# Long-Range Dependence and On-chip Processor traffic

Antoine Scherrer, Antoine Fraboulet, Tanguy Risset

# Long-Range Dependence
# and On-chip Processor Traffic

Antoine Scherrer [b] Antoine Fraboulet [a] Tanguy Risset [a]

[a] *CITI - INSA Lyon*
*Bat 502, 20 avenue Albert Einstein,*
*F-69621, Villeurbanne, France*
*firstname.lastname@insa-lyon.fr*

[b] *Laboratoire de Physique - ENS Lyon*
*69364 Lyon Cedex 07, France*
*firstname.lastname@ens-lyon.fr*

**Abstract**

Long-range dependence is a property of stochastic processes that has an important impact on network performance, especially on the buffer usage in routers. We analyze the presence of long-range dependence in on-chip processor traffic and we study the impact of long-range dependence on networks-on-chip. We propose to investigate the presence of long-range dependence in communication traces of *processor* IP*s* at the *cycle-accurate level*. We also study the impact of long-range dependence on a real network-on-chip using the SocLib simulation environment and traffic generators of our own. Our experiments show that long-range dependence is not an ubiquitous property of on-chip processor traffic and that its impact on the network-on-chip is highly correlated with the low level communication protocol used.

*Key words:* Network on Chip, System on Chip, embedded software, long range dependence, network traffic

## 1 Introduction

The next generation of embedded systems on chip (SoC) will soon integrate many processor cores together with dedicated circuits connected via a network. This is particularly true for chips integrated in telecommunication and multimedia devices which require more and more computing power. These devices must be designed faster to meet the market demand which implies that more and more embedded software is present on these chips. Embedded software brings a faster and more flexible design process. However, as performance is

decreased compared to dedicated hardware, parallelization and efficient communication schemes must be implemented to reach the required computing power. In particular, controlling network throughput and contention is essential and is in general very difficult, because of non determinism introduced by parallelism. This is why network prototyping will become a major issue in future SoC design processes.

Recent advances in Internet traffic analysis have shown that predicting performance of a network is a difficult task. Stochastic modeling of the traffic must be used and it has been shown [24] that this modeling must take into account second order stochastic properties (covariance in addition to marginal law). More recently Varatkar and Marculescu [30] have shown that on-chip traffic might require a similar modeling. They have found long-range dependent behavior in communications between different parts of the MPEG-2 video decoding application. The presence of long-range dependence (LRD) in embedded applications, as in Internet traffic, could imply that IPs communication behavior will not be correctly modeled by a simple random traffic even if it is adjusted to a particular first order statistical law. However, the experiments of Varatkar and Marculescu were not done at cycle accurate level and it is still not clear how the LRD will impact the network-on-chip.

We want to emphasize the originality of our work with respect to the seminal paper of Varatkar and Marculescu [30], in which it was shown that the aggregated throughput of the macro-block arrival process at the IDCT/IQ module of a hardware MPEG-2 decoder exhibits long-range dependence. Standard queuing analysis with self-similar input was then used in order to evaluate the buffer usage. In the present work, we first investigate the presence of long-range dependence in the traffic produced by an on-chip *processor* IP at the *cycle accurate level* and then we show that LRD has a very different impact when the low level communication protocol uses non-split transactions. In practice, if the application is executed by a processor connected to a cache, the fact that each read transaction is waiting for an answer (this is called a *non-split* transaction), makes the impact of LRD negligible. If the application is executed by a dedicated IP that computes on data streams (i.e. not waiting for answers: *split transactions*), LRD decreases performance.

We also demonstrate here a way to adjust the size of the FIFOs given particular statistical properties of the executed application. As pointed out in [9], "buffers account for the main part of the NoC router area, it is a major concern to minimize the amount of buffering necessary under performance requirements". We quantify experimentally the impact of LRD on FIFO usage and on average communication latency and hence propose a way to evaluate the required size of FIFOs to cope with the presence of LRD. This is possible because we developed a complete NoC prototyping environment [26, 27] in which we are able to produce LRD traffic. The NoC used in the simulations is DSPIN [14], and

the simulation is done at cycle accurate level: routers are precisely simulated and instrumented in order to observe the exact behavior of what would occur on a real chip.

The paper is organized as follows: the next section presents the theoretical background needed to understand the notion of LRD, Section 3 presents our NoC prototyping environment. Experiments and results about the presence and the impact of LRD are presented in Section 4. Related works are referenced in Section 5.

## 2 Network traffic and long-range dependence

We start this section by introducing a simple on-chip traffic modeling formalism. We then give theoretical background on long-range dependence whereof the impact on performance of on-chip networks is the main topic of the paper.

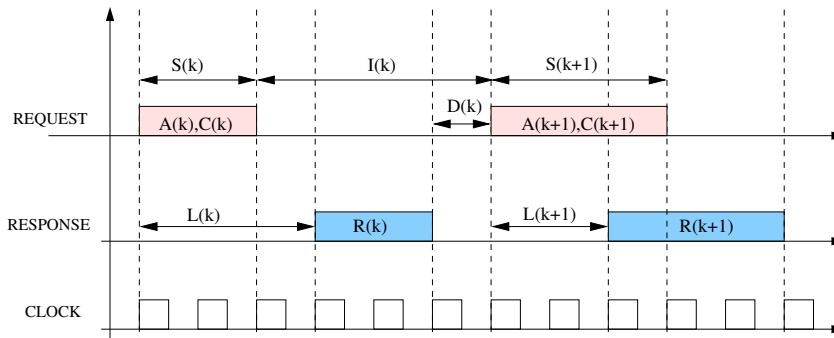### 2.1 On-chip traffic modeling



Fig. 1. Traffic modelling formalism

The traffic produced by a component is modeled as a sequence of transactions composed of *flits* (flow transfer units) corresponding to a bus-word. The $k^{th}$ transaction is a 5-uple $(A(k), C(k), S(k), D(k), I(k))$ meaning respectively, target address, command (read or write), size of transaction, delay and inter-request time. This is illustrated in Fig. 1. We also define the latency of the $k^{th}$ transaction $L(k)$ as the number of cycles between the start of the $k^{th}$ request and the start of the associated response. This is basically the round-trip time in the network and will be used in our experiments to evaluate contention. One can distinguish two main communication schemes used by IPs: the *non-split transactions* scheme where the IP is not able to send a

3

request until the response to the previous one has been received, and the *split transactions* scheme in which new requests can be sent without waiting for the responses. The non-split transaction scheme is used by processors and caches for instance (although for cache, it might depend on the cache parameters). The split transaction scheme is used by dedicated IPs performing computation of streams of data which are transmitted via dedicated *direct memory access* (DMA) modules.

