

# ON-CHIP COMMUNICATION ANALYSIS FOR MULTIMEDIA APPLICATIONS<sup>†</sup>

Girish Varatkar      Radu Marculescu

Department of Electrical and Computer Engineering  
Carnegie Mellon University, Pittsburgh, PA 15213-3890

## ABSTRACT

The objective of this paper is to introduce self-similarity as a fundamental property exhibited by the bursty traffic behavior between different on-chip modules in typical MPEG-2 video applications. Statistical tests performed on relevant traces extracted from common video clips establish unequivocally the existence of self-similarity in on-chip video traffic. Using a generic on-chip communication architecture, we discuss the implications of our findings on on-chip buffer space allocation. We also describe a synthetic trace generation procedure for speeding up the buffer simulation process.

## 1. INTRODUCTION AND OBJECTIVES

Nowadays, people see the need for portable embedded multimedia appliances capable of handling advanced algorithms required in 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. It consists of both *fixed* processing resources (e.g. ASICs) and *programmable* resources (e.g. processors) that co-operate to run the target application (e.g. MPEG-2 audio/video decoder, 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 [4].

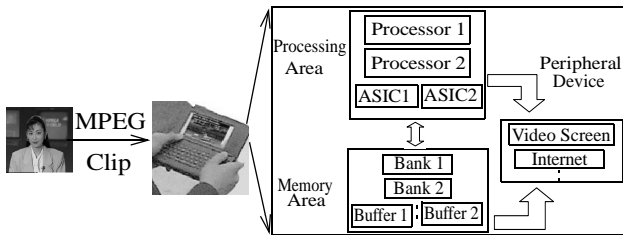


Fig. 1 A generic portable embedded multimedia system

In this paper we address the fundamental issue of selecting the optimal communication resources between different on-chip modules. 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.

In order to exploit the regularity in large SOC design, 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. What makes this generic architecture very attractive is that it can offer well-controlled electrical parameters which enables high-performance circuits to reduce latency and increase bandwidth for on-chip networks.

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)).

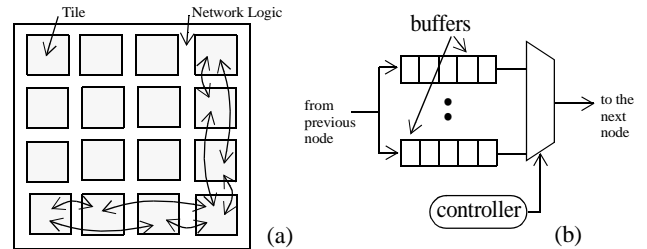


Fig. 2(a) Die module tiles and network logic, Fig.2(b) A generic input controller and its buffers

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. This makes the on-chip network design problem quite challenging and especially relevant to the large class of portable embedded multimedia systems where the QoS requirements vary considerably from one media to another and buffer space is very limited. As we will show later in this paper, making use of the knowledge of the traffic pattern for achieving a certain QoS with optimal resources turns out to be extremely helpful.

### 1.1. Contributions of the paper

The contributions of this paper are twofold:

- First, we provide evidence about the presence of *self-similar phenomena* in *on-chip traffic* generated by multimedia applications. This has very important consequences on queueing properties of on-chip networks because self-similar processes have

<sup>†</sup>Research supported by NSF CCR-00-93104, DARPA/Marco Gigascale Research Center (GSRC), and SRC 2001-HJ-898.

properties which are completely different from traditional *short-range dependent* autoregressive (ARMA) or Markovian processes traditionally used in system-level analysis [3][5][6].

- Second, knowing the *Hurst parameter* [9] which characterizes the traffic pattern for a particular application can be used to generate *synthetic traces* with statistical properties similar to the original ones. These synthetic traces can be used to speed up the simulation process for multimedia applications where tens of hours of simulation is typically required to get useful information for on-chip network design.

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

## 1.2. Organization of the paper

Section 2 defines the self-similarity and describes a statistical method used to establish the presence of *long-range-dependence* (denoted as LRD). In Section 3, we present a detailed analysis of traffic for the MPEG-2 video decoder and show the results for four very different video clips. In Section 4, we illustrate the implications of the LRD on the on-chip network design process. Finally, we conclude by summarizing our main contribution.

## 2. WHAT IS SELF-SIMILARITY?

*Self-similarity* and fractals are concepts pioneered by Mandelbrot [8]. They describe the phenomenon where a certain property of an 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.

