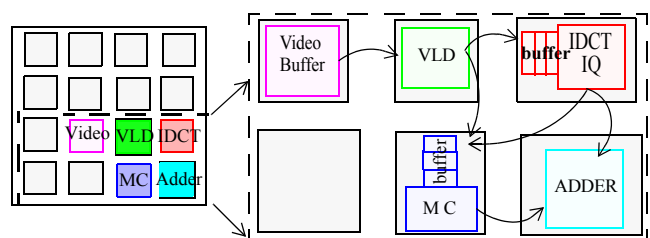


Fig.4b A possible mapping of the MPEG-2 decoder onto architecture in Fig.1

To create the Stateflow model of the MPEG-2 video decoder, the *sequential* C-code of the decoder was split into several processes and the communication among processes made explicit by using synchronization signals. We model the process graph obtained from the application in Fig.4a following the *Producer-Consumer* paradigm; that is, we describe the VLD process as the *Producer* and the IDCT/IQ unit as the *Consumer*. We assume that all computing processes are mapped onto the architecture discussed in Section 1 as shown in Fig.4b. The remaining unused tiles can be used to map other applications (e.g. audio decryption, audio/video synchronization etc.).

Using the *Mpegstat* tool developed at Berkeley [13], we analyze a MPEG-2 video stream at the *macroblock level* and find the detailed information about the *macroblocks* in the frames of the video. Depending upon whether a frame is of I, P or B type, the macroblocks are processed differently. An I type macroblock is processed only by the VLD and IDCT. Some of the P/B macroblocks need to be processed by VLD, IDCT and MC block with motion compensation performed using one frame stored in the memory. The P/B macroblocks of “skip” type do not need to be processed by IDCT and just a single frame memory access is needed. Thus macroblocks follow different paths in the block diagram and take different time to process. This results in various traffic patterns for different videos.

We monitored the arrival processes at the IDCT and MC modules recording their corresponding traces (that is, *Trace 1* and *Trace 2* in Fig.4a). The corresponding traces obtained were further evaluated using the analytical procedures discussed in Section 2. Using these methods, we were able to obtain the variance-time plots and R/S plots for the two traces. These results are discussed in the following section.

### 4.2 Results

Our approach to traffic modeling is “data driven”. We rely upon four video sequences (*Clouds*, *Simpsons*, *Disc ir*, *Hawaii*) of different frame sizes ranging in length from 27 seconds (88000 macroblocks) to 1 second (43000 macroblocks). This represents all kinds of different of scenes as shown in Table 1.

We focus on long sequences ( $X_i: i = 1, 2, \dots, N$ ) of data, where  $X_i$  represents the *number of bits* which contain the compressed and coded information for a macroblock in a frame of an MPEG video. Based on statistical analysis of

the sequences, our main finding is that LRD is a characteristic of the MPEG video traffic traces between different modules of MPEG decoder. The monitored trace file consists of two columns. The first column gives the time measured from the beginning of the trace at which a block of the video stream arrived at a module in the system. The second column gives the integer size in bits of the macroblock. We consider the discrete version of the process where the process is averaged within a window of size  $\delta$  .[8]

| Video Clip      | I frames | P frames | B frames | Macroblocks per frame |
|-----------------|----------|----------|----------|-----------------------|
| <i>Clouds</i>   | 24       | 12       | 0        | 1200                  |
| <i>Simpsons</i> | 136      | 136      | 542      | 108                   |
| <i>Disc_ir</i>  | 18       | 9        | 0        | 1024                  |
| <i>Hawaii</i>   | 195      | 96       | 0        | 300                   |

Table 1. Statistics for different clips

To compute the  $H$  parameter, we perform test based on the variance-time analysis of the time series  $X$  as described in Section 2.1. The variance-time plots are obtained by plotting  $\log(\text{var}(X^m))$  against  $\log(m)$ . As an illustration, Fig.5 shows the plot corresponding to the video clip *Simpsons* listed in Table 1.

