QuaLe: A Quantum-Leap Inspired Model for Non-Stationary Analysis of NoC Traffic in Chip Multi-Processors

Paul Bogdan, Miray Kas, Radu Marculescu and Onur Mutlu
Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213-3890, USA
{pbogdan, mkas, radum, omutlu}@ece.cmu.edu

***Abstract* — This paper identifies non-stationary effects in grid like Network-on-Chip (NoC) traffic and proposes *QuaLe*, a novel statistical physics-inspired model, that can account for non-stationarity observed in packet arrival processes. Using a wide set of real application traces, we demonstrate the need for a multi-fractal approach and analyze various packet arrival properties accordingly. As a case study, we show the benefits of our multi-fractal approach in estimating the probability of missing deadlines in packet scheduling for chip multiprocessors (CMPs).**

***Keywords*: Chip Multi-Processors, Networks-on-Chip, Self-Similar Stochastic Processes, Multi-fractal Analysis.**

I. Introduction

Traditionally, on-chip communication used a bus-based or point-to-point communication infrastructure. Given the lack of scalability in these approaches, Networks-on-Chip (NoC) emerged as a promising solution to on-chip communication [7]. General purpose CMPs with NoC-based communication are typically implemented in a tile-based structure where each tile consists of a processing element (PE), private/shared cache banks and a router [8].

On-chip networks resemble traditional data networks as the switches, routers and the packet-based communication constitute the basic elements of both types of networks. However, on-chip networks differ from general computer networks in many aspects, most notably in terms of optimizations needed to satisfy various performance, power, and area constraints. Nevertheless, the need for an in-depth understanding of the network traffic is unavoidably common to all networks as it is the key for optimized network design.

Previous research includes several attempts to analyze and model the traffic behavior observed in different network types such as local area networks (LAN), wide area networks (WAN) [30] and the Internet (WWW) [6]. These papers focus on identifying self-similarity in network traces, leaving the issue of a more general multi-fractal model open. Since NoC design is a relatively new research area, the need for a multi-fractal traffic approach is yet unaddressed.

In this paper, we propose a statistical physics-inspired model which is able to capture the statistical characteristics of NoC traffic patterns accurately and explain the transition of the NoC traffic from mono to multifractal behavior. More precisely, in our approach, we consider each buffer in the NoC architecture as being characterized by a fitness distribution (with or without time dependency) based on whether or not the changes in the NoC traffic occur as a function of the intrinsic variability exhibited by the target applications or due to the fluctuations in the number of applications executing on the NoC at any given time.

Based on the statistical features of the fitness distribution, we demonstrate that the NoC traffic can exhibit either a monofractal or multifractal behavior. Also, in contrast to traditional network designers that frequently use synthetic traffic patterns (*e.g.*, uniform-random, immediate neighbor, tornado, or hotspot) for testing their designs, in this paper, we analyze network behavior using traces from a wide set of real applications and eliminate the inaccuracies observed in statistical network traffic models due to the use of artificial traffic patterns.

In summary, our main contributions are as follows:

- We propose *QuaLe*, a novel quantum-leap-inspired model for characterizing packet arrival processes which can capture the multifractal characteristics of NoC traffic in CMP platforms. Similar to *Quale*, defined as the universal property of an object which is independent from the object itself, our quantum-leap model introduces the fitness concept as a tool to encompass the universal multifractal behavior that is observed in NoC traffic due to various interactions among various traffic flows.
- We illustrate the existence of non-stationary effects in NoC traffic and quantify the degree of multifractality when running various real world applications. We also estimate the probability of missing real-time deadlines and thereby discuss the implications of the multifractal approach in packet scheduling for CMPs.

The rest of the paper is organized as follows: Section II provides a brief background on the theoretical concepts we use throughout the paper. Section III reviews the scientific approaches proposed for self-similar traffic. In Section IV, we present *QuaLe,* a novel statistical physics-inspired model which is able to account for the multifractal nature of NoC traffic. In Section V, we present our experimental methodology. In Section VI, we present an in-depth analysis of the main theoretical findings and in Section VII, we discuss their implications in CMP design, pointing out some future research directions. We conclude the paper by summarizing our main contributions in Section VIII.

II. Background on Self-Similarity

Self-similarity is a real-life concept that has been observed in many natural phenomena [36]. An object or data is called self-similar if it is still similar to the shape of the entire object/data when we zoom in at different scales.

To better understand the self-similar (or fractal) nature of some objects, we show two examples in Figure 1: Figure 1.a shows that as we zoom in over a 3D Euclidean object, the final object starts to deviate significantly from the initial one. In contrast, Figure 1.b shows a 3D perspective of a mountainous area displaying self-similarity as we zoom in across its surface.

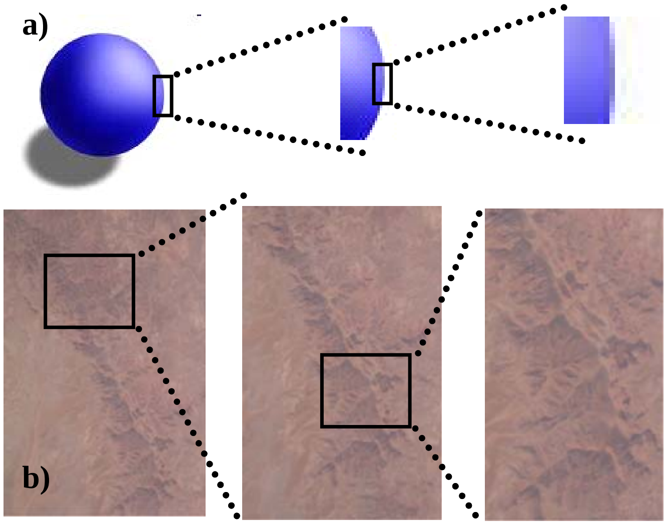

Figure 1. a) An iterated zoom in process over a 3D sphere showing that Euclidean objects do not display self-similar characteristics. b) An iterated zoom in process over a mountainous area which displays a self-similar behavior in valley branching.

Apart from self-similar characteristics observed in various geometrical objects, self-similarity can be also conceived in a temporal manner. For instance, some data are considered to be self-similar in time, if the time series preserves its temporal properties with respect to scaling in time.

Along the same lines, the self-similarity property of a stochastic process $X(t)$ implies that its *distribution* over two non-overlapping time intervals (*i.e.*, $X(t)$ and $X(bt)$, for any real $b > 0$) remains the same up to a scaling factor (*i.e*, $P(X(bt) < x) = P(b^{H}X(t) < x), b > 0, H \in R$, where $0.5 < H < 1$ is called the Hurst parameter). In addition, a self-similar stochastic process is called long-range dependent (LRD), if its autocorrelation function decays as a power law $R(k) \sim k^{2H-2}$ [20]. Moreover, if the H exponent varies in time, the stochastic process is called *multifractal*.

