

# Application Traffic Modelling & Optimization for NoC Communication

*Submitted in fulfillment of the requirement of the degree of*

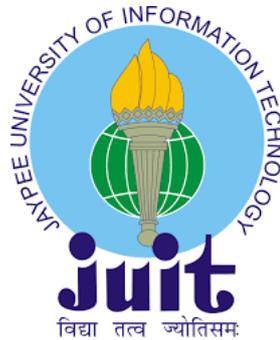
**Doctor of Philosophy**

*by*

**Amit Chaurasia**

*Supervisor*

**Dr. Vivek Kumar Sehgal**



**Department of Computer Science & Engineering & Information  
Technology**

**JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY,  
WAKNAGHAT, SOLAN-173234, HIMACHAL PRADESH,  
INDIA**

**(APRIL 2020)**

Dedicated to

*The Grand Weaver* of my life,

***LORD KRISHNA***

# Acknowledge

“... *karmaṇyevādhikāraṣṭe mā phaleṣu kadācana mā karmaphalāheturbhūrmā te saṅgo'stvakarmaṇi...*” -Bhagawad Gita 2:47

Throughout this adventure, I felt as if *I was only chasing the moments of fire that were already predefined meticulously by Someone*. I am thankful to that **Lord Krishna** for encouraging me in accomplishing this thesis. He has been with me in accomplishing this thesis. On several occasions, it looked as though, for one reason or another, there had been no way ahead to progress. Yet, at the eleventh hour and sometimes even at the twelfth hour, a way was opened. Without God's help, guidance, encouragement and support through some very difficult times, my thesis could not have been written at all.

*A best teacher would invigorate the passion of a student*. My deep sense of gratitude and profound thanks to my supervisor **Dr. Vivek Kumar Sehgal**, for his valuable guidance, patience, encouragement and the freedom given to me in carrying out research. Throughout my PhD tenure, he is always approachable and helpful. Discipline has been put as our first priority and this played a vital role in the success of this thesis.

I sincerely thanks to Prof. (Dr.) Satya Prakash Gharera, for his valuable suggestions and comments throughout the whole presentations. I am also thankful to the faculty members on my reading and orals committee's members: Dr. Harsh Sohal, Dr. Jagpreet Sidhu, Dr. Amit Kumar Jakhar and Dr. Yugal Kumar for providing a valuable feedback on the research results. Although it is not possible to name individually, I cannot forget my well wisher of Jaypee University of Information Technology for their persistent support and cooperation I acknowledge the Department of Computer Science Engineering and Information Technology for the infrastructure provided by them to carry out my research work. I also thank all my friends for their fellowship, without whose support my life might have been miserable here.

A big *THANK YOU* to all. Finally, I thanks my father “Anant Lal Chaurasia” and my mother “Anita Chaurasia” my wife “Usha Chaurasia” and little lovely daughter “Yashika” for supporting me throughout my studies at University and as required they always shown fullfledged matured behaviors to complete my work without requesting any of the unnecessary needs.

**Amit Chaurasia**

# Supervisor's Certificate

This dissertation entitled **Application Traffic Modelling & Optimization for NoC Communication** by **Mr. Amit Chaurasia**, Roll no. 136201 at **Jaypee University of Information Technology, Wagnaghat, Solan (HP), India**, is approved for the degree of Doctor of Philosophy.

A handwritten signature in blue ink that reads "Vivek Sehgal". The signature is written in a cursive style and is underlined with three parallel lines.

Supervisor

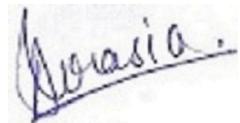
Dr. Vivek Kumar Sehgal

Date: 24-04-2020

Place: Solan

# Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.



(Amit Chaurasia)

Roll No: 136201

Date: 24-04-2020

Place: Solan

# Abstract

The present study aims to develop a simplified methodology for the generation of Synthetic traffic for multicore architecture. To begin with, self-similarity processes are studied for the generation of synthetic traffic. The exponent known as the Hurst parameter is calculated by various method from Rescaled-range method to Whittle's Estimator used to represent the self-similar process. The synthetic traffic is used to simulate on the Mesh architecture of different scalable sizes to decrease the simulation time and it provides the flexibility to network designers. The synthetic traffic is generated for the advanced class of MPEG-4 videos with HEVC. The algorithm presented here requires less time for its generation as its complexity is less as compared with other existing algorithms for the generation of Synthetic traffic.

The synthetic traffic is simulated on different classes of traffic patterns such as Complement, Neighbourhood and Uniform patterns which constitute the selection source and destination pair. Some of the important parameters such as end-to-end latency, link utilization, packet loss probability are calculated based on the simulation outcomes helps in understanding the communication paradigm and helps the network designer for a better selection of communication resources in the early design process of Networks-on-chip architecture.

Subsequently, the role of non-Gaussian processes is studied with the Hermite process. In this study, the role and requirement of Non-Gaussian process are identified, various class of processes identified in Hermite process, the Rosenblatt process is one of them whose distribution is non-Gaussian. The algorithm designed for generating Non-Gaussian process based on Rosenblatt process whose function is the integral of time and space domain concerning *Fractional Brownian Motion*.

The generation of Gaussian and Non-Gaussian process are compared with Rosenblatt Non-Gaussian process shows the burstiness in dataflow is low in Gaussian and Non-Gaussian process as compared with Rosenblatt Non-Gaussian process. The results show the Rosenblatt Non-Gaussian process are better to simulation and representing the actual dataflow.