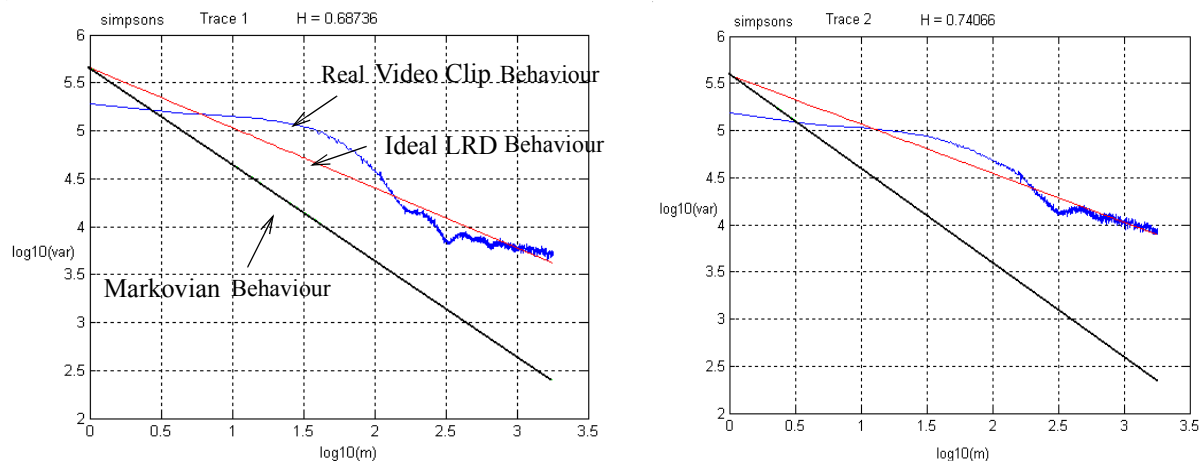


Fig.5: Variance-time plots for *simpsons* video clips at the IDCT module in the (trace 1 in Fig.4a) and at the MC module (trace 2 in Fig.4a) in the MPEG-2 decoder. The graph shows the least square fit line as well as the line with slope -1. The least square fit line can be seen to lie above the line with -1 slope which shows self-similarity. For large  $m$ , the slope of best fit line lies in  $(-1,0)$ .

We also verified our results using the standard *rescaled adjusted range statistics method (R/S method)* of analysis of the time series. Details about this *R/S* analysis method can be found in [2] For convenience, a summary of the estimated Hurst parameters is also given in Table 2. As we seen the values of  $H$  lie between 0.5 and 1 clearly indicating the presence of long-range dependence. Also, the values of  $H$  obtained from both methods are sufficiently close to each other to further support the claim about the presence of LRD.

| Video Clip      | Trace 1<br>H by Variance-time<br>method | Trace 1<br>H by<br>R/S plot method | Trace 2<br>H by Variance-time<br>method | Trace 2<br>H by<br>R/S plot method |
|-----------------|---|------------------------------------|---|------------------------------------|
| <i>Clouds</i>   | 0.7240                                  | 0.7646                             | 0.7603                                  | 0.7639                             |
| <i>Simpsons</i> | 0.6874                                  | 0.7432                             | 0.7407                                  | 0.7943                             |
| <i>Disc_ir</i>  | 0.8108                                  | 0.8180                             | 0.8421                                  | 0.8131                             |
| <i>Hawaii</i>   | 0.7238                                  | 0.7453                             | 0.5455                                  | 0.6839                             |

Table 2. Hurst parameter values for different clips by two methods for two different traces.

## 5. Implications of LRD traffic in designing on-chip networks

Beyond its statistical significance, long-range dependence has considerable impact on queueing performance of on-chip network. Only a small number of analytical queueing results are available for long-range dependent traffic[6]. In traffic analysis, we typically deal with time series with hundreds of thousands of observations. If we try to fit the best ARMA model to such a process, then the number of parameters will tend to infinity. Using an excessive number of parameters is undesirable as it increases the uncertainty of the statistical inference and parameters are difficult to interpret. Thus we need to model these processes with parametrically parsimonious models.

