Traffic Analysis for On-chip Networks Design of Multimedia Applications†

Girish Varatkar Radu Marculescu

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

{gvv,radum}@ece.cmu.edu

Abstract: *The objective of this paper is to introduce self-similarity as a fundamental property exhibited by the bursty traffic between 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 video traffic. Using a generic communication architecture, we also discuss the implications of our findings on on-chip buffer space allocation and present quantitative evaluations for typical video streams. 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.*

Categories and Subject Descriptors: C.4 [**Performance of systems**]: Modeling techniques; B.8.2 [**Performance and reliability**]: performance analysis and design aids.

General terms: performance, design

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

1. Introduction and objectives

A fundamental issue in system-level design consists of selecting the optimal mechanism of communication between different on-chip modules [16][17]. 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 communication resources, especially since the on-chip buffer space is usually very limited.

Recently, Dally and Towles [1] proposed a novel on-chip interconnection network (Fig.1(a)) which can be used instead of the classical ad-hoc global wiring structure. What makes this generic architecture very attractive is that it offers well-controlled electrical parameters which enables high-performance circuits to reduce latency and increase bandwidth.

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

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAC 2002, June 10-14, 2002, New Orleans, Louisiana, USA.

Copyright 2002 ACM 1-58113-461-4/02/0006...$5.00

As shown in Fig.1(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.1(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*[1]. 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 reduces 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 unique and challenging.

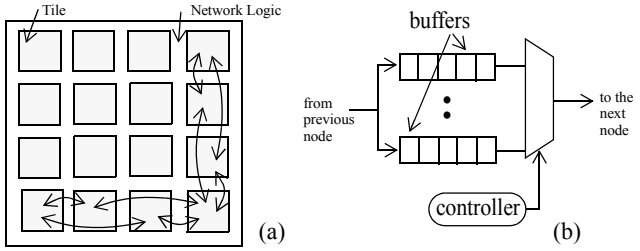

Fig.1(a) Die module tiles and network logic, Fig.1(b) A generic input controller and its buffers.

The objective of this paper is to propose a novel traffic analysis approach as a precise way to characterize the on-chip communication pattern of multimedia applications. More precisely, we propose a technique for traffic modeling based on *self-similar* or *Long-Range Dependent* (denoted as LRD[2]) stochastic processes. By analyzing the statistical properties of the arrival process at different points in a generic architecture like the one in Fig.1, for a standard MPEG-2 application, we first demonstrate that the self-similarity is a characteristic behavior of the on-chip traffic. Second, 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.1 which will guarantee a certain *Quality of Service* (QoS).

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 (LRD).

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 turns out to be extremely helpful.

1.1. Contributions of the paper

The contributions of this paper are threefold:

• First and foremost, we propose a completely new way to address the problem of on-chip network design. To this end, we show how *self-similar* (or *LRD*) processes can be used effectively to model the bursty traffic behavior at chip-level.

• Second, we provide evidence about the presence of self-similar phenomena in on-chip 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 [15][18-20].

• Third, knowing the Hurst parameter which characterizes the traffic pattern for a particular application can be used to generate *synthetic traces* with statistical properties similar to the original ones [30]. 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 buffer sizes and use large multimedia data benchmarks more effectively. 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 [21-25][28]. 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][21][27]. Typically, tens of hours are needed to simulate a few minutes of video data. Moreover, simulating randomly video data, without a precise (quantitative) measure of traffic characteristics, is dangerous since the actual implications of traffic on the system performance may be obscured by using inappropriate data. These issues prompted our attention towards a more *formal* approach for on-chip communication analysis with emphasis on precise characterization of multimedia traffic.

As such, 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 [2][29]. 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 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 describes the motivation behind self-similarity and its definition. In Section 3, we present a detailed analysis of traffic for the MPEG-2 video decoder and show the results for four 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 [11]. They describe the phenomenon where a certain property of an object - for instance, a natural image or a time series - is preserved with respect to *scaling* in space and/or time. If an object is self-similar (or fractal), then its parts, when magnified, resemble - in a suitable sense - the shape of the whole. For example, a two dimensional (2D) *deterministic* Cantor set is obtained by starting with a black unit square, scaling its size by 1/3, then placing four copies of the scaled square at the four corners, and repeating this process recursively ad infinitum (Fig.2).