Our goal is to emulate real traffic by means of traffic generators. With the above formalism, traffic generation means producing $(A(k), C(k), S(k), D(k), I(k))$, for each $k$. This can be done either deterministically (replay of a recorded trace), or using stochastic processes. In this work, we use stochastic traffic generators. Note that, in this case, we make the assumption that the elements of the transaction sequence are independent.

Let us recall that a stochastic process $X$ is a sequence of random variables $X[i]$ (we use brackets to denote random variables). We will consider two statistical characteristics of stochastic processes: the marginal law, which represents how the values taken by the process are distributed, and the covariance function, which gives information on the correlation between the random variables of the process as a function of the time lag between them. For instance, the sequence of delays $D(k)$ can be generated as the sample path of some stochastic process $\{D[i]\}_{i \in \mathbb{N}}$, with prescribed first and second statistical orders.

## 2.2 Long-range dependence

### 2.2.1 Definitions

Long-range dependence (LRD) is a property of a stochastic process which is defined as a slow decrease of its covariance function [23]. This function describes how the random variables of a process are correlated with each-other as a function of the time lag between these random variables. The covariance function $\gamma_X$ of a stochastic process $\{X[i]\}_{i \in \mathbb{N}}$ is defined as follows ($\mathbb{E}$ is the expectation):
$$\gamma_X(i, j) = \mathbb{E}(X[i]X[j]) - \mathbb{E}(X[i])\mathbb{E}(X[j])$$

If the process $X$ is wide-sense stationary, then its mean is constant ($\forall (i, j) \in \mathbb{N}^2, \mathbb{E}(X[i]) = \mathbb{E}(X[j]) \triangleq \mathbb{E}(X)$) and its covariance reduces to a one variable function as :

$$\forall (i, j) \in \mathbb{N}^2, \quad \gamma_X(i, j) = \gamma_X(0, |i - j|) \triangleq \gamma_X(|i - j|)$$

We expect this function to be decreasing, because correlated data are more

likely to be close (in time) to each other. However, if the process is long-range dependent, then the covariance decays very slowly and is not summable:

$$\sum_{k \in \mathbb{N}} \gamma_X(k) = \infty$$

Therefore, long-range dependence reflects the ability of the process to be highly correlated with its past, because even at large lags, the covariance function is not negligible. This property is also linked to self-similarity which is more general: in can be shown that asymptotic second order self-similarity implies long-range dependence [7].

LRD is an ubiquitous property of Internet traffic [18, 24], and it has also been attested on on-chip multimedia (MPEG-2) applications by Varatkar and Marculescu [30]. The main interest in LRD resides in its strong impact on network performance [23]. Above all, the needed memorization in the buffers is higher when the input traffic has this property [12]. As a consequence, for macro-networks as well as for on-chip networks, LRD should be taken into account if it is manifested in the traffic that the network will have to handle.

A long-range dependent process is usually modeled with a power-law decrease of the covariance function as follows :

$$\gamma_X(k) \underset{k \to +\infty}{\sim} ck^{-\alpha}$$

The exponent $\alpha$ (also called *scaling index*) provides a parameter to quantify the process' long-range dependency ($0 < \alpha \leq 1$). The Hurst exponent, denoted $H$, is the classical parameter for describing self-similarity [23]. Because of the analogy between LRD and self-similarity, it can be shown that a simple relation exists between $H$ and $\alpha$: $H = (2 - \alpha)/2$. As a consequence, $H$ ($1/2 < H < 1$) is the commonly used parameter for characterizing long-range dependence. Note that when $H = 0.5$, then there is no long-range dependence (this is also referred to as short-range dependence).

### 2.2.2   Estimation of the Hurst parameter

We have used a standard wavelet-based methodology for the estimation of the Hurst parameter [7] in our experiments. Let $\psi_{j,k}(t) = 2^{-j/2}\psi_0(2^{-j}t - k)$ denote an orthonormal wavelet basis, designed from the mother wavelet $\psi_0$. The $j$ index represents the *scale*: the larger $j$ is, the more the wavelet is dilated. The $k$ index is a shift in time.

For any $(j, k)$, $d_X(j, k) = \langle \psi_{j,k}, X \rangle$ are called the *wavelet coefficients* of the stochastic process $X$ ($\langle ., . \rangle$ is the inner product in the $L^2$ functional space).

These wavelet coefficients permit a study of the $X$ process at various times (values of $k$) and various scales (values of $j$).

When $X$ is a long-range dependent process with parameter $H$, the following limit behavior for the expectation of wavelet coeficients can be shown [7]:

$$\forall j, \quad \mathbb{E}(d_X(j,k)^2) \underset{j \to +\infty}{\sim} C2^{j(2H-1)} \tag{1}$$

It can also be shown that the time averages $S_j$ for each scale $j$ ($n_j$ is the number of wavelet coefficients available at scale $j$):

$$S_j = (1/n_j) \sum_{k=1}^{n_j} |d_X(j,k)|^2 \tag{2}$$

can be used as relevant, efficient and robust estimators for $\mathbb{E}(d_X(j,k)^2)$ [7]. From Eq. (1) and (2), the estimation of $H$ is as follows: *i)* plot $\log_2 S_j$ versus $\log_2 2^j = j$ and *ii)* perform a weighted linear regression in the limit of the coarsest scales (see for instance Fig. 2). These plots are commonly referred to as log-scale diagrams (LD) . In such diagrams, LRD is demonstrated by a straight line behavior in the limit of large scales. In particular, if the line is horizontal, then $H = 0.5$ and there is no long-range dependence.

To illustrate how we use this tool to evaluate the Hurst parameter, we provide in Fig. 2 a typical log-scale diagram extracted from an Internet trace [6]. Along the $x$ axis are the different values of the scale $j$ at which the process is observed. For each scale, $log_2 S_j$ is plotted together with its confidence interval (vertical bars). The Hurst parameter can be estimated if the different points plotted are aligned on a straight line for large scales.
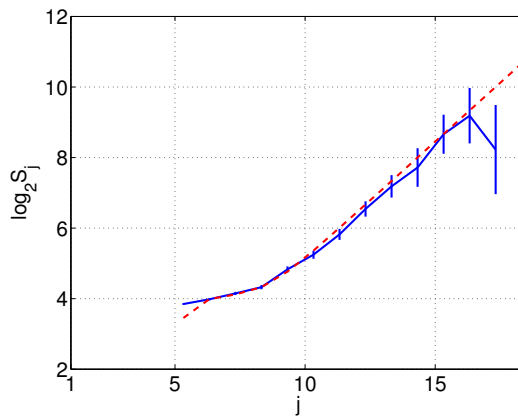


Fig. 2. Example of log-scale diagram (LD). The Hurst parameter is estimated by the slope of the dashed line (here, $H = 0.83$)

### 2.2.3 Synthesis of long-range dependent processes

The synthesis (generation of sample paths) of long-range dependent processes is easy if the marginal law is Gaussian [8]. The so-called *Fractional Gaussian Noise* (FGN) is commonly used for that. However, if one wants to generate a long-range dependent process whose marginal law is non-Gaussian, the problem is more complex. The inverse method [30] only guarantees an asymptotic behavior of the covariance function. We have developed, for several common laws (exponential, gamma, $\chi^2$, etc.), an exact method of synthesis described in [28]. We can thus produce synthetic long-range dependent sample paths that can be used, as it will be shown in Section 4, in order to evaluate the impact of LRD on the performance of an on-chip network. For instance, delay sequences are not likely to have a Gaussian distribution, but rather an exponential one as we expect many small delays and few big ones. Using our synthesis method, we can produce a synthetic exponential process with long-range dependence.