III. Related Work and Novel Contribution

The concept of self-similarity dates back to the early efforts of A. N. Kolmogorov to explain the chaotic nature of turbulence [16]. More precisely, in that paper, Kolmogorov proposes a mathematical formalism relating the self-similarity and small scale statistics of turbulent flows to the energy dissipation and universal scaling laws. Later on, several attempts were made to bridge more closely the theory with real measurements [12]; this includes the non-stationary aspects of turbulence via random cascade models [24][18][11][2][28]. We should also note that Mandelbrot constructed a mathematical formalism of roughness, and introduced the concept of fractal to denote the geometric scale-inference [19]. Over the years, self-similarity and multifractal formalisms have found application in many other fields such as diffusion-limited aggregation [32], dielectric breakdown [1], biological systems [33].

More recently, self-similarity and fractal approaches have been employed to study the structure of complex networks [9][23] or elucidate the departure of experimental measurements in various information networks from the standard Poisson assumption of packet arrival time distribution [26][15][10]. Other experimental studies identify the existence of self similar behavior in World Wide Web, local and wide area networks [6][30].

Informally, self-similarity in network traffic can be perceived as statistical similarity observed in bursty patterns of network traffic over a wide range of time-scales. This corresponds to scale-invariant burstiness [27][29][31] or mono-fractal behavior in network traffic, where the level of burstiness is typically captured via a single (Hurst) parameter. The most significant impact of self-similar behavior is the existence of long-range dependence (LRD), or long-range memory effects.

In the NoC context, applications are mapped to the available network resources. This leads to interactions and contention at various network resources. Analyzing the actual characteristics of the network traffic is therefore a primary concern for optimization purposes. Indeed, one of the major challenges in constructing an accurate performance model for network analysis is the presence of non-stationary effects in traffic behavior. This is because operating the network close to (or at) criticality can only make the non-stationarity become more pronounced and the analysis more difficult. To date, there exist many experimental studies that demonstrate the existence of LRD, self-similarity, and even multifractality to some extent in various traffic traces [6][30]. However, proposing a model that is able to *explain* many features of the network traffic while being intrinsically related to the dynamics of NoCs remains an open problem.

Towards this end, we present a statistical physics model for NoC traffic characterization based on fitness distributions which is able to retrieve the mono-fractal and multi-fractal behaviors as particular cases. We test the validity of the theoretical conclusions by investigating the multifractal features of several application configurations running on a CMP platform where communication happens via the NoC approach.

IV. *QuaLe*: A Quantum Leap Inspired Traffic Model for NoCs

To investigate the temporal characteristics of NoC traffic, we adopt the quantum approach proposed in [4][5] and assume that, at any point in time, each buffer is characterized by a fitness function *E*. More precisely, we establish an analogy between thermodynamic systems and communication networks such that the number of packets stored in a buffer corresponds to the number of particles on a certain energy level. Consequently, the routing of packets (which naturally affect the number of arrivals at a certain buffer in the network) becomes similar to the migration of particles (*i.e.*, quantum leap) among different energy levels.

Within this quantum approach, the number of particles is determined by a fitness function *E*. Similarly, we assume that this fitness function determines the number of arrivals at a certain buffer. To study the temporal behavior of packet arrivals, we denote by $a_i(t)$ and $P_i(a, t|E)$ the cumulative number of arrivals at buffer *i* by time *t* and the probability that buffer *i*, characterized by fitness *E* received *a* packets by time *t*, respectively.

As a first step in analyzing the multifractality of the NoC traffic, we investigate the statistical properties of the $P_i(a, t|E)$ distribution of packet arrivals at any buffer *i* (*i.e.*, $i = 1 \div NB$, where *NB* is the number of buffers in the network) by using a master equation as follows:

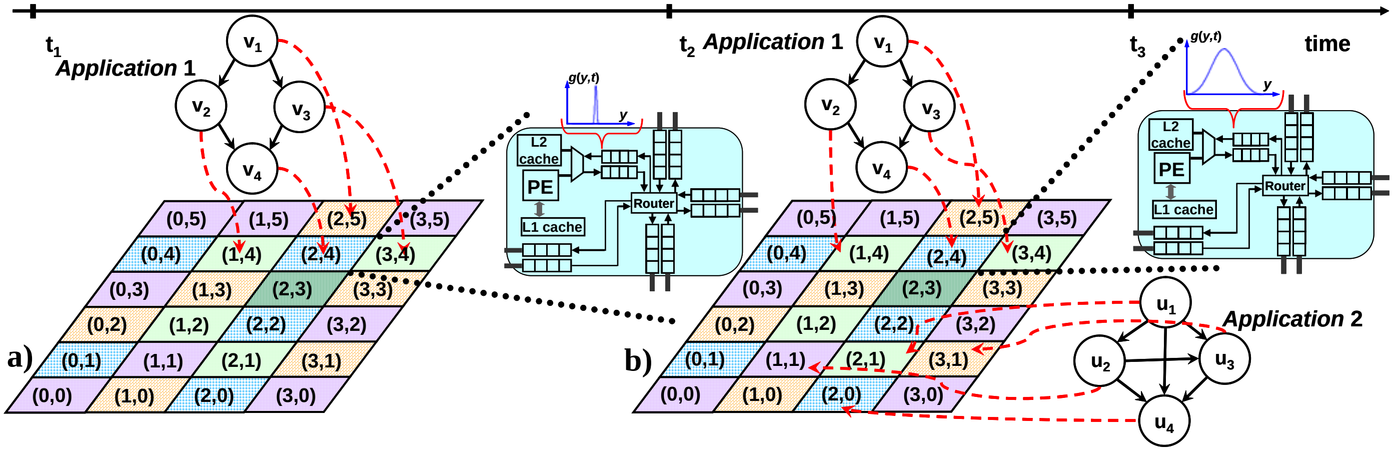

Figure 2. Schematic representation showing: a) An application task graph mapped at time t_1 onto a 4×6 NoC architecture and the distribution g(y,t) of scaling exponents associated with the input buffer of the PE located at (2,4). b) At time t_2, a new application task graph is mapped onto the same architecture. Due to the changes in the overall traffic patterns and the increase in the communication load, the distribution g(y,t) associated with the input buffer of the local PE located at (2,4) can become very different compared to case a).

$$\frac{\partial[tP_i(a,t|E)]}{\partial t}+\frac{\partial}{\partial a}\int\frac{aP_i(ya,t|E)}{A(t)y}g(y,t)dy = 0 \quad (1)$$

In Equation 1, $A(t)$ represents the total number of packets in the network up to time t, y is the scaling exponent which depends on the fitness E as described in [5] and $g(y, t)$ denotes the distribution of scaling exponents y in packet arrivals. Note that $g(y, t)$ changes as a function of traffic patterns at time t.