*Stochastic* self-similarity admits the infusion of probabilistic behavior. Unlike the deterministic fractals, 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 [7][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 distance” is referred to as LRD and it is formally defined in next section.

### 2.1. LRD: Definition and properties

In this section we give a brief description of the concept of LRD and outline statistical method for analyzing LRD data.

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 slowly varying function and  $\sim$  denotes the “asymptotically close” condition:  $\lim_{t \rightarrow \infty} L_1(tx)/L_1(t) = 1$ , for

all  $x > 0$ . From equation (1), we see that LRD is characterized by an autocorrelation function that decays *hyperbolically* rather than exponentially fast. There are several ways to test the presence of LRD [9]. In what follows, we describe the variance-time method for testing LRD in a time series.

### 2.2. 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  $m = 1, 2, \dots$ ;  $X^m$  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 LRD, 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, the value of  $\beta$  is related to the rate at which autocorrelations decay for large values of the lag. The relation between Hurst parameter  $H$  and the rate at which the autocorrelation decays is given by  $H = 1 - \beta/2$  [9].

## 3. TRAFFIC ANALYSIS FOR MPEG-2 VIDEO DECODER

Since on-chip buffer space is very limited, it is very important to use the appropriate traffic model to optimally allocate available communication resources. Our main observation is that, the traffic pattern between different modules for a MPEG-2 decoder exhibits LRD. This is explained subsequently through the example of a MPEG-2 video decoder (Fig.3a) [10].

### 3.1. Modelling and measurement setup

The decoder consists of the VLD (Variable Length Decoder), IQ (Inverse Quantization), IDCT (Inverse Discrete Cosine Transform), Motion Compensator (MC), and the associated buffers. We model the MPEG-2 Video decoder using the Stateflow component of Matlab which uses the semantics of Statecharts, formally proposed by Harel [11].

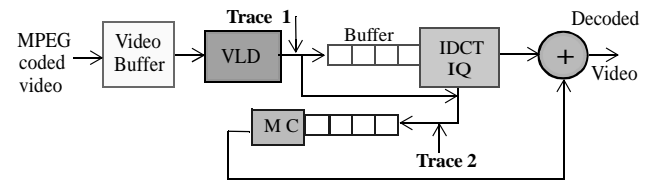


Fig.3a The block diagram of the MPEG-2 decoder

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 assume that all computing processes are mapped onto the architecture discussed in Section 1 as shown in Fig.3b. The remaining unused tiles can be used to map other applications (e.g. audio, encryption, etc.).

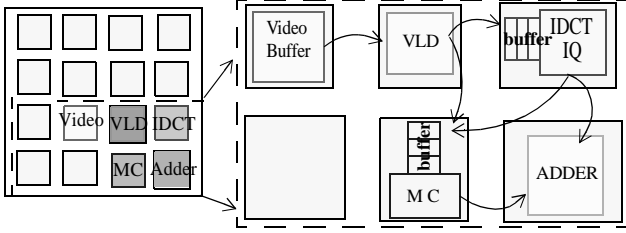


Fig.3b A possible mapping of the MPEG-2 decoder onto the architecture in Fig.1

Using the *Mpegstat* tool developed at Berkeley [12], we analyze MPEG-2 video streams and find detailed information about the *macroblocks* in the frames of the video. Depending upon the type of a macroblock (I, P or B), they follow different paths in the block diagram and then take different times to process. The sizes of the macroblocks in terms of the number of bits also vary; this results in various on-chip traffic patterns.

We monitored the arrival processes at the IDCT and MC modules recording their corresponding traces (*Trace 1* and *Trace 2* in Fig.3a). The corresponding traces obtained were further evaluated using the analytical procedure discussed in Section 2. Using this method, we were able to obtain the variance-time plots for the traces. The results are discussed in the following section.

### 3.2. Results and discussion

Our approach to traffic modeling is “data driven”. We rely upon four video sequences (*Fish*, *Simpsons*, *Disc\_ir*, *Hawaii*) of different video screen sizes ranging from 27 seconds (110000 macroblocks) to 1 second (43000 macroblocks). We focus on sequences ( $X_i; i = 1, 2, \dots, N$ ) where  $X_i$  represents the *number of bits* which contain the compressed and coded information for a *macroblock* of MPEG coded video.

To compute the  $H$  parameter, we use the variance-time analysis of the time series  $X$ . As an illustration, Fig.4 shows the plot corresponding to the video clip *Hawaii*. As we can see, the value of  $H$  is 0.72 (which is greater than 0.5) clearly indicating the presence of LRD.