---

## List of Symbols

---

### Variables

$H$  Hurst Exponent

$k$  Self-similarity order

### Greek Symbols

$\gamma$  Gamma function

$\sigma^2$  Variance

### Acronyms

CIET Circular Embedding Technique

FBM Fractional Brownian Motion

FFT Fast Fourier Transform

FGN Fractional Gaussian Noise

LRD Long-range Dependency

SRD Short-range Dependency

---

## List of Figures

---

1.1	(a) Fractal-Tree (b) Hilbert-Contour (c) Sierpinski-triangle (d) Koch-symbol	3
3.1	(a) 64 nodes Mesh Topology (b) 16 nodes Mesh interconnection network by OMNET++	17
3.2	4 × 4 Mesh interconnection network's avg. hop-count for the (a) Uniform-traffic (b) Tornado-traffic (c) Complement-traffic	20
3.3	8 × 8 Mesh interconnection network's avg. hop-count for the traffic (a) Uniform-traffic (b) Tornado-traffic (c) Complement-traffic	21
3.4	4 × 4 Mesh interconnection network's avg. end-to-end latency for distinct traffic for buffer-size (a) 1024 bits (b) 512 bits (c) 256 bits	22
3.5	8 × 8 Mesh interconnection network's avg. end-to-end latency for distinct traffic for buffer-size (a) 1024 bits (b) 512 bits (c) 256 bits	23
4.1	Variance-Time representation for four Synthetic Traces (a) 1 (b) 2 (c) 3 (d) 4	25
4.2	RS method representation for four Synthetic Traces (a) 1 (b) 2 (c) 3 (d) 4	27
4.3	Fractional Gaussian Noise (FGN) for Synthetic Traffic (a) 1 (b) 2 (c) 3 (d) 4	30
4.4	Fractional Gaussian Noise generation procedure	31
4.5	Synthetic Trace generation procedure	34
4.6	Application mapping of the functions for the MPEG4	35
4.7	4 × 4 Mesh interconnection network's average end-to-end latency for distinct patterns (a) Uniform-traffic (b) Complement-traffic and (c) Tornado-traffic for buffer-size 32 bytes	38

4.8	4 × 4 Mesh interconnection network’s average end-to-end latency for distinct patterns (a) Uniform-traffic (b) Complement-traffic and (c) Tornado-traffic for buffer-size 64 bytes . . . . .	39
4.9	4 × 4 Mesh interconnection network’s average end-to-end latency for distinct patterns (a) Uniform-traffic (b) Complement-traffic and (c) Tornado-traffic for buffer-size 128 bytes . . . . .	40
4.10	4 × 4 Mesh interconnection network under distinct load average power for distinct patterns (a) Uniform-traffic (b) Complement-traffic (c) Tornado-traffic . . . . .	42
4.11	Analytical packet loss-probability with average input-rate ( $m$ ) fixed with varying utilization ( $\rho$ ) Original-files (a) 1 (b) 2 (c) 3 (d) 4 and Synthetic Traces (g) 1 (h) 2 (g) 3 (h) 4 Using Norros Technique . . . . .	43
4.12	Analytical packet loss-probability with mean input-rate ( $m$ ) varying with fixed utilization ( $\rho$ ) Original-files (a) 1 (b) 2 (c) 3 (d) 4 and Synthetic TracesS (g) 1 (h) 2 (g) 3 (h) 4 Using Norros Technique . . . . .	44
4.13	Packet loss-probability for Synthetic-Trace (a) 1 (b) 2 (c) 3 (d) 4 . . . . .	46
4.14	Power [mW] for each cores by traces generated from <i>CIET</i> -algorithm and <i>Fast-Approximation</i> technique (a) Complement-traffic (b) Tornado-traffic (c) Uniform-traffic . . . . .	47
5.1	Gaussian Synthetic Traces average link-utilization for 64 nodes mesh interconnection network for (a) Bit-complement-traffic (b) Neighbourhood-traffic (c) Tornado-traffic (d) Uniform-traffic . . . . .	52
5.2	Non-Gaussian Synthetic Traces average link-utilization for 64 nodes mesh interconnection network for (a) Bit-complement-traffic (b) Neighbourhood-traffic (c) Tornado-traffic (d) Uniform-traffic . . . . .	53
5.3	Non-Gaussian and Gaussian Synthetic Traces avg. latency for 8 × 8 architecture for (a)Uniform (b) Tornado (c) Neighbourhood (d) Bit-Complement traffic . . . . .	54
5.4	Non-Gaussian and Gaussian Synthetic Traces average packet Loss Probability for 8 × 8 mesh architecture (a) Trace 1 (b) Trace 2 (c) Trace 3 (d) Trace 4 . . . . .	55
6.1	Average Distance(number of hops) for (a) 16 (b) 64 nodes Mesh topology for Complement, Tornado, Neighbor & Uniform traffic . . . . .	60
6.2	Average estimation of End-to-End latency(ns) for (a) 16 (b) 64 nodes Mesh topology for Complement, Tornado, Neighbor & Uniform traffic . . . . .	60
6.3	Average Network latency(ns) for (a) 16 (b) 64 nodes Mesh topology for Complement, Tornado, Neighbor & Uniform traffic . . . . .	61
6.4	Average Loss probability for (a) 16 (b) 64 nodes Mesh topology for Complement, Tornado, Neighbor & Uniform traffic . . . . .	61

6.5	Boxplot for Average Link Utilization(%) for (a) 16 (b) 64 nodes Mesh topology for Complement, Tornado, Neighbor & Uniform traffic . . . . .	62
6.6	Power estimation for each tile of $4 \times 4$ Mesh network (a) Fractional Rosenblatt process (b) Simple Hermite process . . . . .	62
6.7	Average latency estimation of End_to_End (ns) for 64 nodes Mesh topology for non-Gaussian, Gaussian and Rosenblatt process for (a) Complement-traffic (b) Tornado-traffic (c) Neighbor-traffic (d) Uniform-traffic . . . . .	64

---

## List of Tables

---

4.1	Hurst Exponent Using 3 Methods . . . . .	28
4.2	Cores Implementing the Processes . . . . .	33
4.3	Statistics for different Original Video files . . . . .	37
4.4	Latency Comparison for different buffer length . . . . .	37
4.5	Router's Components power under different fractional-load . . . . .	41
4.6	Power Estimation under different traffic pattern . . . . .	41
5.1	Hurst Exponent for Non-Gaussian & Gaussian traces . . . . .	51
5.2	Gaussian's & Non-gaussian's traces Skewness & Kurtosis . . . . .	51
5.3	Non-Gaussian & Gaussian traffic's link-utilization for traffic patterns . . . . .	51
5.4	Non-Gaussian & Gaussian traffic's avg. latency for traffic patterns . . . . .	52
5.5	Parameters used for Simulation on OMNET++ . . . . .	56
6.1	Average Packet-Loss Probability for comparison for 64 mesh architecture nodes . . . . .	65
6.2	Average end-to-end latency [ns] comparison for 64 mesh architecture nodes . . . . .	65