Equation 1 involves two main components. The first term states that the probability distribution $P_i(a,t|E)$ is proportional with time. Therefore, this term captures the long-term memory effects of the arrival process. The second term encompasses the statistical properties of the arrival process $a_i(t)$ at buffer i as a function of changes in the traffic patterns. For instance, as shown in Figure 2.a, one can assume that between t_1 and t_2 only Application 1 is running in the system and thus the distribution $g(y, t)$ is skewed around a single or few fractal exponents. In contrast, when a new application enters into the system, depending on the network region where it is mapped to, the traffic pattern and the communication load may vary drastically. This situation has also a significant effect on the $g(y, t)$ distribution associated with each buffer, causing it to become either more or less skewed around some particular values (see the change in the $g(y, t)$ distribution of the input buffer at location (2,4) in Figure 2.b due to Application 2 entering the system). Simply speaking, the second term of Equation 1 states that the distribution of a new self-similar (*i.e.*, re-scaled) stochastic process can be obtained from an initial distribution via a scaling relationship [20].

By substituting the k-th order moment of the number of arrivals at node i $M_k(t) = \int a^k P_i(a,t|E)da$ into Equation 1, we obtain its time dependence as follows:

$$M_k(t)\approx t^{\tau(k)}, \quad \tau(k) = \frac{k}{\bar{A}}\int\frac{g(y,t)}{y^{k+1}}dy-1 \quad (2)$$

where $\bar{A}$ denotes the maximum number of packets injected by time t and the nonlinear exponents $\tau(k)$ represent a multifractal signature. It should be noted that when there are no changes in the network traffic pattern and the fitness distribution $g_i(y,t)$ associated with buffer i obeys the relation $g_i(y,t)\approx y_0^k\delta(y-y_0)$, Equation 2 characterizes a mono-fractal stochastic process.

Based on the nonlinear exponents $\tau(k)$, the multifractal spectrum can be expressed as follows:

$$f(\alpha) = min_k[\alpha k-\tau(k)] \quad (3)$$

where α represents the fractal dimension. Different from mono-fractal processes, the multifractal spectrum defined in Equation 3 states that a stochastic process can be characterized by multiple fractal dimensions α and their normalized weights $f(\alpha)$.

Generally speaking, if the multifractal spectrum $f(\alpha)$ is large around a certain fractal dimension α, then there are many points characterized by this fractal dimension. In other words, the multifractal spectrum plays the role of a probability distribution function for the scaling exponents characterizing a stochastic process. Moreover, if the support of fractal dimensions α is wide, then we can state that the stochastic process is characterized by many α exponents. As such, a multifractal spectrum is more likely to characterize the NoC traffic of CMP platforms as the complexity of the application and the distribution $g(y, t)$ increases.

V. Experimental Methodology

The experimental results are obtained using an in-house cycle-accurate x86 NoC based CMP simulator. The front end of the simulator depends on Pin [17] and iDNA [3]. The high-level view of the simulated architecture is presented in Figure 2. In this architecture, the NoC routers are virtual channel (VC) buffered 2D-mesh routers with 5 physical ports: one for each {North, South, East, West} direction and extra one for the local core the router is attached to.

The NoC traffic is dominated by the communication between PEs and shared L2 cache banks. Each PE has a private L1 cache. When an application cannot find the data in its private L1 cache, an address packet is created and sent to the L2 cache bank the requested data resides in. When the requested data becomes available in the L2 cache, a data packet is injected into the network as a reply to the received request.

We model a static non-uniform cache architecture (S-NUCA) where the L2 cache bank the cache line resides in is determined via the lower order bits in the address of the cache line. Therefore, depending on the requested data address, an address packet might be destined to any of the shared cache banks in the network. In addition, we model our PEs to be self-throttling, thus preventing a PE from injecting new packets into the network when its injection buffers are full. Table 1 lists the major system parameters.

Table 1: Baseline processor, cache, memory and network configurations used in the experimental setup.

Processing Element pipeline	2 GHz Processing Element, 128-entry instruction window, 12-stage pipeline
Fetch / Exec / Commit width	3 instructions per cycle in each core; only 1 can be a memory operation
L1 caches	Private, per-PE, 4-way set associative, 128B block size
L2 caches	Shared 1MB bank per PE, 16-way set associative, 128B block size, XOR based address-to-bank mapping
Network router	Buffered, wormhole switched, XY routing, Virtual channel (VC) flow control, 4 VCs per port, Round-Robin packet scheduling, 4 flit buffer depth, 1 flit per Address Packet, 4 or 8 flits per data packet
Network topology	10x10 2D-mesh, each tile has a router, PE, private L1 cache, shared L2 cache bank

We model the PEs and the caches as in real systems.Thus, we do not have a direct control over the packet injection rate of real applications. L1 misses result in the creation of new address packets, which are then sent to the shared L2 caches to request data. Therefore, a smaller L1 cache size results in a higher number of misses and implicitly a higher packet injection rate, whereas a larger cache results in a lower injection rate. To discuss the impact of changing the injection rate over the distributions of inter-arrival times, we report in Section VI statistical information using different L1 cache sizes.

Table 2: Classification of SPEC 2006 applications.

Set	Application	Description	Inj. Rate (Packets/ Cycle)
SET-I	400.perlbench	Perl scripting language	0.001147
	401.bzip2	File compressor	0.039072
	403.gcc	C Language optimizing compiler	0.052649
	445.gobmk	Go playing program	0.025125
	450.soplex	Simplex Linear Program solver	0.077675
	454.calculix	3D Finite Element code	0.005126
	482.sphinx	Speech recognition	0.049308
SET-II	429.mcf	Single-depot vehicle scheduler	0.196724
	462.libquantum	Quantum computer simulation	0.096904
	464.h264ref	Video compression program	0.012494
	433.milc	Quantum Chromodynamics	0.047728
	437.leslie3d	Computational fluid dynamics	0.096399
	447.dealII	Adaptive finite elements	0.007556
SET-III	456.hmmer	A gene sequence database search	0.044859
	458.sjeng	Chess & variants playing games	0.015726
	471.omnetpp	Ethernet network simulator	0.043674
	473.astar	2D path finding library	0.016980
	435.gromacs	Simulator for Lysosome protein	0.015665
	444.namd	Biomolecular systems simulator	0.065465