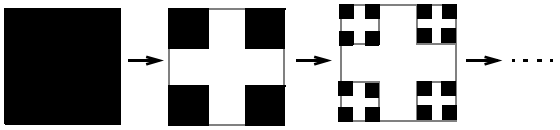

Fig.2. Deterministic fractal example: Two-dimensional Cantor set

The one-dimensional Cantor set can be obtained by projecting the 2D Cantor set onto a horizontal time axis. This can be further interpreted as an ON/OFF time series which model data traffic [26].

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 [2][5]. 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.

In the case of video traces, we concentrate on traffic characteristics at the *macroblock level*. If we look within each frame of a video, there are certain objects. All the macroblocks within a single object carry similar amounts of information and hence they get coded using almost the same number of bits. Since the macroblocks within an object generally occur next to each other in a frame, this leads to long range correlations. This may be the intuitive reason behind observing the LRD phenomenon in video traffic, at macroblock level.

We also note that, from a practical point of view, LRD has a considerable impact on queueing performance of the communication architecture. The traditional *short-range dependent* (or Markovian) processes have an autocorrelation function which

decays *exponentially* fast. But the LRD processes exhibit a much slower decay of correlations; that is, their correlation functions typically obey some power-law decay. Intuitively, the presence of LRD indicates that while long-range correlations are individually small, their *cumulative effect* is non-negligible. This produces scenarios which are drastically different from those experienced with traditional short-range dependent models such as Markovian processes. This is the subtle point where the long-range dependence analysis we propose surpasses classical Markovian analysis and proves its practical value.

2.1 Definition of Long-Range-Dependence

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

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

where $0 < \beta < 1$, $L_1(t)$ is a slowly varying function and ~ denotes the ''asymptotically close'' condition; that is, $\lim_{t \to \infty} L_1(tx)/L_1(t) = 1$, for all $x > 0$.

From equation (1) we see that LRD is characterized by an autocorrelation function that decays *hyperbolically* rather than exponentially fast. 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 LRD, namely that while high-lag correlations are individually small, their *cumulative effect* matters and gives rise to features which are very different from those of short-range dependent processes. In what follows, we describe two methods for testing LRD in any time series X.

2.2 Variance-Time Analysis

Let X be a wide-sense stationary time series. For each $m = 1,2,3,...$ let $X^{(m)} = X_k^m : k = 1, 2, 3,...$ 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 $m = 1,2,3,...$; X^m is given by $X_k^m = \frac{1}{m}(X_{km-m+1} + \ldots + X_{km})$, $k > 0$.

The variances of X^m, $m = 1,2,3,...$ for short-range dependent 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.

Cox [4] shows how a specification of the sequence $(\text{var}(X^m) : m > 0)$ is equivalent to a specification of the autocorrelations given by (1). More importantly, for a constant c, we have

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

with $0 < \beta < 1$. Actually, this value of β 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 β.

2.3 R/S Analysis

Historically, stochastic processes with long-range dependence are important because they provide an elegant explanation of an empirical law that has been observed in many naturally occurring time series. What has since come to be known as the *Hurst effect* can be described as follows. Given the observations $(X_k : k = 1,2,...,n)$ with sample mean $\bar{X}(n)$ and variance $S^2(n)$, the *rescaled adjusted range statistic* (denoted as *R/S*) statistics is given by

$$\frac{R(n)}{S(n)} = \frac{1}{S(n)}[Max(0,W_1,W_2,\ldots,W_n) - min(0,W_1,\ldots,W_n)] \tag{3}$$

where $W_k = (X_1 + X_2 + \ldots + X_k) - k\bar{X}(n)$, $1 \leq k \leq n$. In his study of the rescaled adjusted range [3], Hurst found that many historical records appeared to be well represented by

$$E[R(n)/S(n)] \sim cn^H, \text{ as } n \to \infty, \tag{4}$$

with Hurst parameter H about 0.7. On the other hand, if X_k 's are Gaussian pure noise or *short range dependent*, then $H = 0.5$ in equation (4) and the discrepancy is referred to as the Hurst effect. The Hurst effect is fully accounted for by stationary stochastic processes with long-range dependence. The relation between Hurst parameter and the rate at which the autocorrelation decays is given by $H = 1 - \beta/2$.