---

## Table of Contents

---

	Page
Acknowledge	i
Supervisor's Certificate	ii
Declaration	iii
Abstract	iv
List of Symbols	v
List of Figures	vi
List of Tables	ix
<b>1 Introduction</b>	<b>1</b>
1.1 Self-Similarity Basics . . . . .	2
1.2 Motivation . . . . .	5
1.3 Identified Research Gap . . . . .	5
1.4 Objectives of the Thesis . . . . .	5
1.5 Organization of the Thesis . . . . .	6
<b>2 Literature Review</b>	<b>7</b>
2.1 General . . . . .	7
2.2 Use of MPEG2 for traffic modeling . . . . .	9

2.3	Limitations of non-Gaussian synthetic traffic . . . . .	9
2.4	Optimization of buffers placed in MPSoC . . . . .	15
<b>3</b>	<b>Synthetic Self-similar Traces based Optimization of Buffer-Size for Different Traffics for Networks-on-Chip</b>	<b>17</b>
3.1	Traffic Distribution Patterns . . . . .	17
3.2	Experimental Results . . . . .	18
3.3	Conclusion . . . . .	19
<b>4</b>	<b>Energy Efficient Application Traffic Modeling and Synthesis for MPEG4 based on networks-on-chip architecture</b>	<b>24</b>
4.1	Long-Range Process Dependency . . . . .	24
4.2	Various Methods to calculate Hurst Parameter . . . . .	25
4.2.1	<i>Variance-Time Method</i> . . . . .	25
4.2.2	<i>Rescaled Adjusted Range Analysis</i> . . . . .	26
4.2.3	<i>Whittle's Estimation</i> . . . . .	28
4.3	Synthetic Traffic Generation . . . . .	29
4.4	Packet Buffer-Loss Probability . . . . .	33
4.5	Results . . . . .	36
4.6	Conclusion . . . . .	45
<b>5</b>	<b>Synthetic Traffic Performance for Non-Gaussian and Gaussian based traffic on NoC</b>	<b>48</b>
5.1	Hermite Process of Non-Gaussian . . . . .	49
5.2	Hurst Parameter or Exponent for non-Gaussian process . . . . .	50
5.3	Experimental Results . . . . .	51
5.4	Conclusion . . . . .	57
<b>6</b>	<b>Non-Gaussian Traffic Modelling for Multicore Architecture Using Wavelet Based Rosenblatt Process</b>	<b>58</b>
6.1	Rosenblatt Process . . . . .	58
6.2	Algorithm for generation of Fractional Rosenblatt Process . . . . .	59
6.3	Experimental Results . . . . .	62
6.4	Conclusion . . . . .	65
<b>7</b>	<b>Conclusions and Future Work's Scope</b>	<b>67</b>
7.1	Conclusions . . . . .	67
7.2	Scope for Future Work . . . . .	68
	<b>Appendix Appendix</b>	<b>70</b>
.1	Validation of Rosenblatt-process of Hermite Class . . . . .	70

<b>Bibliography</b>	<b>71</b>
<b>List of Publications</b>	<b>82</b>

# CHAPTER 1

---

## Introduction

---

The advancement in technology makes the life of people simpler and simpler but this is happened because of the introduction of algorithms and fast computational devices. With billions of transistor are embedded into a single die makes computation much faster with very little heat dissipation. As the technology is improving day by day the need for much faster computational devices is the need of the hour. The case of scaling, assimilation of heterogeneous new functionalities into agile technologies became a propulsive component for the technology. The trend, of multiple processing variety in conjunction with minimization of resources, increased difficulty in the multicore cycle has led to an increasingly central estimation elements [1]. There, for example, is an design for the protein-folding computation with more than 1200 cores [2]. Therefore the need for bus-based connectivity becoming less popular as a comparison to parallelism as it decreases the latency and power and also increases the throughput. Yet, it brings up other challenges before the network-designer mainly the use of optimal communication resources used for networking of packets. The system know how, the network design knows application-based simulation is required for network understanding. The requirement of using the target function is the inflexibility and takes enormous day-time for simulation and if traffics used with shorter-length of target application should swift the simulation and hence the traffic modelling need becomes important.

Traffic modelling is an critical dimension for quality service for which the optimum usage of the communication resources. In the conventional communication, where the number

of calls or data points represented as the Poisson system under which the occurrence of traffic-packets with respect to time interpreted as an exponential vector. Nevertheless, with the greater complexity is implemented in connectivity with the type of multimedia traffic, which in essence is bursty, traffic analysis shows not just the direction of arrivals, but also the variation bandwidth-connectivity.

In this thesis, we address the effect and significance of self-similar traffic for various mesh architecture for communication. The generation of traffic with self-similarity principle using the multimedia applications statistical properties [3] whose sum is not possible from Eq. (1.1) shows burstiness property. For the generated traffic the *auto-correlation function* will not be a summable value, but rather an indefinite value. The *auto-correlation function* break downs so gradually that any aggregate limit function won't remove the *auto-covariance* from this method. The sum of the function of auto-correlation is represented in Eq. (1.1) as

$$\sum_{-\infty}^{\infty} c(\kappa) = \infty \quad (1.1)$$

where  $c(\kappa)$  is the function of *auto-correlation* with number of lags equal to  $\kappa$ . The *Long-Range Dependence Framework (LRD)* displaying this dependency property. If this *auto-correlation function* is summed to any positive value, this process is known as to be *Short-Range Dependence Framework (SRD)*.