For our simulation experiments, we use a subset of SPEC 2006 applications [14]. Each benchmark is compiled using gcc 4.1.2 with -O3 optimizations and a representative execution phase is chosen using PinPoints [25]. Our experiments involve two different scenarios simulated on a 10x10 mesh NoC:

- **Single-application scenarios**: In this set of simulations, out of 100 PEs, only one PE executes an application. Depending on the requested memory address, address (request) packets may hit on any L2 bank in the system. We also vary the data packet size (*i.e.*, 1 flit, 4 flits and 8 flits) and L1 cache size (*i.e.*, 8K, 16K, 32K, 64K).
- **Dynamic multiple-application scenarios**: In this set of simulations, all 100 PEs are utilized. We randomly divide the applications into three groups (see Table 2). The applications in the first group are executed during the entire run. The applications in the second group are dynamically replaced with the applications in the third group. We perform experiments with dynamic multiple-applications scenario on a 10×10 mesh NoC using two different L1 cache sizes (*i.e*., 64K and 8K), and two different routing algorithms (XY wormhole routing and deflection wormhole routing [22]) with 8-flit long data packets.

VI. Experimental Results and Analysis

A. Impact of Workload on Traffic Statistics

One way to elucidate the presence of temporal scaling in the NoC traffic involves computing the power spectrum of the inter-arrival times of data packets because the power spectrum of a time series measures the magnitude of variability and the degree of scaling as a function of frequency. To be more precise, a time series exhibits a *scaling behavior* over a certain frequency interval $[f_1, f_2]$ if the power spectrum obeys a power law relation $E(f) \sim f^{-\beta}$, where $0 < \beta < 2$ is the scaling exponent. In other words, the slopes observed in the power spectrum show that the fluctuations in variance are scale-invariant to certain frequency bands. Moreover, stochastic processes exhibiting a nonlinear power spectrum are called *non-stationary* [20][33]. Apart from mathematical intricacies, the power spectrum of inter-arrival times can show if the NoC traffic is non-stationary, *i.e*., whether or not the properties of the applications traffic patterns change as time progresses.

Figure 3.a, Figure 3.b, and Figure 3.c show the power spectrum of benchmark 400.perlbench (on the log-log scale) while running on a 10×10 mesh NoC with various L1 cache sizes (*e.g*., 8K, 16K, 32K, 64K) and for several data packet sizes (1 flit (a), 4 flits (b) and 8 flits (c)). Figure 3.a shows that by reducing the L1 cache size from 64K to 8K, the non-stationary effects become more pronounced. For instance, for 1-flit long data packets, the slope of the power spectrum in Figure 3.a increases from 0.09257 to 1.07. This can be regarded as a non-stationarity signature since a stationary stochastic process would have a slope close to zero (*i.e.*, it would look like an horizontal line). Similar trends can be observed in Figure 3.b and Figure 3.c for data packet sizes of 4 and 8 flits where the power spectrum slopes change in the range of [0.084, 1.072] and [0.09232, 1.075], respectively. This shows that the impact of packet size on the network traffic statistics is minimal, especially when compared against the influence of packet injection rate.

The impact of the increased packet injection rate over non-stationarity is also confirmed by power spectrum of data from several other applications: 403.gcc (Figure 3.d), 444.namd

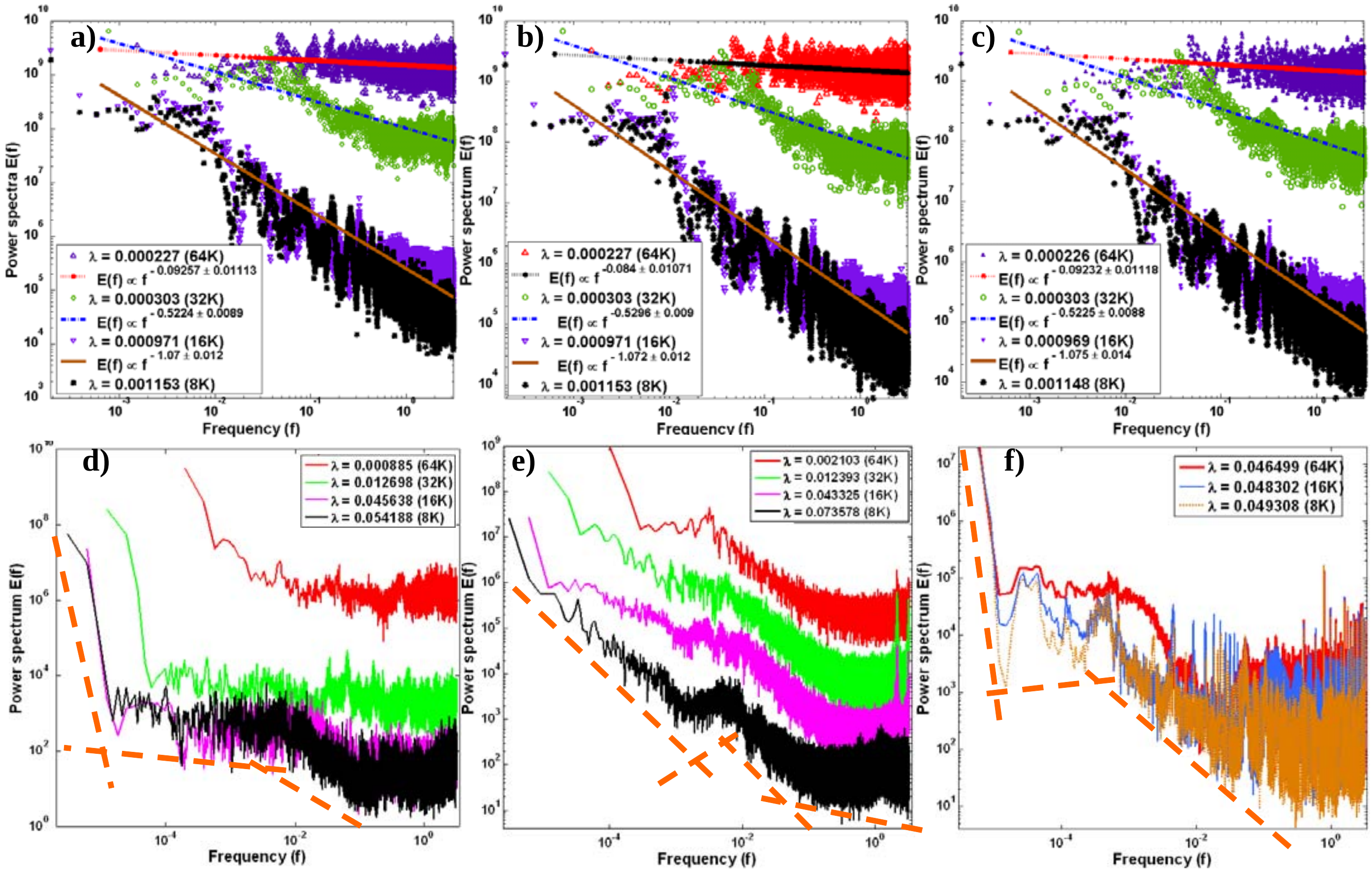

Figure 3. Power spectrum of inter-arrival times of data packets for different applications running on a 10×10 NoC with various L1 cache sizes: 400.perlbench for three different packet sizes: 1 flit (a), 4 flits (b) and 8 flits (c). Power spectrum of the inter-arrival times of data packets for three applications: 403.gcc (d), 444.namd (e) and482.sphinx3 (f) running on a 10×10 NoC with various L1 cache sizes and 4-flit long data packets.