In practice, *R/S* analysis is based on a heuristic graphical approach, originally described in [11][12]. Formally, given a sample of N observations, $(X_k : k = 1, 2,..., N)$, one subdivides the whole sample into K *non-overlapping* blocks and computes the rescaled adjusted range $R(t_i, d)/S(t_i, d)$ for each of the new starting points $t_1 = 1, t_2 = N/K + 1, t_3 = 2(N/K) + 1, \ldots$ which satisfy $(t_i - 1) + d \leq N$. Here $R(t_i, d)$ is defined as in (3) with W_k replaced by $W_{t_i+k} - W_k$ and $S^2(t_i, d)$ is the sample variance of $X_{t_i+1}, X_{t_i+2}, \ldots, X_{t_i+d}$. The slope of the least square fit line fitting the set of values of *R/S* gives the asymptotic value of parameter H.

3. Traffic analysis for MPEG-2 video decoder

Our main observation is that, the traffic between different modules for a MPEG-2 decoder exhibits LRD. This is explained through the example of an MPEG-2 video decoder (Fig.3a) [9]. The decoder consists of the VLD (Variable Length Decoder), IQ (Inverse Quantization), IDCT (Inverse Discrete Cosine Transform), Motion Compensation (MC) units, and the associated buffers.

3.1 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]. 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.3a following the *Producer-Consumer* paradigm; that is, we describe the VLD process as the *Producer* and the IDCT/IQ unit as the *Consumer*. The VLD decodes the input stream, generates macroblocks, and

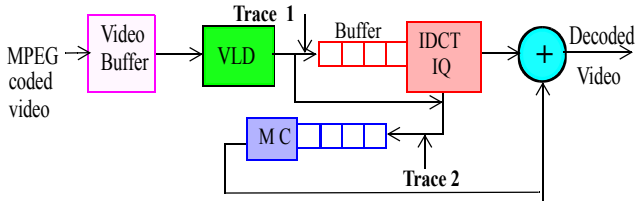

Fig.3a The block diagram of the MPEG-2 decoder

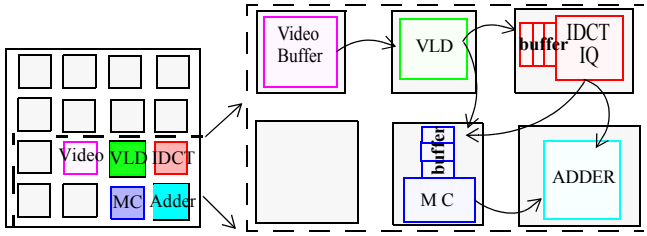

Fig.3b A possible mapping of the MPEG-2 decoder onto architecture in Fig. 1

puts them into the buffer. These are picked up by the *Consumer* to compute IDCTs and output data to reconstruct the frames. 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.)

Using the *Mpegstat* tool [13], we analyze a MPEG-2 video stream and find the detailed information about the macroblocks in the frames of a video. Depending upon whether a frame is I, P or B type, the macroblocks are processed differently and they follow different paths in the block diagram and then take different times 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.3a). The corresponding traces obtained were further evaluated using the variance-time method and the *R/S* method mentioned 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.

3.2 Results and discussion

Our approach to traffic modeling is "data driven". We rely upon four video sequences (*Clouds, Simpsons, Disc_ir, Hawaii*) of different video screen sizes ranging in length from 27 seconds (88000 macroblocks) to 1 second (43000 macroblocks). This represents all kinds of different scenes as shown in Table 1 by the statistics of I, P and B frames.

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

Table 1. Statistics for different clips

We focus on long sequences (X_i: i = 1,2,..., 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-2 video traffic traces between different modules of a MPEG-2 decoder.

The monitored trace file consists of two columns. The first one records 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. The actual traffic therefore consists of alternating sequence of macroblock arrivals and silence periods. We consider the discrete version of the process where the process is averaged within a window of size δ [8]. To compute H, we perform two tests[1]:

- The ***first test*** corresponds to the variance-time analysis of the time series X [2]. As an illustration, Fig.4 shows the plots corresponding to *Simpsons* video clip. We plot the least square fit line in the graph. The slope of the line gives the value of the parameter β, from which we find out the Hurst parameter as $H = 1 - \beta/2$.
- The ***second test*** corresponds to the R/S analysis of the time series [3]. The plots corresponding to the rescaled adjusted range statistics for *Simpsons* video clip are shown in Fig.5.