### 5.1 Buffer length prediction using Hurst parameter

Norros in [7] used Fractional Brownian Motion (FBM) model which parsimoniously captures LRD effects. This model finds out the *lower bound* for the probability that the queue length  $Q$  exceeds a certain buffer size  $x$ , under the assumption of having an infinite buffer. Mathematically:

$$P(Q > x) \sim \exp[-cx^{2-2H}] \quad (7)$$

with

$$c = \frac{m^{2H-1}(1-\rho)}{2a} \left[ \left( \frac{1-H}{H} \right)^H + \left( \frac{H}{1-H} \right)^{1-H} \right]^2 \quad (8)$$

where:  $m$  is the mean input rate,  $\rho$  is the utilization of the queue (that is, the ratio of average service time to average interarrival time);  $H$  and  $a$  are the Hurst parameter and the ‘‘peakedness’’ values which are obtained from variance-time plots in Fig.4.

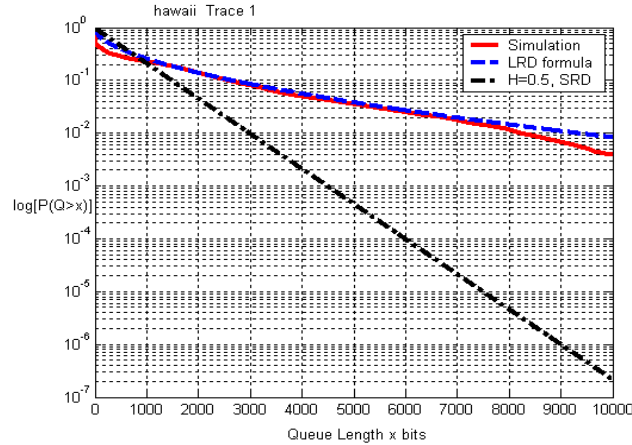


Fig.6 Complementary queue length distribution plots predicted by eqs. (7) and (8) for different video clips. The dashed curves indicate simulation results for an infinite queue with the arrival process following empirical trace (server utilization is 0.5). The straight lines indicate the prediction by a SRD model ( $H = 0.5$  in eqs.(7) and (8)).

To assess the *accuracy* and *impact* of our predictions on the overall performance of the on-chip network, the complementary buffer length distributions for four different video traces are shown in Fig.6. (The values of  $H$  and  $a$  from the Fig.5 were used to plot these graphs). In these graphs, the dashed curves indicate the *predicted* probability values given by eqs. (5) and (6) while the continuous curves indicate the results obtained by *simulation*. There are a few conclusions which can be derived from these plots:

- First, the predicted and simulated curves show a very good agreement as a function of buffer length. That is, the small difference between them is because the simulation corresponds to just one instance of the arrival process while the analytical formula gives the result averaged over *many* traces.
- Second, the dash-dot lines (obtained for  $H = 0.5$ ) correspond to short-range dependent (SRD) models like Markovian ones. From plots in Fig. 6, we can see that Markovian models significantly underestimate (typically 1-2 orders of magnitude) the buffer overflow probabilities which may cause severe performance degradation at chip-level.

There is also another way of using the plots in Fig. 6 (and therefore eq. (7)) for on-chip network design. For instance, if the QoS needed by the target application asks for not more than 1% of lost macroblocks, then from the first plot in Fig.6, one can easily see that we need a buffer length of 9000 bits at the IDCT module. This way, we have a theoretical basis for choosing the buffer length. On the other hand, the Markovian analysis will predict a buffer length of 3000 bits and that will result in around 10% bits lost (instead of the target 1%). This represents a very serious performance degradation for an MPEG video decoder.