(Figure 3.e), and 482.sphinx3 (Figure 3.f). In contrast to previously discussed applications, these plots show a more complex behavior which require nonlinear fits (see the dashed lines) and display multiple scale breaks. This kind of more complex nonlinear behavior observed in the power spectra of these applications shows not only the existence of non-stationarity, but also a more complex behavior than a monofractal one.

Considering the data presented in Figure 3, it can be stated that the power spectrum exhibits a wide range of scaling in time. These observations should not only raise awareness to non-stationarity effects, but also suggest that fractional Brownian motion [20] and other similar approaches are *not* necessarily adequate for modeling real NoC traces as they rely on a single scaling exponent characterized by a single fractal dimension. Instead, the richness of scaling displayed by various traces supports the existence of multiple fractal dimensions. We discuss this issue later in the paper.

B. Multiscale Analysis of NoC Traffic

Rather than complicating the traffic characterization problem, the multifractal approach basically reduces the statistics of a long times series to a distribution of scaling exponents (*i.e.*, Equation 3) which encompasses the non-stationary aspects as well [13]. Therefore, we investigate next the presence of multifractality in NoC traffic.

Single application scenarios: Figure 4 reports the multifractal spectrum (see Equation 2) of the inter-arrival times of the data packets for three applications from single application scenarios (400.perlbench (a), 403.gcc(b), and 401.bzip2 (c)), running on a 10×10 mesh NoC with various L1 cache sizes (*i.e.*, 8K, 16K, 32K, 64K).

For all these applications and data packet sizes, the support of the multifractal spectrum shrinks with the increasing packet injection rate. However, this does not imply, the existence of a monofractal behavior because for a mono-fractal process, the spectrum would appear as a delta function (*i.e.*, as a very narrow spike) centered around a certain value on the *x*-axis. In other words, for a mono-fractal stochastic process, if the time series of inter-arrival times between data packets were segmented into disjoint sets, each newly created time segment would be characterized by the same fractal dimension. On the other hand, for a multi-fractal stochastic process, each of the newly created time segment can have its own fractal dimension based on the particular characteristics of the time dependent generated network traffic. From this discussion, it can be concluded that the network evolves toward a congested state as the multifractal spectrum shrinks.

We should also note that the 403.gcc and 437.leslie3d applications exhibit opposite behaviors with increasing packet injection rate, especially in terms of persistence tendency. The persistence tendency means that the stochastic process is characterized by some kind of periodicity reflected via higher correlation moments; this implies that the process exhibits higher