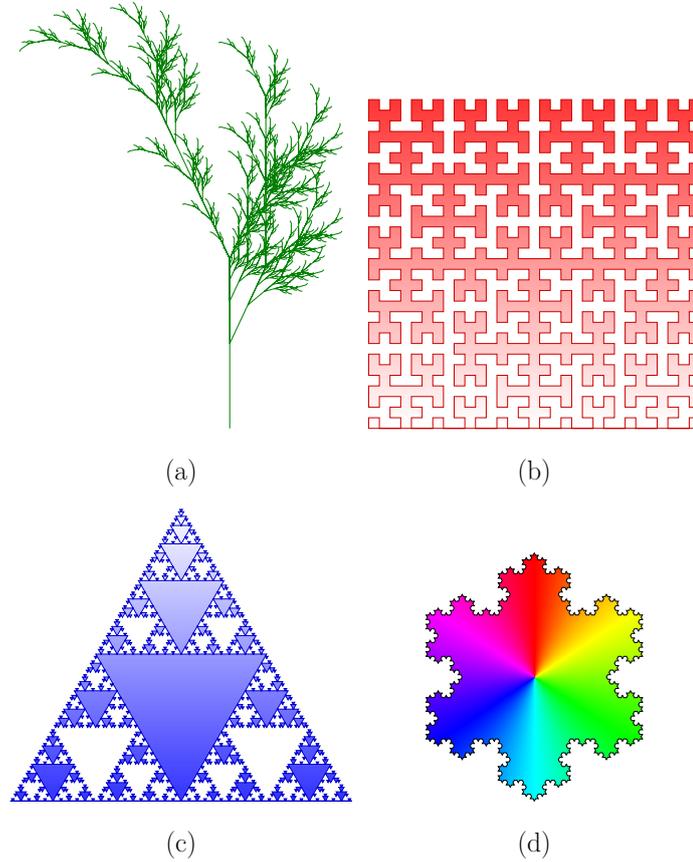
For simulated modeled traffic [3], performance analysis [4] is needed to assess the NoC behavior and efficiency during the initial design phase simulation. NoC efficiency evaluation during the initial design phase depends heavily on the kind of patterns selected in traffic coordination. Therefore, the range of traffic to demonstrate correlations target applications, it is supposed to be pivotal.

There are three categorized different classes of traffic patterns [5] for example real-time traffic trends for services, digital traffic structures and the latter derive statistical features from the current application because of its simplicity to create simplified models of traffic than initial application traffic.

## 1.1 Self-Similarity Basics

The attributes to the same features designed at all the possible scales is known as self-similar process [6]. Specific examples of self-similarity can be seen in the Fig. 1.1. The auto-similar properties are shown by four subfigures of the *Sierpinski-triangle*, *Fractal-Tree*, *Hilbert-Contour* and *Koch-symbol*. They are scaling to a greater degree where finer details display the same properties that are generated from and this cycle can be recursively replicated. The structure of its origin will be established. This will be close in terms of traffic 100 *ms* network containers are collected into the 100 *seconds* network containers

and this 100 *seconds* network containers are collected into 100 *minutes* network container. In the Poisson process as the distribution is not interrelated and as the increase the process that makes the flat smooth line so the self-similar are different from the Poisson process. There is no place for burstiness in the Poisson traffic.



**Fig. 1.1** (a) Fractal-Tree (b) Hilbert-Contour (c) Sierpinski-triangle (d) Koch-symbol

Processes identical to each other are assessed by the single exponent ranges from  $(0, 1)$  known as *Hurst exponent* ( $H$ ). A time-series is chosen as the process  $L(t)$  such that the  $t^{\text{th}}$  time, of its aggregated length  $L^{(m)}$  of  $L$  at gathering level of  $m$ ,

$$L^{(m)}(i) = \frac{1}{m} \sum_{m^{(i-1)+1}}^{mi} L(t) \quad (1.2)$$

$$L^{(m)} = m^{H-1} L \quad (1.3)$$

$$L = m^{1-H} L^{(m)} \quad (1.4)$$

The  $L(t)$  is split into a non-overlapping pieces of  $m$  sizes. The number of blocks are represented by  $i$  and such blocks are averaged. The  $\phi^{(m)}(k)$  denotes the function of

auto-covariance for the  $L^{(m)}$ .

**Definition 1.1.1. (2<sup>nd</sup> Order Self-Similarity)** A method is called by *2<sup>nd</sup> order self-similar* according to function of *auto-covariance* accomplishes the Eq. (1.5),

$$\phi(p) = \sigma^2 \left( (p-1)^{2H} \times 2^{-1} - (p)^{2H} + (p+1)^{2H} \times 2^{-1} \right) \quad (1.5)$$

and it's *asymptotically 2<sup>nd</sup> order self-similar* if satisfies the Eq. (1.6),

$$\lim_{m \rightarrow \infty} \phi^{(m)}(p) = \frac{\sigma^2}{2} \left( (p-1)^{2H} - 2(p)^{2H} + (p+1)^{2H} \right), \quad (1.6)$$

where  $p \geq 1$  and  $H \in [0.5, 1]$

**Definition 1.1.2. (2<sup>nd</sup>-Order Stationary)** A *2<sup>nd</sup> order stationary* process is said if its *auto-covariance* function is represented by  $(\phi(x, y) = \mathbb{E}[(Q(t) - x)(Q(t) - y)])$  is adaptation invariance i.e.,

$$\phi(x, y) = \phi(x + k, y + k), \quad (1.7)$$

where  $k, y, x \in Z$ .

These two definitions are important characteristic for the network traffic simulation for multicore architecture for it's exact simulation on synthesis traffic.

Let us suppose the aggregate time-series  $Y(t)$  of the cumulative process  $X(t)$  to give the expression  $X(t) = Y(t) - Y(t-1)$  which is a process of self-similarity.

$$Y(t) = a^{-H} Y(at) \quad (1.8)$$

where  $a$  is the deflation or contraction factor which can stretch for  $a > 1$  and dilate if  $a < 1$ . The scaled version of time series  $Y(t)$  is represented as  $Y(at)$ , where both pursue the same distribution.

$$Y(t) = t^H Y(1) \quad (1.9)$$

So aggregated-variance  $X^m$  is derivable as,

$$\text{var} (X^{(m)}) = m^{2H-2} \sigma^2. \quad (1.10)$$

So  $Y(t)$  which is used for the modeling of synthetic traffic design is known to be FBM (*fractional\_brownian\_motion*) and its aggregate process is called FGN (*fractional\_gaussian\_motion*)

## 1.2 Motivation

While NoC has many advantages, it has been suggested as a promising approach to the overheads of packet collision in bus-based communication and multiprocessor system-on-chip (MPSoCs) by means of general-purpose communication architectures. When utilizing fast connectivity connections instead of using a single bus, power usage may also be minimized and efficiency increases [7], [8]. In fact, the wire duration and latency of the electrical links can be easily managed while at the same time utilizing the wires more effectively, consuming fewer wiring. NoCs have therefore been the preferred on-chip connectivity model by providing an optimized approach to a broad variety of difficulties in designing the massive MPSoC. As the system becomes huge as the number of nodes or Intellectual property (IP) increases in the communication SoC there will be an increase in the volume of data transferred and to better analyze the system design it requires full simulation outcomes. But, the real-time traffic takes a huge amount of hours and often lacks the flexibility in order to overcome these issues synthetic traffics are designed for fully understanding the network design and the communication resources required even before the actual design of the chip. It allows the network architect to pick optimal resources and to understand the early design process better.