To summarize, we can predict the buffer length more *accurately* using the LRD traffic analysis compared to Markovian analysis. We point out that this prediction of buffer length distribution *does* account for the bursty behavior of the traffic. This is in deep contrast with standard Markovian analysis which *under-estimates* the buffer length since the distribution of the arrival process is assumed to be short-range dependent i.e. exponential.

Last but not least, the LRD analysis framework offers a *fast* approach for buffer sizing compared to simulation-based buffer sizing. More precisely, for the simulation-based approach for buffer sizing, if we want to assess the impact of changing the speed of the processors in the on-chip network, then we need to rerun the simulations all over again for all the typical video clips. On the contrary, in the LRD-based analysis, the value of  $H$  will *not* be affected as it depends only on the underlying video clip statistical properties. Consequently, we just need to change the value of utilization factor  $\rho$  in eq.(8) and get the new buffer length.

## 5.2 Application of synthetic traces in speeding up buffer simulations

In order to evaluate the ‘goodness’ of our synthetic trace generation procedure, we perform the following simulation. We look at a set of three actual MPEG clips and obtain their macroblock level traces using the *Mpegstat* tool [13]. From one particular video clip, each macroblock is assumed to be arriving at a constant time interval to the buffer at the VLD. But different clips start sending macroblocks at different times which are uniformly distributed in one macroblock interval of one another. Thus macroblocks from the same source arrive at regular intervals of one another, but the packets from different sources arrive at different times, which are uniformly distributed within one macroblock interval. All these three sources are statistically multiplexed into a common buffer as shown in Fig.7. The server at the buffer is assumed to be serving the macroblocks at constant drain rate in terms of the number of bits processed per second. By varying this drain rate, the utilization  $\rho$  of the queue can be varied. Using the same setup, we then replace each video source by a synthetically generated trace, which is *half the length* of the original full length trace, having the same Hurst parameter as that of the original video and perform the simulations for buffer loss probability and delay again. Similar simulation results are available in the literature[29][30] but the analysis there is at the frame level and the models used in them are Markovian models.

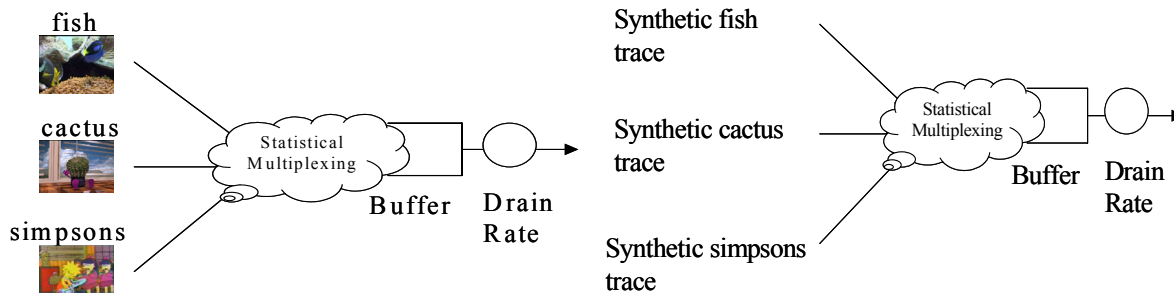


Fig.7 Experimental setup for evaluating delay and loss for statistically multiplexed traces

The results of our simulations are shown in the Fig.8. We can see from the plots that the loss probability curves for the real trace and the synthetic trace are practically the same for high levels of utilization more than 0.25. The delay curves can also be seen to be close to each other. Note that the simulation time for the synthetic traces is reduced to half because the length of the synthetic trace is half of that of the original trace.