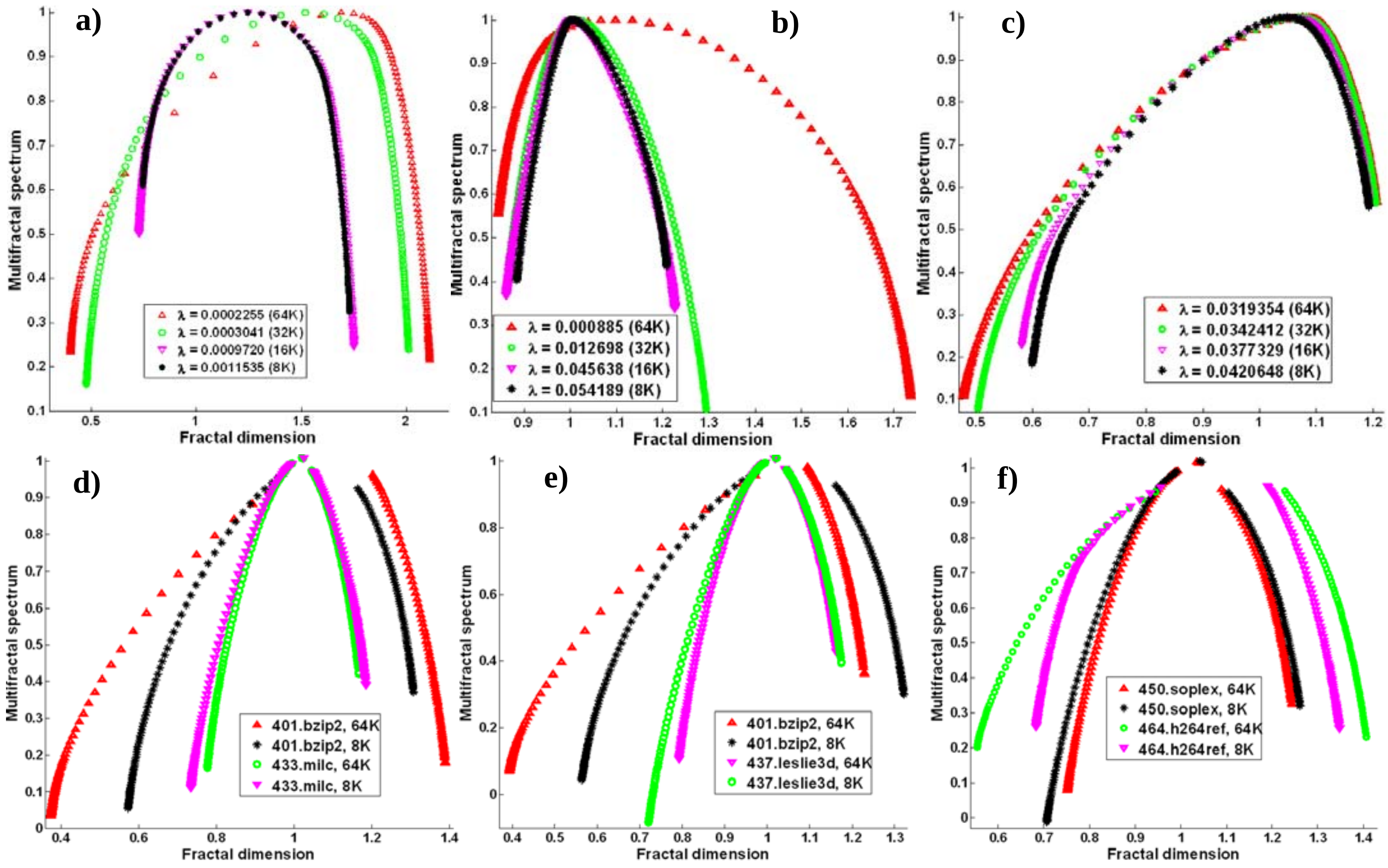

Figure 4. Multifractal spectrum of the inter-arrival times of the data packets for three applications running on a 10×10 NoC with varying L1 cache sizes: 400.perlbench with 1 flit (a), 403.gcc with 4 flit (b), 401.bzip2 with 4 flit (c) data packets. The broad range of fractal dimensions exhibited in these graphs confirm the existence of multifractality in NoC traffic. Multifractal spectrum of inter-arrival times of data packets for six applications from dynamic multiple-applications scenarios, running on a 10×10 mesh NoC with various L1 cache sizes (i.e., 8K, 64K) and 8 flits data packets: 401.bzip2 (d) mapped on the PE located at (0,0) and the 433.milc mapped on PE located at (9,9), 401.bzip2 (e) mapped on the PE located at (0,0) and the 437.leslie3d mapped on PE located at (9,9) in Figure 4.e, and 450.soplex mapped on the PE located at (0,0) and the 464.h264ref (f) mapped on PE located at (9,9) in Figure 4.f.

order memory effects [21]. On the other hand, the anti-persistent tendency shows that the stochastic process deviates from time periodicity and has memory effects of lesser degree.

While the multifractal spectrum for 403.gcc shrinks towards the anti-persistent region (lower support of fractal dimensions - left), the multifractal spectrum of 401.bzip2 application shrinks towards the persistent region (higher support of fractal dimensions - right). Nevertheless, both graphs display a broad range of fractal dimensions concentrated around 1 which confirms the existence of multiscale, as well as a high degree of memory.

Dynamic multiple-application scenarios: To investigate the impact of running multiple applications on the NoC traffic characteristics, we report the multifractal spectrum of the inter-arrival times of the data packets for six applications from dynamic multiple-applications scenarios, running on a 10×10 mesh NoC with two L1 cache sizes (*i.e.*, 8K, 64K) and 8-flit data packets:

- 401.bzip2 runs on the PE at (0,0) and 433.milc runs on the PE at (9,9) in Figure 4.d;
- 401.bzip2 runs on the PE at (0,0) and 437.leslie3d runs on the PE at (9,9) in Figure 4.e;
- 450.soplex runs on the PE at (0,0) and 464.h264ref runs on the PE at (9,9) in Figure 4.f.

All these plots exhibit a wide range of fractal dimensions which correspond to a highly nonlinear exponent $\tau(k)$ in Equation 2. It should also be noted that, with the increased communication load, the multifractal spectrum shrinks around 1 which corresponds to a high degree of memory effects. We also notice that the multifractal spectrum for 401.bzip2 and 464.h264ref exhibit a short discontinuity which can be attributed to the artifacts of fitting simulation data. However, this does not affect the conclusion about the existence of multifractal behavior. Also, the asymmetry displayed by all these multifractal spectra can be interpreted as the heterogeneity observed in both single and multiple application workloads traffic patterns. As discussed in the next section, the existence of multifractality in NoC traffic has direct implications in the design and optimization of NoC architectures.

VII. Causes and Implications of Multi-fractal behavior Case Study on Estimating Deadlines

In this section, we discuss possible causes of multifractality of the data inter-arrival times. We especially consider the compute versus stall times of applications. Finally, we briefly discuss a few implications of multi-fractality in CMP scheduling.

Compute vs. Stall times, Request-Reply Latencies, and MLP: Generally speaking, during its execution an application alternates between useful compute periods (*i.e.* periods when

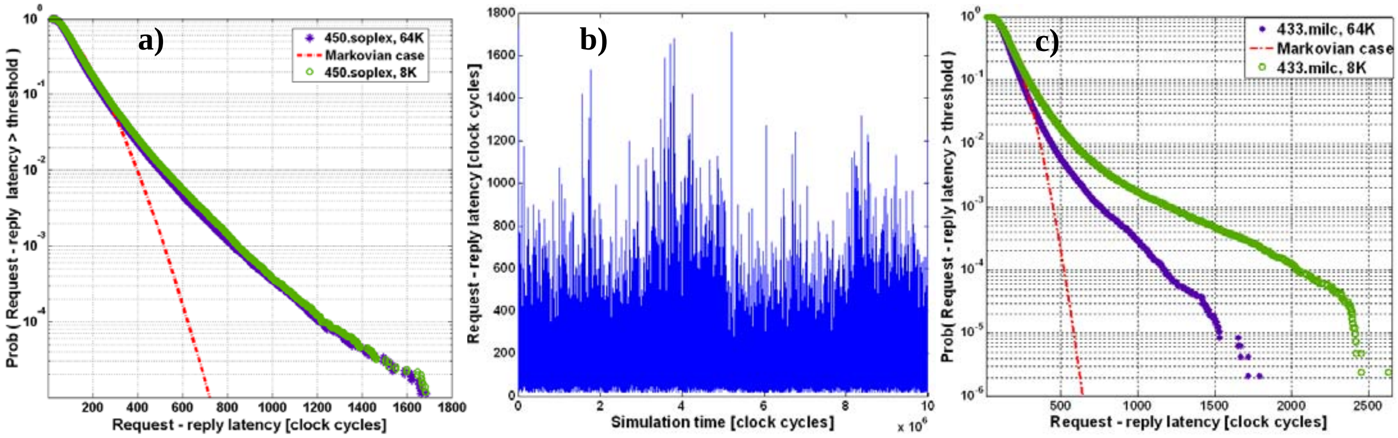

Figure 5. a) The probability of the latency encountered by a data request to exceed a certain threshold for the PE located at (2,4) when running 450.soplex on a 10×10 NoC with 64K and 8K L1 cache sizes , XY wormhole routing and 8 flits per packet. b) The plot for the latencies encountered for each data requested by node (2,4) from other nodes on the 10×10 NoC. c) The probability of the latency encountered by a data request to exceed a certain threshold for the PE at (4,4), running 433.milc on a 10×10 NoC with two L1 cache sizes (64K and 8K), wormhole deflection routing and 8 flits per packet.

there is forward progress on the application execution) and useless stall periods [34]. While a PE is in the stall state, it makes no progress on the application execution.