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
Clouds	0.7240	0.7646	0.7603	0.7639
Simpsons	0.6874	0.7432	0.7407	0.7943
Disc_ir	0.8108	0.8180	0.8421	0.8131
Hawaii	0.7238	0.7453	0.5455	0.6839

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

For convenience, a summary of the estimated Hurst parameters is also given in Table 2. As we can see, the values of H lie between 0.5 and 1 clearly indicating the presence of LRD. 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.

4. 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 LRD 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[2]. Using an excessive number of parameters is undesirable as it increases the uncertainty of the statistical inference and parameters are difficult to interpret. So we need to model these processes with parametrically parsimonious models (that is, using the minimal number of parameters) .

Norros in [7] used Fractional Brownian Motion (FBM) model which parsimoniously captures the 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:

$$P(Q > x) \sim \exp[-cx^{2-2H}] \quad (5)$$

1. We present the complete set of results for both tests in [31].
2. It will be as good as approximating a hyperbolically decaying function by a sum of exponentials.

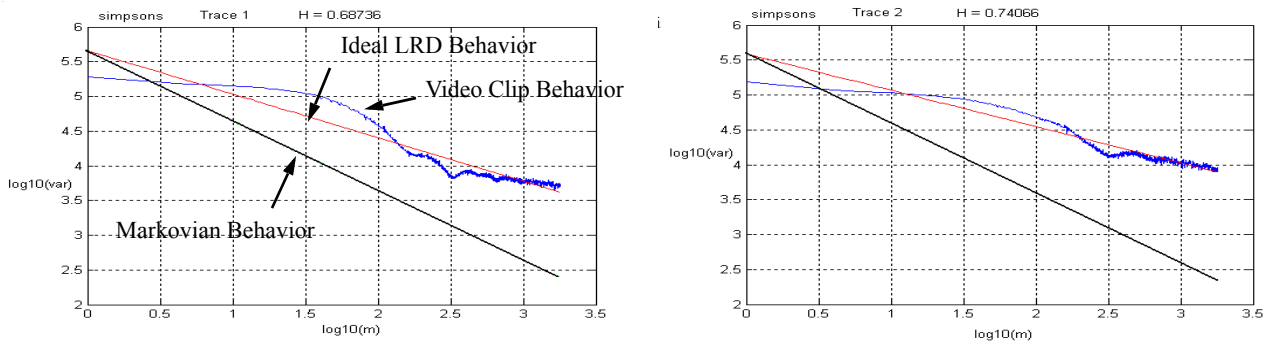

Fig.4 Variance-time plots for *Simpsons* at the IDCT module (Trace 1 in Fig.3a) and at the MC module (Trace 2 in Fig.3a).

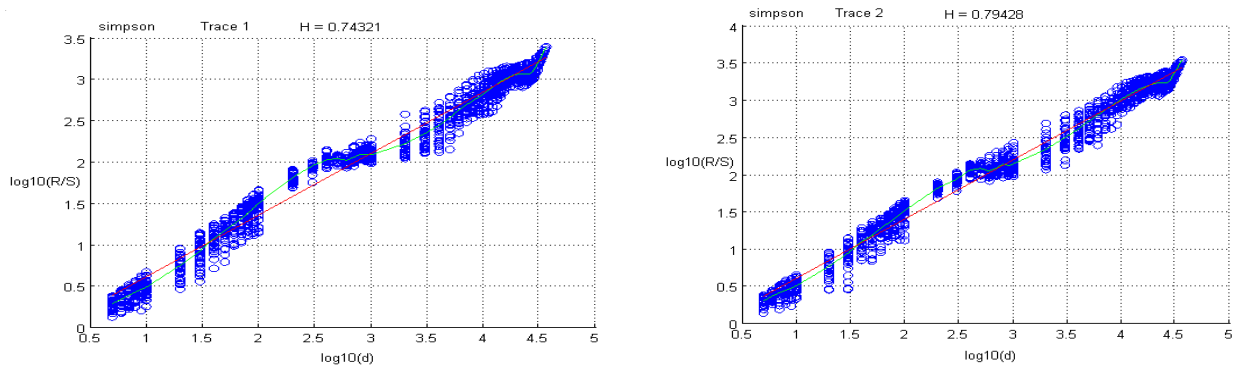

Fig.5 *R/S* plots for *simpsons* video clip at IDCT module (Trace 1 in Fig.3a) and at the MC module (Trace 2 in Fig.3a).

with

$$c = \frac{m^{2H-1}}{2a}\left(\frac{1-\rho}{\rho}\right)^{2H}\left[\left(\frac{1-H}{H}\right)^{H} + \left(\frac{H}{1-H}\right)^{1-H}\right]^{2} \qquad (6)$$

where: m is the mean input rate, ρ 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 can be obtained from plots like those in Figs.4-5.