## 1.3 Identified Research Gap

- The generated traffic is available for MPEG2 video codec which are not bursty in nature and not for the MPEG4 videos.
- The algorithms used for generation of synthetic traffic are algorithmic complex and the requirement of less complex algorithm required for the synthetic traffic generation.
- Analysis is not done based on the source-destination pairs which can be done by using the traffic patterns. The other issues such as Power and Packet loss probability were not taken in consideration for synthetic traffic performance on multicore architecture.
- The role and requirement of Non-Gaussian synthetic traffic for NoC Communication is identified as the problem to generate Non-Gaussian synthetic traffic.

## 1.4 Objectives of the Thesis

- To optimize the buffer-size using self-similar traffic.
- To generate synthetic traffic model based on self-similarity principle for MPEG4 videos.
- To generate the traffic for non-Gaussian process and compared with existing traffic.

- To propose an algorithm for non-Gaussian traffic generation and discuss about the requirement of mentioned traffic for networks-on-chip architecture.

## 1.5 Organization of the Thesis

This thesis presents the study carried out. This thesis consists of six chapters and is organized as follows:

**Present chapter** This chapter provides an overview of Synthetic traffic and self-similarity, motivation and objectives of the work.

**Second chapter** In this chapter provides a comprehensive literature review on the present subject, state of the art approaches for synthetic traffic generation.

**Third chapter** The synthetic traffic is generated for the MPEG-4 videos using the properties of self-similar process. The optimal buffer sizes are analyzed using the Norros principle.

**Fourth chapter** Performance of non-Gaussian and Gaussian process is analyzed in this chapter. The synthetic traffic generated for the MPEG-4 videos and the Non-Gaussian process and analyzed.

**Fifth chapter** In this chapter Non-Gaussian synthetic traffic is generated using Rosenblatt process.

**Sixth chapter** This chapter reports the conclusions of this study and scope of future work under the domain of synthetic traffic generation.

### 2.1 General

In today's world the entire system of computing can be played by a single chip which consists of different processing elements, this system came to know as System-on-Chip (SoC). Since the SoC's are scalable hence it leads to the issues of power minimization, synchronization, and system reusability. As the bus-based system suffers from the issue of scalability as the number of collision increases with the number of systems interconnect increases in the system makes the packets prone to collision [9], [10]. As the number of cores or Intellectual Property (IP) core is increasing the volume of data transfer also increases hence the bus-based communication is replaced by packet-based communication [11], [12], [13].

Traffic Models are the mathematical characteristics generated for various class of applications [3]. Network efficiency is highly dependent on real traffic, so reliable traffic models are required to better understand the vast architecture of network protocols, topologies, and implementations. Real implementations are time intensive and lack the flexibility, so computational models are used to measure network efficiency early in the design phase [5]. NoC simulators have very high precision forecasts. It is important to provide a thorough performance review before the prototype's early design. The traffic model requires to abstract two main elements, namely the source and destination of each flow, and the inter-arrival time distribution of packets[14]. As per our best knowledge, no complete

traffic model exists for NoC architecture. Therefore, the network designer relies on the synthetic traffic models in addition with traffic patterns such as uniform traffic, tornado traffic *etc.* for the complete test of the architecture [15], [16].

The traffic models are the inter-arrival time of the packets arrival at source which can be modelled using various technique can be found in [17], [18], [19], [20], [21], [22] and [23]. The traffic models are designed using the distribution function such as Poisson distribution, Bernoulli distribution *etc.*. In [24] describes the failure of the Poisson distribution based traffic model for wide area network. In this the inter-arrival time packets are independent of each other hence lacks the feature of capturing the burstiness. Several works investigated the on-traffic properties and suggested a variety of traffic models to represents their patterns. In [25] gives a summary of the NoC benchmarking requirements. In [4] suggested a tri-tuple traffic model for empirically capturing the multicore architecture traffic characteristics. A generalized traffic model to explain the traffic on-chip properties based on burstiness, injection speed between pairs from origin to terminal [26].

For evaluation technique and generation of traffic for mesh interconnection network are discussed in [27], where 2 types of performance assessment mechanism were used to do the assessment to achieve more practical performance on each network channel and to provide testing of the integrated network. The statistical method for NoC traffic modeling using Gaussian-models which can be used to characterize on-chip design communication patterns [28], [29].

Problems such as the need of traffic interaction for the current sampling technique, the current calculation for representing implementations, and the variety of instruction counts, suggested by the authors in [30]. By refining their analytical parameters NoCLabs and NoCPoint will assess metrics. The literature in [31] evaluated the effect on CMPs system phases of on-chip network activity and suggested methods for using connectivity actions to evaluate the task phases. The analytical models allow for easy calculation of NoC output with reasonable precision and can be implemented in recent design stages. The authors in [32] have created traffic for the network correctly by modelling cache-coherence operation and network injection along with variations of network traffic-patterns. In comparison, the development process is streamlined by leveraging Markov chains with the reduced simulation time. The strategies for traffic classification and buffer behaviour analysis with a novel traffic simulation can be found in [33], [34], [35].