## 3  Multi-phase traffic generator environment

In the following, we present our synthesis flow for building multi-phase traffic generators that can be used to replace an IP in cycle-accurate NoC performance evaluation.
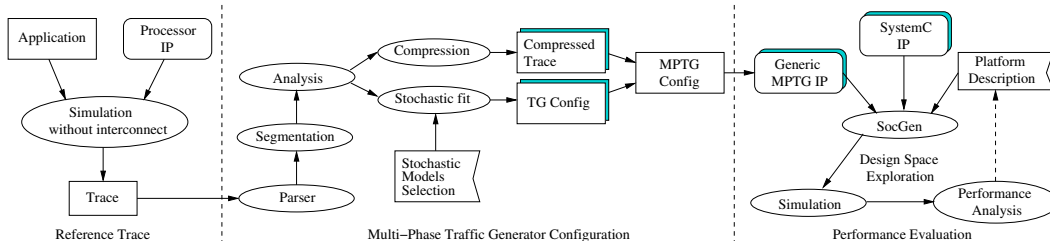


Fig. 3. Multi Phase Traffic Generator (MPTG) Framework: Traffic analysis and synthesis flow

### 3.1  SoC simulation environment

We use an open source, SystemC-based, cycle-accurate and bit-accurate simulation environment: SOCLIB [1]. The environment contains cycle-accurate models for various IPs. For instance, it contains a MIPS R3000 processor (with its associated data and instruction cache), on-chip memories, and a component used for displaying output (referred to as TTY). SocLib also includes cycle true simulators of the networks on chip developed at the LIP6 laboratory: SPIN and DSPIN [14].

The application running on the MIPS is composed, in addition to bootstrapping information, of the C program cross-compiled with GCC to a MIPS target. In our experiments, we use several embedded programs from the multimedia domain (still image, video and audio decoding). We make the assumption that contention only delays the communications, without changing their ordering. This is realistic for most applications and networks-on-chip.

The global simulation flow is depicted in Fig. 3. First, we generate a reference trace by running the processor IP to be emulated. This trace is obtained with an ideal network environment (no network contention), hence capturing the intrinsic communication behavior of the application. Next, we process the trace in our traffic analysis and synthesis tool explained hereafter and we obtain configuration files for our traffic generators. A generic traffic generator, further referred to as MPTG, has been written once for all.
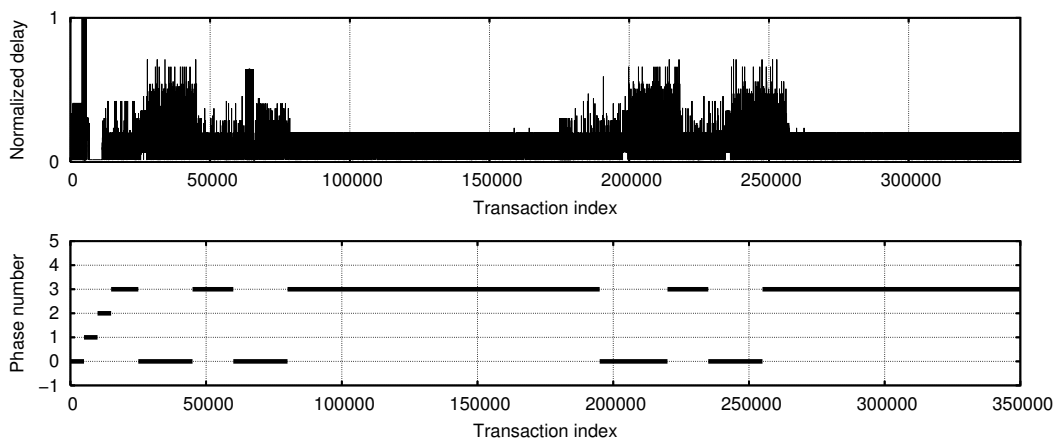


Fig. 4. Segmentation of the MP3 aggregated throughput into four phases

## 3.2 Traffic analysis

The first step of the analysis is the *segmentation*. It consists in identifying phases in the trace with a regular behavior. We have developed an automatic procedure for that based on existing work on CPU simulation. This part of the work is detailed in [26]. An example of phases detected for the MP3 application is shown in Fig. 4.

On each of the identified phases, each element of a transaction: $A(k)$, $C(k)$, $S(k)$, $D(k)$ and $I(k)$ (see 2.1), is independently analyzed with the following procedure:

8

- If the designer chooses deterministic traffic generation, each transaction 5-uple is recorded in a file and compressed using the BZ2 program, which results on average in dividing the size of the trace by 70. The resulting trace can be directly replayed by our MPTG.
- If the designer chooses stochastic traffic generation, a statistical analysis is performed by an automatic fitting procedure that adjusts the first and second statistical orders. The designer must choose which model they want to use (with or without LRD for instance) before the analysis can take place.

  The probability distribution function (first statistical order) can be either fitted to some classical distributions (Gaussian, Exponential, Gamma, Log-Normal, etc.) or kept as it is (the model is then the probability of appearance of each value of the process). The fit is done using Maximum Likelihood Expectation and a $\chi^2$ goodness-of-fit test is used to compare and evaluate all different solutions [16].

  The covariance function (second statistical order) can be fitted to the one of an ARMA [1] process (short range correlations only), a FGN [2] (long-range-dependence only) process, or a FARIMA [3] process (both short and long-range correlations). We use a wavelet-based estimation of the Hurst parameter [7] widely adopted in the network traffic analysis domain (see section 2.2).

At the end of this step, a configuration file is generated. It contains, for each phase, all needed information for the traffic synthesis.

### 3.3  Traffic synthesis

The deterministic traffic generation is simply done by reading the compressed traffic trace.

For stochastic traffic generation purposes, we have developed an independent random number generator that can produce realizations of a wide variety of processes including non-Gaussian long-range-dependent ones (see Section 2.2.3). This generator is integrated in the MPTG and the analysis simply produces the adequate MPTG configuration file.

The platform designer then describes the desired platform architecture (e.g. as the one presented in Fig. 12) and uses a perl script (referred to as `SocGen` in Fig. 3) generating all files needed for the simulation.