The stall time experienced by a PE is tightly coupled with the experienced request-reply latencies. We define the *request-reply latency* (RRL) as the time elapsed between the creation of an address request (address packet) and the receiving of its associated reply (data packet).

If we consider the use of strict in-order PEs that cannot overlap the latency of multiple packets, then the stall time experienced by a PE is dependent only on request-reply latencies, and can be defined as the sum of request-reply latencies of all packets. However, in our experimental setup, we evaluate state-of-the-art out-of-order execution PEs that can overlap the latency of multiple packets. Therefore, the stall time experienced by the core is not a simple sum of all packet latencies. The degree of memory level parallelism an application has is one of the most significant reason why the experienced stall time deviates from the sum of the request-reply latencies.

Memory Level Parallelism (MLP) is defined as issuing and servicing of multiple requests in parallel [35]. During the execution of an application by an out-of-order PE, an application might have multiple outstanding requests. The latencies for some of the requests will overlap with the latencies of older requests. The degree of overlap between request latencies relates to the application MLP; higher MLP applications have more overlap among their requests.

Implications: With increasing amount of traffic in the NoC, request-reply latencies start to increase. As a result, stall periods start dominating the compute periods, leaving the PE idle for most of its execution time and thus reducing its performance. In our experiments with multiple-application scenarios, we observe that the stall times dominate the processing times for several applications[1]. This is especially true for applications with high injection rates (*e.g.*, 450.soplex - See Table 2).

Case Study: In Figure 5.a, we report the probability of request-reply latencies exceeding a certain threshold for 450.soplex. The presented data is collected in a dynamic multi-application scenario where 450.soplex runs on the PE located at (2,4) of a 10×10 mesh NoC. The routing algorithm is XY routing and the simulation runs for 10M clock cycles. We should note that the PE located at (2,4) running 450.soplex spends 82.7% of the execution time in stall state. The dotted line represents the standard Markovian behavior which corresponds to an exponential probability of the request-reply latencies to exceed a given threshold. More precisely, the probability of request reply latency to exceed a certain threshold is given by: $P(RRL > x) \sim e^{-[x/c]^b}$, where b and c are the shape and scale parameters, respectively.

As it can be observed from Figure 5.a, the Markovian curve underestimates the probability of exceeding a certain latency (i.e., the probability of missing a deadline). For instance, according to the Markovian curve the probability of waiting for a certain data packet approximately 600 clock cycles is 0.0001, while it becomes 0.007 for the multifractal curves. Thus, Markovian curves are quite optimistic in their estimations. This implies that when the multifractal effects are ignored, the likelihood of missing a deadline and failing to predict the actual request-reply latency is at least an order of magnitude higher. For completeness purposes, in Figure 5.b we report the magnitude and burstiness of request-reply latencies encountered at node (2,4) over the entire simulation.

In Figure 5.c we present the probability of exceeding a certain threshold in request-reply latency for another application, 433.milc. The presented data is again collected in a dynamic multi-application scenario, at the node located at (4,4) of a 10×10 mesh NoC using two different L1 cache sizes (*i.e.*, 64K and 8K). In this scenario, the PE located at (4,4) running 433.milc is stalled 60% of the time. The Markovian curve again underestimates the probability of exceeding a certain threshold in data latency. However, the difference between the estimations of multifractal curves and the Markovian curve is higher, especially when the packet injection rate is higher. For instance, the Markovian case predicts a probability of 0.00006 to exceed a threshold of 530 clock cycles, while the multifractal approach predicts a probability of 0.0045.

Effect of the Routing Algorithm: Different from the results presented in Figure 5.a, in the experiment presented in Figure 5.c, wormhole deflection is used as the routing algo-

[1] If a core spends more than 60% of its execution in stall state, we say that its stall times dominate its processing times.

rithm. Comparing the results presented in Figure 5.a and Figure 5.c, it can be observed that the two multifractal probability curves corresponding to 64K and 8K L1 cache sizes are almost the same in Figure 5.a while they drastically deviate from one another in Figure 5.c. This is mostly due to the change in the routing algorithm. Therefore, it can be stated that the fixed XY routing does not introduce secondary effects as the deflection algorithm does. In addition, the higher degree of freedom in packet routing introduces *i)* higher latencies as L1 cache size decreases and *ii)* a more pronounced nonlinear behavior in the distribution of stall times. Using the *QuaLe* model in Section IV, it can be shown that the distribution of stall times has multifractal features that are captured via the fitness distribution.

Future Work: There are two major directions for future research. First, understanding the reasons for multi-fractality in NoCs is important. We hypothesize that the discrepancy between the Markovian curves and multi-fractal curves are due to effects like memory-level parallelism and phase behavior in the memory intensity of applications. Further research is needed to pinpoint the causes of multi-fractality.

Second, developing better NoC policies based on the understanding of multi-fractality can prove fruitful. In our experiments, we have used application/distribution oblivious routing and packet arbitration policies (XY and deflection routing as routing algorithms and round-robin for packet arbitration). However, as discussed in this section, stall times, which are a function of the application network intensity and memory level parallelism, tend to dominate compute times. Therefore, new research aimed at designing application/stall time aware routing and scheduling policies, that can account for the multifractal features of various applications and thus prioritize the applications in the network accordingly, can be very promising.

VIII. Conclusion

This paper provides evidence that in NoC-based CMP systems with a large number of components, the NoC traffic needs to be characterized using a multifractal approach rather than standard Markovian approach. Using a new theoretical model, we have investigated the effect of packet injection rate and the data packet sizes on the multifractal spectrum of NoC traffic. For several applications, we have shown how the existence of multifractality can be identified and used in estimating the probability of missing a deadline for applications with packet deadline requirements. We have further shown that the stall times experienced by the applications start dominating the compute times, especially in loaded traffic scenarios, reducing the utilization of cores drastically. Therefore, our future work will focus on developing application/distribution aware routing and scheduling policies for CMP platforms based on multifractal features of the NoC traffic model proposed in this paper.

Acknowledgements: The authors (P.B.,R.M.) acknowledge support from Global Research Collaboration (GRC) and National Science Foundation (NSF) via grants 2008-HJ-1823, CCF-0702420, and CCF-0916752, respectively. The authors (M.K.,O.M.) acknowledge the support from CMU SEI, CyLab, and equipment donated by Intel Corporation.

IX. References

[1] A.-L. Barabási and H. E. Stanley, *Fractal Concepts in Surface Growth*, Cambridge University Press, 1995.