The proposed framework of analysis based on statistical physics is shown to be more effective than the typical approaches based on the theory of queueing and the Markov chain. Meanwhile, they show that typical random traffic patterns based on probability distribution are not capable of capturing some essential traffic component and have a negative effect on assessment and optimization of NoC results. Various NoC simulation environments in

present time provide traffic design models for NoC studies to produce random traffic [36], [37], [38], [39]. There are available few benchmarking systems that target CMP systems of general use, such as SPLASH2 [40] and PARSEC[41]. They primarily focuses on computing applications which are science based and with high-performance. The *E3S* addresses the heterogeneity of various processor-types [42]. A trace based simulation technique and framework for NoC, collecting packet dependencies from entire machine simulations observed from[43]. In [44], comprehensive statistical description can be collected and the artificially applicable benchmark traces generated via their constructed process. The TRIPS software and simulators designs [45], [46], [47] for resource consuming and high efficiency multiprocessor systems. They emphasize the importance of practical workload in evaluating NoC systems.

## 2.2 Use of MPEG2 for traffic modeling

In [48] author presented novel work of generating synthetic traffic using self-similar traffic and speed-up the buffer simulations for the MPEG2 videos, according to our best knowledge the bursty traffic is not generated for the MPEG4 videos which will be burstier than MPEG2 videos. The traffic generation’s algorithm used for the synthetic traffic is algorithmically complex .

The single exponent use for traffic generation is known as the Hurst exponent is used for its generation. There are various method to calculate the Hurst parameter [49], [50] having different property ranges from stability to simplicity. The *R/S plot* method [21], *Variance-time* method [51] and the *periodogram* method [52] are few methods for estimating the Hurst exponent. The *R/S plot* method is the most stable method and rich in information as comparison to other methods and therefore the best to apply for protocols based packet control. The *periodogram* approach is the most simple to enforce and its nature to preserve its stability and used to regenerate the traffic-flow even if requirements from the routers are less. The *periodogram* method doesn’t overestimate the value of Hurst exponent, whereas the *R/S plot* approach is difficult to apply but it has the advantage of determining is not whether the traffic is self-similar in long ranges. The *Variance-time* method is the most un-stable.

## 2.3 Limitations of non-Gaussian synthetic traffic

The non-Gaussian traffic is analyzed for the on-chip architecture since the Gaussian traffics are not burstier hence the new model required for traffic modelling to capture the true properties of raw video files. The traffic generated from the independent identical distributed flow have a marginal Gaussian distribution. But, the marginal distribution cannot always follow the Gaussian distribution. The first study is found in [53] where traffic is analyzed for the TCP protocol are having the properties of non-Gaussianity. The inferences of the authors are non-Gaussianity in the traffic degrades the network

performance due to the greedy flows of the traffic and it has small round-trip time due to small hop counts. The traffic is generated using the wavelet base model for the long-range dependent network traffic model [54] are non-Gaussian. The traffic generated is mostly for the internet [55], [54] and as per our best knowledge, very few studies are done for traffic generation for the multicore architecture. Most of the studies found generating traffic for the on-chip network have the marginal Gaussian distribution and no non-Gaussian distribution are taken into consideration.

The non-Gaussian process is found in the [56] where the process is generated from the Hermite-class polynomial. The use of non-Gaussian process is important for river-flow time series modelling [57], in hydrology [58] and in the internet traffic modelling [59]. The special class of Hermite process known as Rosenblatt process is generated, for the definition of the Rosenblatt process can be found in [60], [61], [62], [63] and [64]. The Rosenblatt process is developed using the proposed wavelet-based analysis synthesis procedures [60], which are obtained in 2 ways, either by truncating the approximated time-series and also by the approximated-coefficients in the Rosenblatt wavelet-type process expansion. The method is also the self-similar process whose increments are stationary and as limit in the Non-Central Limit Theorem (NCLT)[65], [66], [67]. NoC system present a modular approach for the on-chip connectivity problem. The use of appropriate flow-control algorithms helps the bandwidth provided by NoCs be used effectively. The algorithms for flow-control mechanism released for networks deteriorate from large load of communication and delays. Therefore it is at best troublesome to include them in the sense of the NoC. For this cause, it is proposed a closed-loop flow-control system which is predictive and making the interpretations: first, designing of a explicitly designed traffic based router models for NoC. Then, these models used to forecast congestion possibility cases [68].

In [69] proposed quite low cost and high capacity switches of critical value in Networks-on-Chip. They suggest a very simple, memoryless move to a standard two-dimensional NoC. Packets are dispersed in an non-ideal path in the event of congestion, often called deflective routing.

In [70] an innovative structure-level buffer scheduling algorithm is introduced, which can be used to modify the NoCs router architecture. More explicitly, despite the target programme's traffic characteristics and the cumulative cost capacity of the required buffering. The proposed methodology calculates the buffer-depth in separate processor-wide routers for each input channel, to maximize overall performance. This compares strongly with the standard distribution of buffering services at present used in NoC design this can greatly reduce overall efficiency on the network. Nonetheless, the experimental outcomes demonstrate that algorithm proposed is fast, compared to the standard buffer

allocation, major performance improvements can be made. Smart buffer allocation using the proposed algorithm will achieve around 80% savings in buffering costs for a complex audio / video program.

This article [71] presents results from a comprehensive series of simulation-based experiments to compare on-chip-communication architecture performance. Such tests were performed using the *RedScarf* simulation system mentioned in the article entitled “*RedScarf: a multi-platform open-source simulation framework for Networks-on-Chip’ performance evaluation*”.

The NoC is a vital interface for other robust processor-on-chip systems. At system-level, they [72] provide a full architecture for developing and optimizing NoCs. A variety of alternative 2D mesh interconnection architectures are produced for a provided SoC by combining a collection of library with of predesigned NoC modules specified in SystemC with high level synthesis. The system support the automated network interface synthesis for translating between node specific messages and its NoC. In this paper illustrated methodology by investigating design space for two complete SoCs running convulsed functions on a high-end FPGA *field-programmable gate array* surface.

Interconnect architecture simulation can be a time-consuming aspect of on-chip multi-processor system turbulence. Accurate simulation of networks-on-chip interconnections will take many hours for practical examples of applications, and this process must be replicated with each design revision, because the interactions between design choices would have a direct effect on the overall device efficiency and latency. It introduces NoC based “*transaction-level model*” (TLM) designs that offer the data transmission mechanism’s abstract view in priority non-preemptive and preemptive NoC, which allows for a substantial reduction in the number of simulation events. The modeling models are tested using two case studies and simulated traffic with practical implementation. Results show that for the majority of flows, the lightweight TLM simulation models can deliver reliable latency figures inside pure flits packets. It achieves more than 93% reliable connection dynamic power consumption and in terms of modeling when simulating gives 2.5 to 3 magnitude orders faster [73].