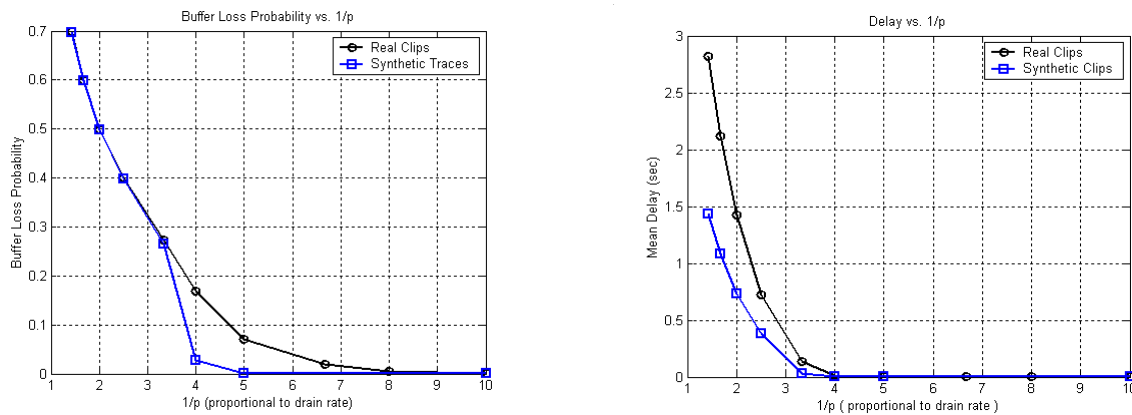


Fig.8 Loss probability and delay for real and synthetic traces using multiplexed streams

## 6. Conclusion

We have presented a technique for on-chip traffic analysis using self-similar processes. For a recently proposed communication architecture based on packet switching, we have shown that, under various input traces, the arrival process at different nodes, for an MPEG-2 video application, exhibits self-similar phenomena. Characterizing the degree of self-similarity via the Hurst parameter helps in finding the optimal buffer length distribution which turns out to be the critical issue for the routers at each node in the on-chip communication network. Finally, the synthetic trace generation procedure can be used to effectively reduce the simulation time for calculating the buffer loss probability and delay. We believe that our findings open up new directions of research with deep implications on some fundamental issues in on-chip network design.