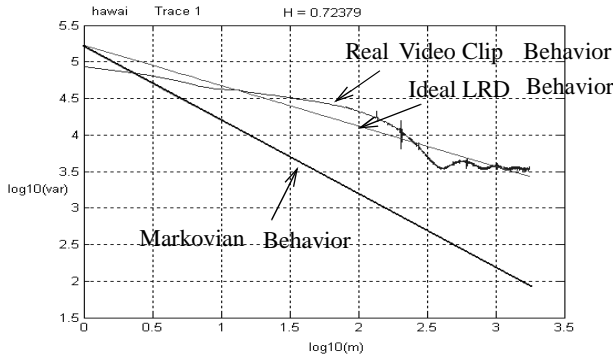


Fig.4: Variance-time plot for *Hawaii* at the IDCT module (trace 1 in Fig.3a) in the MPEG-2 decoder

We also test the presence of LRD using the *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 [9]. For convenience, a summary of the estimated Hurst parameters for all four clips using both the tests is given in Table 1. As we can see, the values of  $H$  obtained from *both* the methods are sufficiently close to each other to support the claim about the presence of LRD.

Table 1. Hurst parameter values for different clips estimated by two methods

Video Clip	Trace 1 H by Variance-time method	Trace 1 H by R/S plot method	Trace 2 H by Variance-time method	Trace 2 H by R/S plot method
<i>Hawaii</i>	0.7238	0.7453	0.5455	0.6839
<i>Fish</i>	0.7251	0.7147	0.6308	0.7136
<i>Simpsons</i>	0.6874	0.7432	0.7407	0.7943
<i>Disc_ir</i>	0.8108	0.8180	0.8421	0.8131

## 4. IMPLICATIONS IN DESIGNING ON-CHIP NETWORKS

Beyond its statistical significance, LRD has a considerable impact on queueing performance of on-chip network. Norros in [13] uses Fractional Brownian Motion (FBM) model to parsimoniously capture the LRD effects. Using this model, one can find 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:

$$P(Q > x) \sim \exp[-cx^{2-2H}] \quad (3)$$

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

where  $m$  is the mean input rate,  $\rho$  is the utilization factor (the ratio of average service time to average interarrival time);  $H$  and  $a$  are the Hurst parameter and the “peakedness” values obtained from variance-time plot in Fig.4.

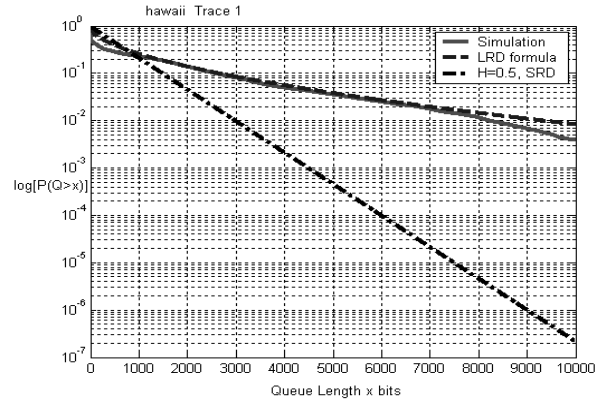


Fig.5 Complementary queue length distribution plots predicted by equation (3) and obtained by simulation of the *Hawaii* clip

### 4.1. Buffer length prediction

To assess the *accuracy* and *impact* of our predictions on the performance of the on-chip network, the complementary queue length distributions for all video clips in Table 1 were simulated<sup>1</sup> and plotted. A representative plot is shown in Fig.5. We can draw a few conclusions from this plot:

- First, the curves predicted by the formula in equation (3) and the curves obtained by simulation show a very good agreement as a function of buffer length.
- Second, the dash-dot lines (for  $H = 0.5$ ) correspond to Markovian (that is, SRD) models. From Fig.5, we can see that SRD

1. Similar plots were obtained for all the other video clips.

models significantly *underestimate* the buffer overflow probabilities which may cause severe performance degradation at chip-level.

#### 4.2. Simulation speedup using synthetic trace generation

In this section we describe a method of generating *synthetic traces* with statistical properties similar to the original traces obtained from real clips. These synthetic traces can be used to dramatically speed up the simulation process for calculating the buffer loss probability. The main steps in the procedure are:

1. Construct  $\{f_1, \dots, f_{n/2}\}$  where  $f_i$  represent the sampled values of the approximated expression of the power spectrum [14], 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  $(0, 2\pi)$ .
4. Expand the sequence from  $n/2$  values to  $n$  values by taking complex conjugates of the  $z$  sequence.
5. Inverse Fourier transform of the full length  $z$  sequence now gives the LRD sample path.
6. Transform the distribution function from the gaussian distribution to the empirically observed distribution.