In [74] proposed “minimum cost flow heuristic algorithm” (LINCA) to optimize the quantification of the hybrid routers used in many core systems, corresponding to the application traffic. LINCA guarantees the chip efficiency of hybrid networks that conform to the demands of the applications. It will reduce substantial expense by using hybrid routers on chip in the hybrid networks. This research led the first computational analysis under a number of implementations to refine hybrid networks-on-chip architectures. This research aims to design dynamically hybrid interconnections for workloads sensitive of bandwidth. The optimization approaches will exploit many core structures for better fit

problems.

Networks-on-chip (NoC) is a robust and flexible networking model used in contemporary system-on-chip architectures is considered an alternative to traditional bus systems. Therefore, comprehensive multidimensional work concerning the design and deployment of NoC based structures can be found. An elemental necessity for these operations is the existence of NoC simulators which enable different technologies to be studied and compared. In this paper addresses the review of various NoC simulators and features their contributions to work on NoC. There are numerous NoC devices are analyzed and their strengths and limitations are illustrated, such as “*NoCTweak, Noxim, Nirgam, Nostrum, BookSim, WormSim, NOCMAP and ORION*”. The comparative analysis includes techniques for measuring latency, power consumption and efficiency. [75].

In [76] a wormhole-switched NoC theoretical model is proposed based on theory of queueing to understand the delay analysis. The proposed system takes an network connectivity a topological graph, maneuvered vector and routing table in the form of matrix as its input, and calculates the mean processing time for latency of the packet in router. This functions with specific network topology under predetermined traffic conditions, with deterministic routing. This model will approximate the average latency per flow correctly and easily, thereby allowing for the exploration of various design specifications in NoC designs in a short design space. From the experimental outcomes suggests the planned computational design can estimate the mean packet-latency more than 4 times higher than an effective simulation, whereas the calculation error for various system-on-chip architectures is less than 10% in unsaturated networks.

The simulator Noxim [77] is built by way of SystemC, a system definition written library in C++. The choice behind the Noxim initiative is inspired by the elementary condition: allowing fast modifiability & flexible efficiency while promoting simulation with process accuracy. The NoC scenario implemented in the “Noxim Runtime Environment” (NRE), which includes the SystemC code to serve the different architecture of NoC. The NRE helps in understanding many topologies, router components such as buffer and packet sizes, distributions of traffic, routing algorithms, injection speeds of packets, *etc.*. Many execution statistics are produced at the end of the simulation, both in terms of output and figures.

In order to replicate the specific network situation closely, it is important to model the traffic stream transported across practical networks to validate the efficacy of the protocol designs. Extensive studies have reflected that real traffic in accessibility and LAN system interconnect (e.g., by video streams and ftp) displays the characteristics of LRD and self-similarity. In this article proposed a functional methodology with defined traffic strength for modeling self-similar traces. Since the Pareto distribution exhibits heavy-tail

properties, the values that can be produced functionally in simulations obey only a slightly different version of the distribution of Pareto, where the largest value is a finite number. It comes about with two reasons that the maximum value that the machine will produce is calculated by the fixed smallest values. This dictates that the traffic produced is often truncated to the end. Second, in simulation, the number of values generated is finite. It indicates the tail-abridge could be even worse [78]

In other natural processes, such as habitats, biological structures, and atmosphere, LRD and non-Gaussianity are universal. It is always not known, though, that the two phenomena will exist collectively in natural systems, and that self-similar property can be a quasi-particle of both physical processes in a system. Such characteristics, typical in complex systems, influence pattern identification, and severe abnormality and clustering. Two dialectical frameworks that can compensate for LRD and non-Gaussianity at the same time are discussed: “*autoregressive fractional integrated moving average (ARFIMA) and linear fractional steady motion (LFSM)*”. The mathematical properties of LRD estimators and their self-similarity are analyzed critically [79].

An analytically constructed model of simulated traffic based on the Negative Exponential Distribution (NED) for homogeneous and heterogeneous NoCs in all dimensions. This synthetic traffic profile reliably captures core statistical actions of practical traces obtained by running various applications on networks-on-chip as opposed to traditional synthetic traffic profiles. The average packet hops for the proposed traffic profile was contrasted with those of certain simulated and practical traffic trends to determine the usefulness of this current NoC traffic model [80].

This paper provides a series of nine network traffic models for NoC projects to be benchmarked. Different metrics allow equal comparison, replication of research findings, and acceleration of growth of NoC. The package is focused on actual applications contained in literature and usable on the publicly accessible Transaction-Generator (TG) benchmarking method. Average size of a task graph is noted to be about 15 tasks, except the MPEG4 encoder created with scripts. For average every assignment has just two targets, and there are tasks of atleast four goals in six graphs. There is one very active vector in four instances, i.e. a function that sends significant fractions of the total traffic, e.g. 68% for the first graph. In four instances, sinks which eject large fractions of traffic can be described. It indicates that traffic is far from standardized, and considerable variety can be found between application and within a single program. The general assumptions were made for the full program descriptions: Internal transfers are either nominal messages or copies of outgoing transfers. If there is no documentation of different activities we presume one activity per PE. When data is obtained at all sources, activities with more than one incoming channels are activated. The two-way edges are split into two unidirectional channels of

half the bandwidth. When there is no obvious device feedback, functions with no incoming channels are activated at short intervals by timers concurrently to execute the program. TG lets several activity graphs be merged to model heterogeneous behaviour. One aspect of broad MP, for example, computes telecommunication feature and the other multimedia processing. Another example is the complex shift of the telecommunications stack between UMTS and WLAN which can be accomplished with timers and few additional scheduling activities [81].