## References

- [1] Dally, W., Towles, B., 'Route Packets, Not Wires: On-chip Interconnection Networks,' *Proc. DAC, Las Vegas, NV*, June 2001.
- [2] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. 'On the self-similar nature of ethernet traffic,' *IEEE/ACM Trans. on Networking*, Vol.2, No.1, Feb.1994.
- [3] J. Beran, 'Statistics for Long-Memory Processes,' Chapman & Hall, New York, 1994.
- [4] D. R. Cox, 'Long-Range dependence: A Review,' in *Statistics: An Appraisal*, H.A.David and H.T.David, Eds., The Iowa State University Press, Ames, Iowa 1984.
- [5] J. Beran, R. Sherman, M. S. Taqqu and W. Willinger, 'Long-Range Dependence in Variable-Bit-Rate Video Traffic,' *IEEE Trans. Commun.*, Vol. 43, No. 2/3/4, 1995.
- [6] A. Erramilli, O. Narayan and W. Willinger, 'Experimental Queueing Analysis with Long-Range Dependent Packet Traffic,' *IEEE/ACM Trans. on Networking*, Vol.4, No.2, April 1996.
- [7] I. Norros, 'A storage model with self-similar input,' *Queueing Systems* Vol. 16, 1994.
- [8] P. Abry and D. Veitch, 'Wavelet Analysis of Long-Range Dependent Traffic,' *IEEE Trans. on Info. Theory*, Dec. 1997.
- [9] Sikora T., 'MPEG Digital Video Coding Standards,' *IEEE Signal Processing Magazine*, September 1997.
- [10] D. Harel, 'Statecharts: A visual formalism for complex systems,' in *Sci. Comp. Prog.*, Vol. 8, 1987.
- [11] B. B. Mandelbrot and J. R. Wallis, 'Computer Experiments with Fractional Gaussian Noises,' *Water Resources Research*, vol.5, 1969.
- [12] B. B. Mandelbrot and M. S. Taqqu, 'Robust R/S Analysis of Long Run Serial Correlation', *Proc. 42nd Session ISI*, Book 2, 1979.
- [13] <http://bmrc.berkeley.edu/ftp/pub/mpeg/stat/>
- [14] K. Lahiri, A. Raghunathan, S. Dey, 'Evaluation of the Traffic-Performance Characteristics of System-on-Chip Communication Architectures', *Proc. Intl. Conf. on VLSI Design*, Bangalore, India, January 2001.
- [15] K. Keutzer, S. Malik, A. R. Newton, J. M. Rabaey, A. Sangiovanni-Vincentelli, 'System-Level Design: Orthogonalization of Concerns and Platform-Based Design,' *IEEE Trans. on CAD*, Vol.19, No.12, Dec. 2000.
- [16] A. Mathur, A. Dasdan, R. Gupta, 'Rate Analysis for Embedded Systems,' *ACM Trans. on Design Automation of Electronic Systems*, Vol. 3, No. 3, July 1998.
- [17] L. Benini, A. Bogliolo, G. A. Paleologo, G. De Micheli, 'Policy Optimization for Dynamic Power Management,' *IEEE Trans. on CAD*, Vol.18, No.6, June 1999.
- [18] A. Nandi and R. Marculescu, 'Probabilistic Application Modeling for System-Level Performance Analysis,' *Proc. DATE*, Munich, March 2001.
- [19] K. Lahiri, A. Raghunathan, S. Dey, 'Fast Performance Analysis of Bus-based System-on-chip Communication Architecture,' *Proc. ICCAD*, Nov. 1999.
- [20] M. Gasteir, M. Glesner, 'Bus-based Communication Synthesis on System Level,' *ACM Trans. Design Automation Electronic Systems*, Jan. 1999.
- [21] T. Yen, W. Wolf, 'Communication Synthesis for Distributed Embedded Systems,' *Proc. ICCAD*, Nov. 1995.
- [22] J. A. Rowson and A. Sangiovanni-Vincentelli, 'Interface Based Design,' *Proc. DAC.*, June 1997.
- [23] K. Park, W. Willinger, (Eds.), 'Self-Similar Network Traffic and Performance Evaluation,' J. Wiley and Sons, 2000.
- [24] M. Sgroi, M. Sheets, A. Mihal, K. Keutzer, S. Malik, J. Rabaey, A. Sangiovanni-Vincentelli, 'Addressing the System-on-a-chip Interconnect Woes Through Communication-Based Design,' *Proc. of DAC.*, Las Vegas, June 2001.
- [25] F. Karim, A. Nguyen, S. Dey, R. Rao, 'On-chip Communication Architecture for OC-768 Network Processors,' *Proc. of DAC.*, Las Vegas, June 2001.
- [26] V. Paxson, S. Floyd, 'Wide-Area Traffic: The Failure of Poisson Modeling', *Proc. ACM SIGCOMM'94*, London, UK, 1994.
- [27] V. Paxson, 'Fast Approximation of Self-Similar Network Traffic', Available through <ftp://ftp.ee.lbl.gov/papers/fast-approx-selfsim.ps.Z>, 1995.
- [28] C. Huang, M. Devetsikiotis, I. Lambadaris, A. Roger Kaye, 'Modeling and Simulation of Self-Similar Variable Bit Rate Compressed Video: A Unified Approach', *Proc. of ACM SIGCOMM*, 1995.
- [29] D. Turaga, T. Chen, 'Hierarchical Modeling of Variable Bit Rate Video Sources', *Packet Video 2001*, Kyongju, Korea, Apr. 30 - May. 1, 2001
- [30] D. Turaga and T. Chen, 'Activity-Adaptive Modeling of dynamic multimedia traffic', *IEEE Intl. Conf. on Multimedia and Expo.*, New York, July 2000