In order to evaluate how good is our synthetic trace generation procedure, we perform the following simulations. We look at three real MPEG clips and obtain their macroblock level traces using the *Mpegstat* tool [12]. 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 time that are uniformly distributed. These three sources are statistically multiplexed into a common buffer as shown in Fig.6.

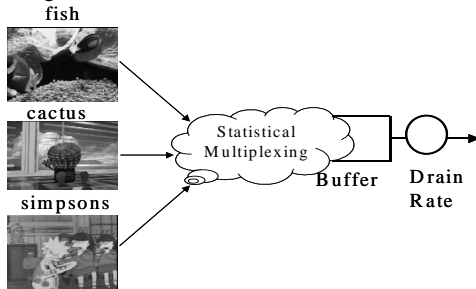


Fig.6 Experimental setup for evaluating buffer loss probability

In Fig.6, 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. 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, (but has the same Hurst parameter as that of the original video) and perform the simulations for buffer loss probability. The results of our simulations are shown in the Fig.7. One 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.

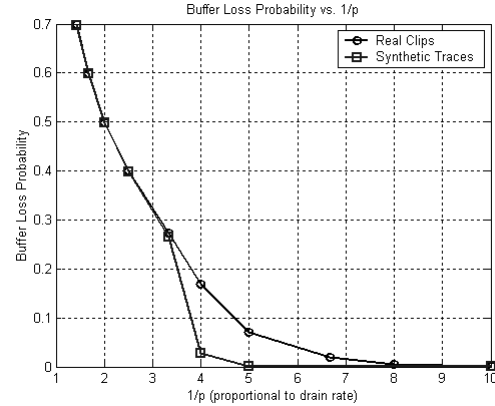


Fig.7 Buffer loss probability curve for real and synthetic traces

## 5. 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 the arrival process at different nodes, in an MPEG-2 video application, exhibits self-similar phenomena. Characterizing the degree of self-similarity via Hurst parameter helped in finding the *optimal* buffer length which is the critical issue for routers at each node in the on-chip communication network. Finally, a synthetic trace generation procedure can be used to significantly reduce the simulation time for calculating the buffer loss probability.

**Acknowledgements:** We would like to thank Mor Harchol-Balter (CMU) for stimulating discussions about LRD.

## 6. REFERENCES

- [1] W. Dally, B. Towles, 'Route Packets, Not Wires: On-chip Interconnection Networks,' *Proc. DAC, Las Vegas, NV*, June 2001.
- [2] W. E. Leland, M. S. Taquq, 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] A. Kalavade, P. Moghe, 'A tool for performance estimation of networked Embedded End-Systems,' *Proc. DAC, San Francisco, CA*, June 1998.
- [4] 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.
- [5] D. Turaga and T. Chen, 'Activity-Adaptive Modeling of dynamic Multimedia traffic,' *Proc. ICME, New York*, July, 2000.
- [6] A. Nandi, R. Marculescu, 'Probabilistic Application Modeling for System-Level Performance Analysis,' *Proc. DATE, Munich*, March 2001.
- [7] K. Park, W. Willinger, (Eds.), 'Self-Similar Network Traffic and Performance Evaluation,' J. Wiley and Sons, 2000.
- [8] B. B. Mandelbrot and J. R. Wallis, 'Computer Experiments with Fractional Gaussian Noises,' *Water Resources Research*, Vol.5, 1969.
- [9] J. Beran, 'Statistics for Long-Memory Processes,' Chapman & Hall, NY, 1994.
- [10] T. Sikora, 'MPEG Digital Video Coding Standards,' *IEEE Signal Processing Magazine*, September 1997.
- [11] D. Harel, 'Statecharts: A visual formalism for complex systems,' in *Sci. Comp. Prog.*, Vol. 8, 1987.
- [12] <http://bmrc.berkeley.edu/fip/pub/mpeg/stat/>
- [13] I. Norros, 'A storage model with self-similar input,' *Queueing Systems Vol.* 16, 1994.
- [14] G. Varatkar, R. Marculescu, 'Modeling and synthesis of on-chip multimedia traffic,' *Packet Video Workshop*, Pittsburgh, April 2002.