To asses the *accuracy* and *impact* of our predictions on the overall performance of the on-chip network, the complementary buffer length distributions for two different video traces are shown in Fig.6. (The values of H and a from the Fig.4 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. That is the reason why simulation curves which represent just one instance of the stochastic process with that particular value of Hurst parameter lie close to the curve predicted by the deterministic formula and sometimes overestimate or underestimate the buffer length.

• Second, the dash-dot lines (obtained for H = 0.5) correspond to short-range dependent (SRD) models (like the 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 since it assumes the distribution of the arrival process to be short-range dependent i.e. exponential. This may cause severe performance degradation at chip-level.

• Third, we 'enriched' the *Disc_ir* trace by concatenating *Clouds* followed by *Simpsons* followed by *Disc_ir* and the second graph in Fig.6 corresponds to the simulation of this edited trace. Again, we can see a very good agreement between the buffer overflow prediction by the LRD formula (5) and the simulation values.

There is also another way of using the plots in Fig. 6 (and therefore eqn. (5)) 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-2 video decoder. More generally, a multimedia system designer may have a set of typical video clips (e.g. news reader, weather channel etc.) that are expected to run using a MPEG-2 decoder. We can find the Hurst parameter for each of these clips and use it to predict different buffer lengths corresponding to the same degree of lost bits. We can then choose the maximum predicted buffer length and have a theoretical support for assuring the QoS guarantee to the consumer.

Last but not least, compared to simulation-based buffer sizing, the LRD analysis framework offers a *fast* approach for buffer sizing. More precisely, for the simulation-based approach, if we want to asses 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. This is extremely time-consum-

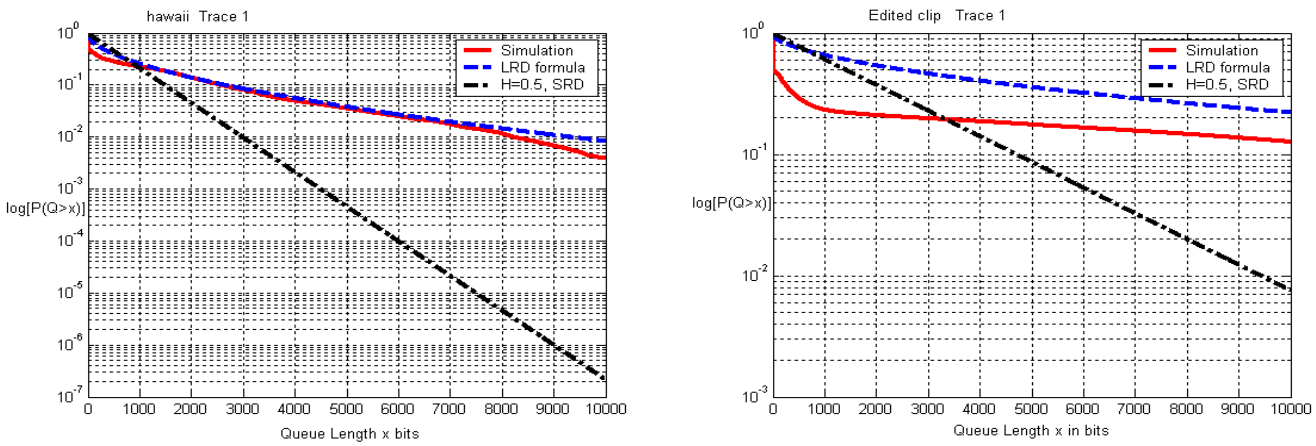

Fig.6 Complementary queue length distribution plots predicted by eqs. (5) and (6) for different video clips. The dashed curves indicate simulation results for an infinite queue with the arrival process following empirical trace (the server utilization is 0.5). The straight lines indicate the prediction by a SRD model (H = 0.5 in eqs.(5) and (6)).

ing since tens of hours of simulation may be needed. Conversely, 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 ρ in eq.(6) and get instantaneously the new buffer length.