[2] R. Benzi, G. Paladin, G. Parisi and A. Vulpiani, “On the Multifractal Nature of Fully Developed Turbulence and Chaotic Systems,” *J. Phys. A: Math. Gen*., 17, pp. 3521-3531, 1984.

[3] S. Bhansali, W. Chen, S. De Jong, A. Edwards, R. Murray, M. Drinic, D. Mihocka, J. Chau, “Framework for Instruction-Level Tracing and Analysis of Program Executions,” *Int. Conf. on Virtual Execution Environments*, 2006.

[4] P. Bogdan and R. Marculescu, “Quantum like Effects in Networks-on-Chip Traffic Behavior,”*Proc. of Design Automation Conference*, San Diego, 2007.

[5] P. Bogdan and R. Marculescu, “Statistical physics approaches for network-on-chip traffic characterization,” *Proc. of IEEE/ACM Intl. Conf. on Hardware/ Software Codesign and System Synthesis*, Grenoble, France, 2009.

[6] M.E. Crovella, A. Bestavros, “Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes,” *IEEE/ACM Trans. on Networking*, 1997.

[7] W.J. Dally, B. Towles, “Route Packets, not Wires: On-chip Interconnection Networks,” *in Proc. of the Design Automation Conference*, 2001.

[8] W. J. Dally and B. Towles, Principles and Practices of Interconnection Networks. San Mateo, CA: Morgan Kaufmann, 2004.

[9] S. N. Dorogovtsev and J. F. F. Mendes, *Evolution of Networks: from biological networks to the Internet and WWW*, Oxford Univ. Press, 2003.

[10] H. Fowler, W. Leland, “Local Area Network Characteristics, with Implications for Broadband Network Congestion Management,” *IEEE JSAC*, 9(7), pp. 1139-1149, 1991.

[11] U. Frisch, P.-L. Sulem, and M. Nelkin, “A Simple Dynamical Model of Intermittent Fully Developed Turbulence,” *J. of Fluid Mech.*, vol. 87, 1978.

[12] H. L. Grant, R. W. Stewart, A. Molliet, “Turbulence spectra from a tidal channel,” *J. Fluid Mech*., 12, No. 2, pp. 241-268,1962.

[13] D. Harte, *Multifractals: Theory and Applications*, CRC Press, 2001.

[14] John L. Henning, “SPEC CPU2006 Benchmark Descriptions”, *ACM SIGARCH Newsletter*, Vol. 34, No. 4, Sept. 2006.

[15] R. Jain, S. Routhier, “Packet Trains - Measurements and a New Model for Computer Network Traffic,” *IEEE JSAC*, 4(6), pp. 986-995, 1986.

[16] A. N. Kolmogorov, “Local Structure of Turbulence in an Incompressible Liquid for very Large Reynolds Numbers,” *Proc. Acad. Sci. URSS*, 30, 1941.

[17] C. K. Luk, R. Cohn, R. Muth, H. Patil, A. Klauser, G. Lowney, S. Wallace, V. Reddi, and K. Hazelwood, “Pin: Building Customized Program Analysis Tools with Dynamic Instrumentation,” *Proc. of the ACM SIGPLAN Conf. on Programming Language Design and Implementation*, USA, 2005.

[18] B. B. Mandelbrot, “Intermittent Turbulence in Self-Similar Cascades: Dievergence of High Moments and Dimension of the Carrier,” *J. of Fluid Mech*., 63, pp. 331-350, 1974.

[19] B. B. Mandelbrot, *Les Objets Fractals: Forme, Hasard et Dimension*, Flammarion Publisher, 1975.

[20] R. N. Mantegna and H. E. Stanley, *An Introduction to Econophysics*, Cambridge University Press, 2000.

[21] R. T. J. McAteer, C. A. Young, J. Ireland and P. T. Gallagher, *Astrophys. J.*, pp. 662-691, 2007.

[22] T. Moscibroda and O. Mutlu, “A Case for Bufferless Routing in On-Chip Networks”. *Proc. of the 36th International Symposium on Computer Architecture (ISCA)*, pp.196-207, Austin, TX, June 2009.

[23] M. Newman, A.-L. Barabási and D. J. Watts, *The Structure and Dynamics of Networks*, Princeton University Press, 2006.

[24] E. A. Novikov and R. Stewart, “Intermittency of turbulence and spectrum of fluctuations in energy-dissipation,” *Izv. Akad. Nauk. SSSR. Geofiz*, 1964.

[25] H. Patil, R. Cohn, M. Charney, R. Kapoor, A. Sun, and A. Karunanidhi, “Pinpointing Representative Portions of Large Intel Itanium Programs with Dynamic Instrumentation,” *Intl. Symp. on Microarchitecture*, pp. 81–92, 2004.

[26] V. Paxson, S. Floyd, “Wide-Area Traffic: The Failure of Poisson Modelling,” *in Proceedings of SIGCOMM*, 1994.

[27] A. Scherrer, A. Fraboulet, and T. Risset, “Automatic phase detection for stochastic on-chip traffic generation,” *Proc. of Intl. Conf. on Hardware/Software Codesign and System Synthesis*, Seoul, Korea, October 22 - 25, 2006.

[28] D. Schertzer and S. Lovejoy, “Physical Modeling and Analysis of Rain and Clouds by Anysotropic Scaling of Multiplicative Processes, *J. Geophys. Research*, 92, pp. 9693-9714, 1987.

[29] V. Soteriou, H. Wang, L. Peh, “A Statistical Traffic Model for On-Chip Interconnection Networks,” *in Proc. of IEEE Int. Symp. on Modeling, Analysis, and Sim. of Comp. and Tel. Sys.*, Monterey, California, September 2006.

[30] M.S. Taqqu, V. Teverovsky, W. Willinger, “Is Network Traffic Self-Similar or Multifractal?,” Fractals, 1997.

[31] G. Varatkar and R. Marculescu, “On-chip traffic modeling and synthesis for MPEG-2 video applications,” *IEEE Trans. on VLSI*, 12, pp.108-119, 2004.

[32] T. Vicsek, *Fractal growth phenomena*, World Scientific, 1999.

[33] T. Vicsek, *Fluctuations and scaling in biology*, Oxford Univ. Press, 2001.

[34] O. Mutlu, H. Kim,Y. N. Patt, “Efficient Runahead Execution: Power-Efficient Memory Latency Tolerance,” Micro, IEEE,Vol. 26, Issue 1, 2006.

[35] A. Glew. MLP Yes! ILP No! Memory Level Parallelism, or, Why I No Longer Worry About IPC. In *ASPLOS Wild and Crazy Ideas Session*, 1998.

[36] S.F. Timashev, “*Self-similarity in Nature*”, AIP Conf. Proc., 2000.