At simulation time, transactions are generated by MPTG according to a phase description file and a sequencer is in charge of switching between phases. At the

---

[1]  Auto-Regressive Moving Average
[2]  Fractional Gaussian Noise
[3]  Fractionnaly Intergarted ARMA

end of the simulation, performance analysis indicates whether any parameters of the platform must be changed or not.

### 3.4   Traffic generator key features

As a summary, we highlight the main characteristics of our MPTG:

- Our MPTG takes into account network properties (latency, contention, etc.). This means that the MPTG will offer a good emulation of the processor independently of the network it is connected to. Hence, we can re-use the same configuration file for many different platforms.
- The global methodology presents as little manual editing as possible (e.g. the complete simulation platform is generated). Moreover, the traffic analysis and synthesis flow can very easily be plugged into another synthesis environment: only the traffic generator IP needs to be adapted.
- Our MPTG is able to run processes having a wide variety of parameters: deterministic replay, various stochastic models with prescribed first and second order statistics. To our knowledge this feature is not present in any other cycle accurate simulators.
- Our MPTG is multi-phase and hence can take into account the non stationarity of the traffic.

With all these features, the designer has a very flexible tool for the design space exploration of NoC. The accuracy of the generated traffic with respect to the original trace it is suppose to emulate has been demonstrated in [27]. On any metric the mean error is less than 4%.

## 4   Experiments

In this section, we investigate the presence of long-range dependence in on-chip processor traffic, and we present experimental results about the impact of long-range dependence on network-on-chip performance. In particular, we show how to use this information in order to evaluate the size of FIFOs in routers.

### 4.1   Presence of long-range dependence

In order to check for LRD in on-chip traffic, we have run several software implementations of applications from the multimedia domain:

- Mpeg-2: we use `libmpeg2`, an open-source implementation of the Mpeg-2 video decoding standard [5].
- Mp3: we use `libmad`, an open-source implementation of the Mp3 audio decoding standard [4].
- Jpeg: we use two Jpeg still image decoding standards. The first one, known as M-Jpeg is a multi-thread implementation and the second one, known as S-Jpeg is a single thread implementation [2].
- Jpeg-2000: we use `libj2k`, an open-source implementation of the Jpeg-2000 image decoding standard [3].

The input data for each application are described in Tab. 1.
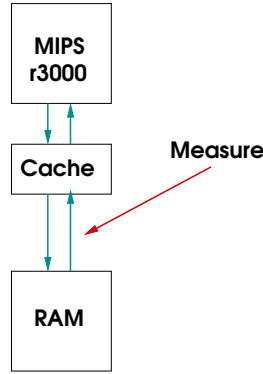


Fig. 5. The simulation platform for LRD detection

We use a simple platform (see Fig. 5) in order to truly characterize the traffic of the triplet (implementation/processor/cache). As explained in Section 3, our goal is to replace this triplet by a traffic generator. If we study communication on a more complex platform (the one in Fig. 12 for instance), the traffic of an IP is influenced by communications between other IPs and by network-on-chip configuration (topology, routing protocol, etc.). Therefore, each simulation is first done on a simple platform including a Mips r3000 processor (associated with instruction and data cache), directly connected to a memory holding all necessary data represented in Fig. 5.

| App. | Input |
|---|---|
| Mpeg-2 | 2 images from a clip (176×144 color pixels) |
| Mp3 | 2 frames from a sound (44,1 kHz, 128 kbps) |
| M-Jpeg | "Lena" picture (256x256) |
| S-Jpeg | "Lena" picture (256x256) |
| Jpeg-2000 | "Lena" picture (256x256) |

Table 1
Inputs used in the simulations

The traffic trace is recorded at the network's interface of the cache. A communication trace is extracted from the VCD (Value Change Dump) trace file, as explained in Section 3. We study the *aggregated throughput*, computed as the number of flits sent in consecutive time-windows of size 100 cycles. Aggregated throughput is interesting because it combines delay and size of transactions. For the sake of clarity, we present results on this quantity alone. Note that LRD is present neither in commands nor in target addresses time series.

In order to estimate the covariance function and investigate the presence of LRD in these traffic traces, we have used the wavelet-based estimator [7] described in Section 2.2. Each log-scale diagram (Fig. 7-11) includes the estimated covariance (full line) and the linear regression for the estimation of the Hurst parameter (dotted line). The estimated value for this parameter, denoted $\hat{H}$ is reported in the caption.

As stated in Section 3, traffic phases can be identified, and one should pay attention, when checking for long-range dependence in a traffic trace, to consider stationary parts only. Indeed, covariance estimation tools can produce misleading results when used on non-stationary time series. Therefore, we have extracted stationary parts of the traffic traces based on information provided by our phase determination software [26]. This is shown in Fig. 6. In each part we compute LD diagrams and estimations of the Hurst parameter as shown in Fig. 7-11. Below are the comments for each application.