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, 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. We plan to explore the application of traffic characterization for other systems in the video domain. We believe that our findings open up new directions of research with deep implications on fundamental issues in on-chip network design.

6. Acknowledgements: We would like to thank Alberto Sangiovanni-Vincentelli for stimulating discussions on this topic and anonymous reviewers for providing excellent feedback.

7. 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, et al. '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, 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, 1984.

[5] J. Beran, et al., '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,' in *IEEE/ACM Trans. on Networking,* Vol.4, No.2, April 1996.

[7] I. Norros, 'A storage model with self-similar input,' in *Queueing Systems* Vol. 16, 1994.

[8] P. Abry, D. Veitch, 'Wavelet Analysis of Long-Range Dependent Traffic,' in *IEEE Trans. on Info. Theory,* Dec. 1997.

[9] Sikora T., 'MPEG Digital Video Coding Standards,' in *IEEE Signal Processing Magazine,* Sept. 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, Jan. 2001.

[15] A. Kalavade, P. Moghe, 'A tool for performance estimation of networked Embedded End-Systems,' *Proc. DAC,* San Francisco, CA, June 1998.

[16] K. Keutzer et al., 'System-Level Design: Orthogonalization of Concerns and Platform-Based Design,' *IEEE Trans. on CAD,* Vol.19, No.12, Dec. 2000.

[17] S. Edwards, L. Lavagno, E. A. Lee, A. Sangiovanni-Vincentelli, 'Design of embedded systems: formal models, validation, and synthesis,' *Proc. IEEE*, Vol.85, No.3, March 1997.

[18] A. Mathur, A. Dasdan, R. Gupta, 'Rate Analysis for Embedded Systems,' in *Trans. on Design Automation of Electronic Systems*, Vol. 3, No. 3, July 1998.

[19] A. Nandi, R Marculescu, 'System-level Power/Performance Analysis for Embedded Systems Design,' in *Proc. DAC,* Las Vegas, NV, June 2001.

[20] A. Nandi, R. Marculescu, 'Probabilistic Application Modeling for System-Level Performance Analysis,' *Proc. DATE,* Munich, March 2001.

[21] K. Lahiri, A. Raghunathan, S. Dey, 'Fast Performance Analysis of Bus-based System-on-chip Communication Architecture,' *Proc. ICCAD,* Nov. 1999.

[22] M. Gasteir, M. Glesner, 'Bus-based Communication Synthesis on System Level,' in *Trans. Design Automation Electronic Systems*, Jan. 1999.

[23] T. Yen, W. Wolf, 'Communication Synthesis for Distributed Embedded Systems,' in *Proc. ICCAD*, Nov. 1995.

[24] J. A. Rowson and A. Sangiovanni-Vincentelli, 'Interface Based Design,' in *Proc. DAC.*, June 1997.

[25] J. Daveau, T. B. Ismail, A. A. Jerraya, 'Synthesis of System-level Communication by an Allocation Based Approach,' in *Proc. ISSS,* 1995.

[26] K. Park, W. Willinger, (Eds.), 'Self-Similar Network Traffic and Performance Evaluation,' J. Wiley and Sons, 2000.

[27] M. Sgroi, et al., 'Addressing the System-on-a-chip Interconnect Woes Through Communication-Based Design,' in *Proc. of DAC.*, Las Vegas, 2001.

[28] F. Karim, A. Nguyen, S. Dey, R. Rao, 'On-chip Communication Architecture for OC-768 Network Processors,' in *Proc. of DAC.*, Las Vegas, June 2001.

[29] V. Paxson, S. Floyd, 'Wide-Area Traffic: The Failure of Poisson Modeling,' *Proc. ACM SIGCOMM'94,* London, UK, 1994.

[30] G. Varatkar, R. Marculescu, 'Modeling and Synthesis of On-chip Multimedia Traffic', *Packet Video Workshop*, Pittsburgh, PA, April 2002.

[31] G. Varatkar, R. Marculescu, 'On-chip Traffic Analysis for Multimedia Applications', *Technical Report*, CMU CAD-01-25, Sept. 2001.