- MPEG-2 (see Fig. 7). The shape of the LD does not exhibit evidence of LRD. Indeed the estimated value for the Hurst parameter, 0.56, means no LRD is present in the trace. In this case, an IID (Independent Identically Distributed) process is a good approximation of the traffic. One can note a peak around scale $2^5$, meaning that a recurrent operation with this periodicity is present in the algorithm which might have an impact on network contention. Such behavior could be captured by an ARMA process for instance [10].
- MP3 (see Fig. 8). As for the MPEG-2 implementation, no trace of LRD can be found in this case: the estimated Hurst parameter is close to 0.5. The other parts of the communication trace do not present any LRD either.
- M-JPEG (see Fig. 9). For this simulation, a linear behavior can be observed in the range of scales $[2^4\ 2^{14}]$. On top of this behavior, a peak is present around scale $2^{11}$. The behavior at larger time scales does not exhibit long-range dependence, but scale $2^{14}$ corresponds to approximately 1.6 million cycles, which means LRD should be taken into account. For the joint modeling of LRD and small-scale behavior, FARIMA processes can be used.
- S-JPEG (see Fig. 10). Firstly, It is interesting to note that the shape of the LD is very different from the M-JPEG implementation, showing that the statistical properties observed are implementation rather than algorithm dependent. For the S-JPEG implementation, long-range dependence can be observed in the limit of large scales ($[2^{10}\ 2^{13}]$). As for other applications,
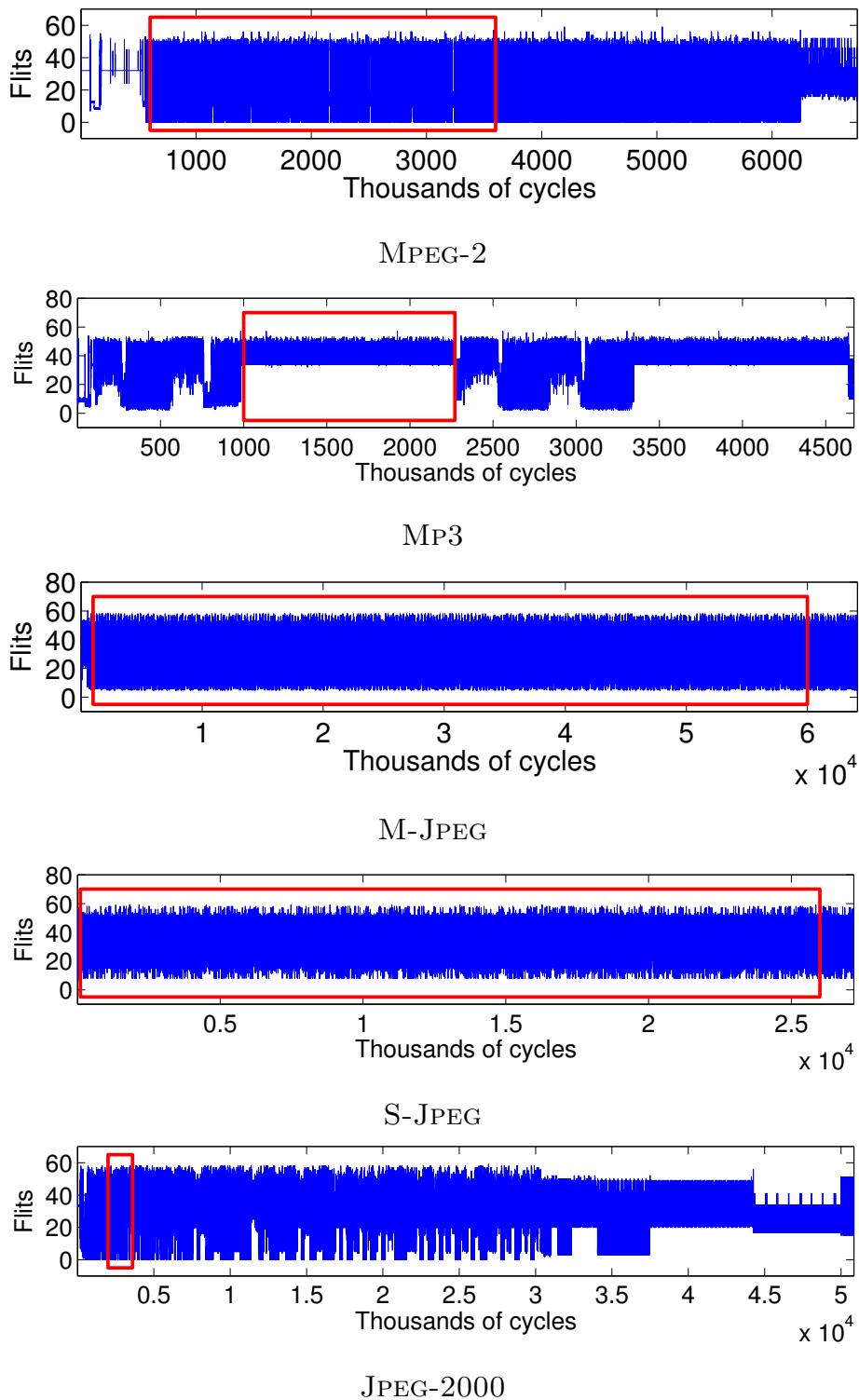
Fig. 6. Aggregated throughput traces obtained from the execution of each implementation. The boxes mark the part of the trace on which the LD curves were obtained.

13

peaks are present in LD.

- JPEG-2000 (see Fig. 11). On this application, the traffic trace exhibit a strong non-stationarity, so that the trace must be split in rather short parts for the analysis. In some of these parts, corresponding specifically to the *Tier-1* entropic decoder of the JPEG-2000 algorithm, LRD is present with an estimated Hurst parameter value between 0.85 and 0.92 (depending on the parts). In the other parts of the algorithm, no long-range dependence could be evidenced.
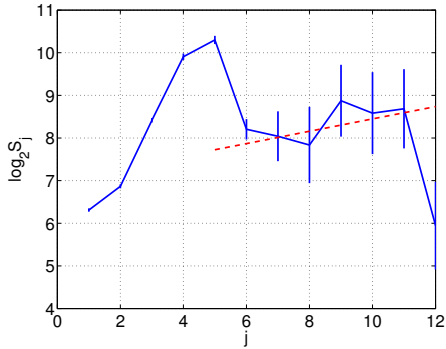


Fig. 7. LD of the traffic trace corresponding to the MPEG-2 implementation. $\hat{H} = 0.56$
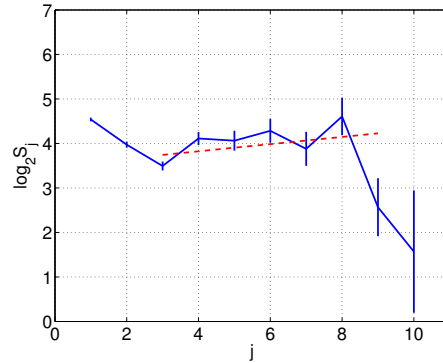


Fig. 8. LD of the traffic trace corresponding to the MP3 implementation. $\hat{H} = 0.52$
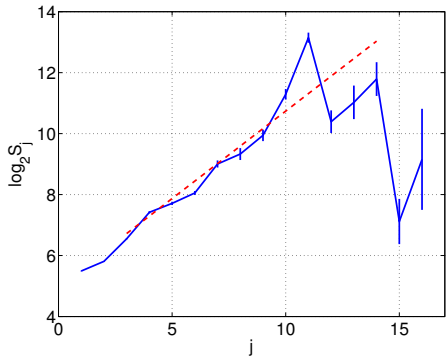


Fig. 9. LD of the traffic trace corresponding to the M-JPEG implementation. $\hat{H} = 0.77$
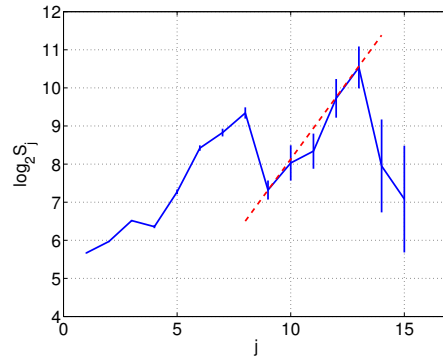


Fig. 10. LD of the traffic trace corresponding to the S-JPEG implementation. $\hat{H} = 0.95$

We can conclude from these experiments that long-range dependence is not an ubiquitous property of the traffic produced by a processor associated with a cache executing a multimedia application. In some parts of the algorithms, LRD is present, however it is combined with periodicity effects which may have an equivalent impact on the network-on-chip performance.
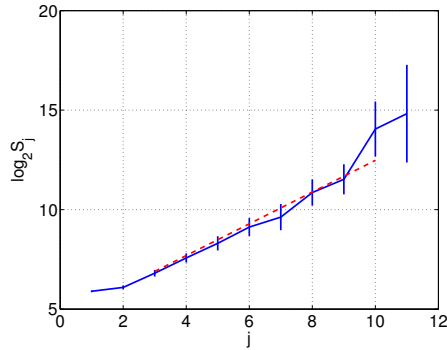
14

Fig. 11. LD of the traffic trace corresponding to the Jpeg-2000 implementation. $\hat{H} = 0.89$

## 4.2  Impact of long-range dependence

In this section we study the impact of long-range dependence on FIFO usage and communication latency in a network-on-chip.
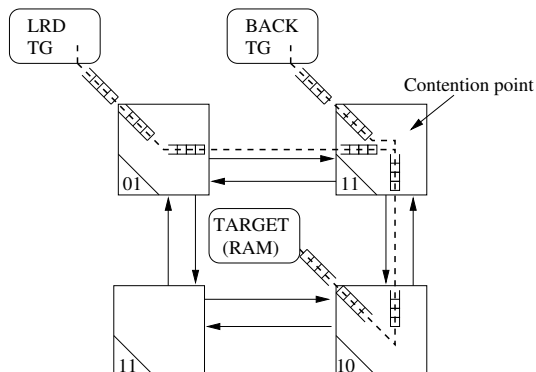
### 4.2.1  Experimental Setup



Fig. 12.  Platform used for evaluating the impact of LRD. Data paths along the switches of the NoC are represented as dash lines, FIFOs used by communications are represented inside the switches.

All the simulations are done on the platform shown in figure 12 in the SocLib simulation environment [21]. The network on chip is DSPIN developed at the Lip6 laboratory as an evolution of the SPIN network [14]. It uses a mesh topology, static $XY$ routing and wormhole memorization in the switches.

The LRD TG component is a traffic generator that produces a traffic in which the delay or inter-request time between transactions has an exponential marginal distribution and a long-range dependence of parameter $H$. Sizes of transactions are IID exponential random variables, with mean 10 flits (it is the typical value observed in the simulation of multimedia applications). The BACK TG

15

traffic generator is used to introduce contention in the network. It injects, at a constant rate, a traffic composed of constant-sized transactions of 10 flits. The load introduced by this component is expressed as a percentage of the maximum load (1 transaction every cycle). The BACK TG component uses split transactions in order to truly inject a constant rate in the network. All transactions of both traffic generators are addressed to the TARGET component: a RAM (with random read or write commands) which answers in one cycle (ideal memory). The simulation length has been fixed to 10000 transactions of the LRD TG.

The major goal of this work is to be able to quantify the size of the buffers present in the switches and the wrappers (components inserted between IPs and switches) as a function of the amount of LRD present in the traffic. For that, we have recorded the evolution of the usage of all FIFOS along the communication path used by traffic generators, as shown in figure 12. We have also computed the latency of the communications adding the latency of all FIFOS. We have run two types of simulations:

- FIXED: in these simulations, the FIFO size in the switches has been fixed to 2 flits in order to study the usage of the wrappers FIFOs. Indeed, with such small-sized FIFOS in the switches, the contention is directly transmitted to the wrappers which will be in charge of adjusting the traffic of the IP to the availability of the network.
- INF: in these simulations, the FIFO size in the switches and in the wrappers has been fixed to a very high (non-realistic) value of 250 flits so that the usage of all FIFOS (switches and wrappers) can be studied.

In both types of simulation, we have used two different communication schemes:

- NO-SPLIT: in this configuration the LRD TG does not use split transaction. It waits for the reception of the $k^{\text{th}}$ transaction before attempting to send the $(k+1)^{\text{th}}$. The generated delay is therefore the time between the reception of the last response and the start of the new request. This communication scheme is used by peripherals and some processors with caches.
- SPLIT: in this configuration the LRD TG uses split-transaction, that is to say the responses do not influence the sending of the requests. Here, the generated delay is the time between requests. This communication scheme is representative of hardware accelerators with direct memory accesses.

A particular simulation will therefore be referred to as a couple $(T,S)$, where $T \in \{\text{FIXED, INF}\}$ and $S \in \{\text{NO-SPLIT, SPLIT }\}$. For instance the simulation (FIXED, SPLIT) corresponds to small FIFOS in the switches and split-transaction communication scheme. The SoC platforms are approximately simulated at 100 000 cycles per second.

16

### 4.2.2  Results

Each figure contains five curves that correspond to different values of the Hurst parameter ($H$) of the LRD TG. The vertical axis is always a performance metric. It might be, for instance, the average latency over the whole execution or the usage (maximum or mean over the execution) of a particular FIFO. The horizontal axis corresponds to the load introduced by the BACK TG (in percentage of maximum load).



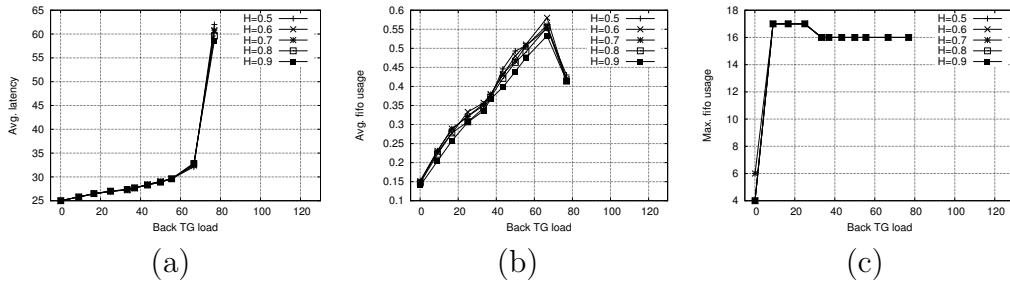(a)                          (b)                          (c)

Fig. 13. Simulations FIXED and NO-SPLIT: average latency and usage of the LRD TG wrapper FIFO. (a: Latency, b: LRD TG wrapper FIFO (Mean), c: LRD TG wrapper FIFO (Max))



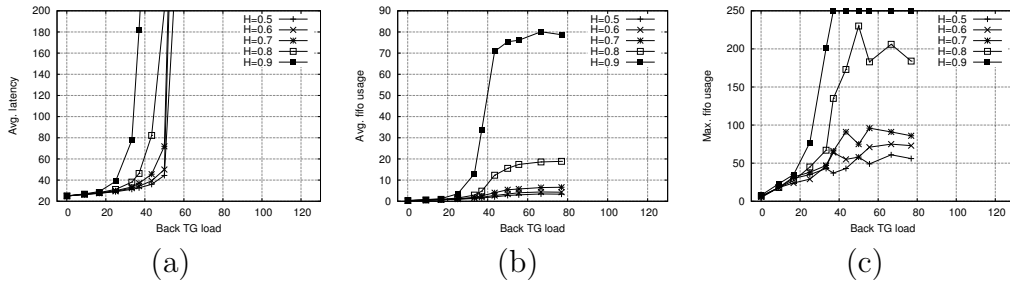(a)                          (b)                          (c)

Fig. 14. Simulations FIXED and SPLIT: same measures as in Fig. 13 but using split transactions. (a: Latency, b: LRD TG wrapper FIFO (Mean), c: LRD TG wrapper FIFO (Max))

The impact of LRD in the case of FIXED is evidenced by the difference between Fig. 13 and 14. Recall that for these simulations, all the buffering occurs in the wrapper FIFOs because the sizes of FIFOs in switches are fixed to 2 flits. Fig. 13 clearly shows that LRD has no impact on performance in the case of non-split transactions because the five curves representing different values of $H$ are identical (similar results occurs when observing the FIFO of the BACK TG wrapper). This is typically what would appear if a processor IP (with cache) were running an application. Conversely, in the case of split transactions, the results of Fig. 14 confirm the results of [30]: the LRD has a strong impact on performance and must be taken into account to correctly prototype NoC performance. The greater the $H$ parameter is, the sooner network contention occurs.

The second set of figures (Fig. 15 and 16) shows how we evaluate the size of the FIFOs needed in the switches. This must be linked with a static analysis

17

of the path taken by the data. Fig. 12 highlights the FIFOs used by the data flowing from the IPs to the target. We may for instance deduce that the North input FIFO of router 10 will never be filled with more than 2 flits because all data coming into router 10 comes from a single source (router 11). This is confirmed in Fig. 16-(c). In these (INF,SPLIT) experiments (Fig. 15 and 16) the size of the FIFOs in switches is fixed to a very large value (250 flits) to evaluate the maximum and usage of the FIFOs. The contention of the network occurs later than in the (FIXED,SPLIT) experiments (comparing Fig. 15-(a) and Fig. 14-(a)) because there are more buffers on the data path. Note that the average latency on Fig. 15-(a) does not diverge except for $H = 0.9$. The latency diverges if the throughput given to communication is less than the injection rate of that communication. Recall that in these simulations, only the BACK TG component injection rate varies. The average load introduced by the LRD TG component is constant. Moreover, the round robin arbitration scheme in the router guarantees a minimum throughput to the communications of the LRD TG. Only if the FIFOs are small (see for instance Fig. 14-(a)), or if the injection rate of BACK TG is sufficiently high, will the latency diverge. In this particular simulation (INF,NO-SPLIT), the average input rate of the LRD TG is not large enough to make the latency diverge. However it is interesting to note that when $H$ is close to 1 the latency diverge as if the average rate of LRD TG was much bigger.

Looking at Fig. 15 and 16, the designer can fix the sizes of different FIFOs. For instance, if the LRD TG IP has $H = 0.8$ and if the BACK TG IP emits at a rate of 40% the maximum rate, then we can see by looking at Fig. 16-(a) that the input West FIFO of router 11 should have a size of 100 flits if we do not want any contention to appear (for instance, if hard real time is required). If the real time constraints are soft, the designer can decide to look at the mean usage of the FIFOs rather than the maximum usage. To have a more precise evaluation of the impact of setting this FIFO size, the designer can also re-run the simulation by fixing the size of router 11 input West FIFO to 50 for instance and check the impact on performance.
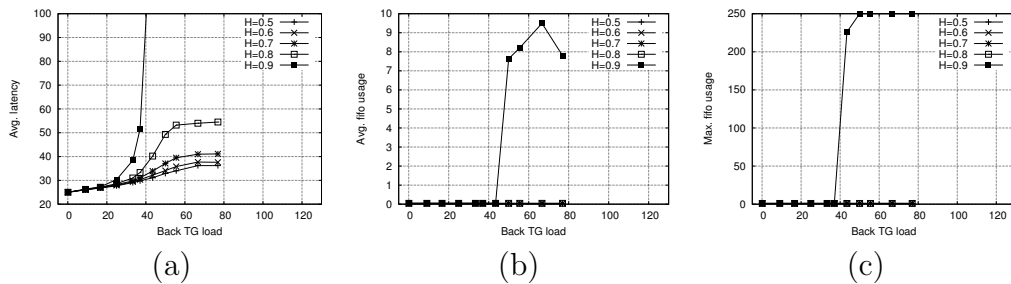


Fig. 15. Simulations INF and SPLIT: Maximum usage of the first three FIFOs on the path from LRD TG to TARGET (a: Latency, b: LRD TG wrapper FIFO, c: router 10 East out FIFO)

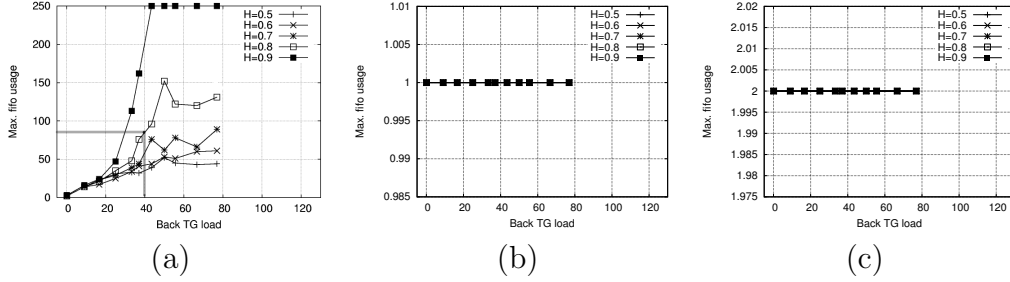As a summary of these experiments: LRD is more likely to have an impact on

Fig. 16. Simulations INF and SPLIT: Maximum usage of the last three FIFOs on the path from LRD TG to TARGET: contention occurs in router 11 (input West FIFO). (a: router 11 West in FIFO, b: router 11 South out FIFO, c: router 10 North in FIFO)

NoC performance when split-transactions communication scheme is used by IPs. Another original result presented here is a practical way to compute FIFO usage and to precisely fix the size of the NoC FIFOs in order to reach a certain level of performance.

## 5   Related work

A recent and up to date review on works in the topic of networks-on-chip is done by Bjerregaard and Mahadevan in [9]. In this survey, traffic characterization for NoC performance evaluation is highlighted as a key issue. More specifically related to on-chip traffic generation, two main approaches have been studied: deterministic traffic generation, in which the objective is to exactly reproduce the traffic of a given IP [13, 19, 20] and stochastic traffic generation which uses random sources in place of real IPs [17, 25, 31]. In the deterministic approach, Mahadevan *et al.* introduced, for instance, a trace compiler able to accurately reproduce the traffic of a processor [20]. As networks on chip are still in early stages of development, the main part of their performance evaluation is done using a random source. In [17], stochastic processes are used for generating transaction sizes and transaction delay using several statistical laws (Poisson, Exponential, and Normal). However, none of these works proposes a *fitting procedure* to determine the adequate statistical models that should be used to simulate a given traffic. Thid *et al.* use self-similar traffic for on-chip prototyping in a recent work [29]. The work of Varatkar and Marculescu about long-range dependence in on-chip traffic is referenced in Section 2.2.

Among the works addressing the FIFO sizing, *Hu et al.* introduced a non-uniform NoC buffer space allocation algorithm [15]. Their algorithm was evaluated under various random traffic patterns as well as exponentially distributed traffic fitted to the communications of real applications. Chandra *et al.* proposed a methodology to size the FIFOs along a communication path [11]. They

19

identified the various parameters that impact FIFO sizing, among which the production rate, consumption rate and data burst size are the most important. Ogras *et al.* also pointed out in [22] the fact that the NoC buffer sizing problem suffers from critical issues such as *realistic traffic patterns*. In this work, we have presented our proposition for having more realistic traffic patterns as well as an application of this to the buffer sizing problem.

## 6  Conclusion

We have studied both the presence and the impact of long-range dependence in the scope of networks-on-chip. Our results show firstly that long-range dependence is not an ubiquitous property of the traffic produced by on-chip processors running multimedia applications. Secondly, using cycle accurate level simulation of a complete SoC, we have shown that the low level protocol used by IPs to communicate has an important impact on the possible performance degradation due to the presence of long-range dependence. In the case of non-split transaction, LRD is likely to have a very small impact on performance while in the case of split transaction, LRD greatly affects performance. We have also shown how the NoC designer can use our traffic generation environment to optimize the NoC for a particular SoC platform.

## References

[1]  Soclib simulation environment, On line: `http://soclib.lip6.fr/` (2005).

[2]  Independent jpeg group, On line: `http://ijg.org/` (2006).

[3]  J2000 - a jpeg2000 codec, On line: `http://j2000.almacom.com/` (2006).

[4]  libmad - mpeg audio decoder library, On line: `http://www.underbit.com/products/mad/` (2006).

[5]  The libmpeg2 library page, On line: `http://libmpeg2.sourceforge.net/` (2006).

[6]  Wand network research group, On line: `http://wand.cs.waikato.ac.nz/wand/wits/` (2006).

[7]  P. Abry, D. Veitch, Wavelet analysis of long-range dependent traffic, IEEE Transactions on Information Theory 44 (1) (1998) 2–15.

[8]  J. Bardet, G. Lang, G. Oppenheim, A. Philippe, M. Taqqu, Long-Range Dependence: Theory and Applications, chap. Generators of long-range dependent processes: A survey, Birkhäuser, 2003, pp. 579–623.

[9] T. Bjerregaard, S. Mahadevan, A survey of research and practices of network-on-chip, ACM Comput. Surv. 38 (1) (2006) 1.

[10] P. Brockwell, R. Davis, Time Series: Theory and Methods, 2nd edn., Springer Series in Statistics, New York, 1991.

[11] V. Chandra, H. Xu, A. abd Schmit, L. Pileggi, An interconnect channel design methodology for high perforance integrated circuits, in: date, 2004.

[12] A. Erramilli, O. Narayan, W. Willinger, Experimental queueing analysis with long-range dependent packet traffic, ACM/IEEE transactions on Networking 4 (2) (1996) 209–223.

[13] N. Genko, D. Atienza, G. D. Micheli, J. M. Mendias, R. Hermida, F. Catthoor, A complete network-on-chip emulation framework, in: DATE 05, 2005.

[14] A. Greiner, P. Guerrier, A generic architecture for on-chip paquets-switched interconnections, in: Design, Automation and Test in Europe, 2000.

[15] J. Hu, R. Marculescu, ApplicationSpecific Buffer Space Allocation for NetworksonChip Router Design, in: IEEE/ACM Intl. Conf. on Computer Aided Design, San Jose, CA, 2004.

[16] R. Jain, The Art of Computer Systems Performance Analysis, John Wiley and Sons, Inc., 1991.

[17] K. Lahiri, A. Raghunathan, S. Dey, Evaluation of the traffic performance characteristics of system-on-chip communication architectures, in: Int. Conf. VLSI Design, 2001.
URL `citeseer.nj.nec.com/lahiri01evaluation.html`

[18] W. E. Leland, M. S. Taqqu, W. Willinger, D. V. Wilson, On the self-similar nature of ethernet traffic (extended version), ACM/IEEE transactions on Networking 2 (1) (1994) 1–15.

[19] M. Loghi, F. Angiolini, D. Bertozzi, L. Benini, R. Zafalon, Analyzing on-chip communication in a mpsoc environment, in: DATE 04, 2004.

[20] S. Mahadevan, F. Angiolini, M. Storgaard, R. G. Olsen, J. Sparsø, J. Madsen, A network traffic generator model for fast network-on-chip simulation., in: DATE 05, 2005.

[21] C. S. L. of Paris VI, Soclib simulation environment, On line: `http://soclib.lip6.fr/` (2006).

[22] U. Y. Ogras, J. Hu, R. Marculescu, Key research problems in noc design: A holistic perspective, in: Intl. Conf. on Hardware/Software Codesign and System Synthesis, 2005.

[23] K. Park, W. Willinger (eds.), Self-Similar Network Traffic and Performance Evaluation, John Wiley & Sons, 2000.

[24] V. Paxon, S. Floyd, Wide-area traffic: The failure of poisson modeling, ACM/IEEE transactions on Networking 3 (3) (1995) 226–244.

[25] S. G. Pestana, E. Rijpkema, A. Rădulescu, K. Goossens, O. P. Gangwal, Cost-performance trade-offs in networks on chip: A simulation-based approach, in: DATE 04, 2004.

[26] A. Scherrer, A. Fraboulet, T. Risset, Automatic phase detection for stochastic on-chip traffic generation, in: CODES+ISSS, Seoul, south Korea, 2006.

[27] A. Scherrer, A. Fraboulet, T. Risset, Generic multi-phase on-chip traffic generator, in: ASAP, Steamboat Springs, USA, 2006.

[28] A. Scherrer, N. Larrieu, P. Borgnat, P. Owezarski, P. Abry, Non gaussian and long memory statistical characterisations for internet traffic with anomalies, IEEE Transactions on Dependable and Secure Computing (TDSC)To appear.

[29] R. Thid, K. Sander, A. Jantsch, Flexible bus and noc performance analysis with configurable synthetic workloads, in: 9th Euromicro Conference on Digital System Design (DSD 2006), 2006.

[30] G. Varatkar, R. Marculescu, On-chip traffic modeling and synthesis for mpeg-2 video applications., IEEE Transactions on Very Large Scale Integration (VLSI) Systems 12 (1) (2004) 108–119.

[31] D. Wiklund, S. Sathe, D. Liu, Network on chip simulations for benchmarking., in: IWSOC, 2004.