In [82] a computing model for Nostrum focused on two traffic forms, Guaranteed-Bandwidth (GUB) and Best-Effort (BEE) traffic was defined. They have suggested a common structure and exchange of infrastructure for the GUB traffic to all GUB networks. Such centralizing will be achieved by a central agent in the network, either at development time or dynamically at run-time. For BEE traffic they have recommended that each tool be assigned traffic budgets. Budget distribution may be carried out again by a central agent in the network at development time or at run-time. The network controller guarantees that the assigned allocation is fulfilled for each assets. As long as growing resource stays within its traffic budgets the BE traffic's efficiency properties are characterized. The benefit of this strategy is that all GB and BE traffic results are concisely defined by formulae, can be prepared, evaluated and written predictably.

The downside is that there is distribution of network space and might not be completely used. When a simulated circuit of GUB is not completely used, other traffic can not access the capacity, so it is lost. When a resource does not allow good use of its allotted allocation, it can not be used by other services. Various combinations and optimizations are possible in this method. At design time traffic preparation for all GUB and BEE traffic may be performed solely dynamically. Conversely, a central network administrator, e.g. a network operating system, can execute traffic reallocation dynamically at run time. If that is advantageous it relies on the program complexity and the overhead involved.

GUB traffic may also be organized on the basis of resource budgets, much like the BEE traffic definition we used. This will mean that virtual-circuits (VCs) would be opened and dynamically controlled by the communication resources GUB allocations, then locked. Likewise, channel by channel, BE traffic may be distributed, just like it was done with GUB traffic. With the four variations, *i.e.* (GUB and BEE resource-allocation or channel-allocation) shows the optimal trade-off depends on the network traffic characteristics and the run-time overheads involved.

BEE traffic classification is based on two analytical criteria, the deflection component and the traffic threshold, in the suggested theory. Because this alternative allows traffic load to spread adaptively over the entire network, Nostrum's flexible, non-minimal routing approach appears feasible. Perhaps, a deterministic routing system involves a particular

method focused on utilizing individual connections. With equivalent outcomes it was thought that is feasible but leave it to the future. The anticipate some variations, though, that will render one kind of network more suited for one application form. For MoCs with more precise output characterizations of well-defined traffic flows, deterministic-routing dependent networks are likely to require.

In systems of lesser recognized or more complex traffic flows, adaptive networks are possibly superior. It was assumed that a MoCs can be established as a result of this study that enables correct characterization of communication efficiency in NoCs. A MoC needs to allow for effective traffic planning. It attached to particular importance to MoCs which allow efficient traffic composition and analysis of the resulting system output. It was assumed this is feasible with a wide variety of routing, swapping and buffering practices, but developing network connectivity systems is necessary in order to devise effective MoCs. Not all arbitrary NoC can create a valuable abstraction of the MoC.

The NoC architecture method involves careful analysis of the program operating on it in order to maximize the efficiency and aspect of the contact services. The method of traffic modeling is the most critical phase in the characterization of dynamic applications. Three approaches can be defined for modeling traffic in the NoC research. The first method suggests that sources transmit data to the network constantly at a constant rate, which is the most widely seen. The second approach uses probabilistic features, including audio and video sources, to model the traffic activity for standard applications. This method's precision is higher, at the extra expense of difficulty and simulation period design. The third approach allows use of traffic traces to measure network efficiency. Simulation period may be prohibitive except with minimal traces. The downside is accuracy which comparable to existing ones. And if a specific protocol is appropriately designed, certain flows conflict with the conduct of the client traffic within the network.

Data about the reciprocal intervention of multiple traffic flows in NoCs are limited. This research has two objectives: compare the efficiency of NoC when various traffic models are used with the same task in terms of throughput and latency and to analyze the effect of network noise traffic on a particular model movement. Preliminary tests indicate how much the real NoC output is when an oversimplified model is utilized for a particular task. The assumption is that NoCs will use internal frameworks to ensure QoS, as noise traffic allows modeled traffic to differ from its expected behavior [83].

## 2.4 Optimization of buffers placed in MPSoC

In [84] also expanded the research with to determine the effects of self-similar traffic on MPSoCs routers buffer-size using DPPBP (Discrete Poisson Pareto Burst Process). Through contrasting theoretical boundaries and detailed simulation performance, checked

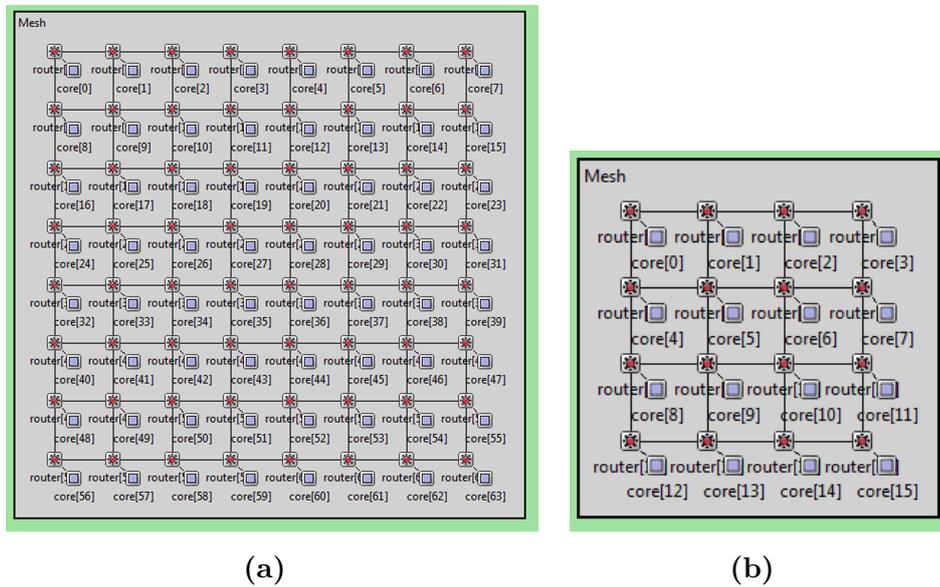
the usefulness and precision of the terms used, and the upper and lower buffer-size boundaries. For limited meshes of deterministic-routing have found that the nodes at the boundaries become bottleneck for terms of buffer-size owing to their own LRD traffic overloads. As a supplementary consequence, demonstrated the benefit of growing the channel operation rate by rising the degree of self-similarity that renders the buffer allocation algorithm less complicated to a more consistent buffer behaviour. This is an significant finding in NoC architecture, because the alternate approach that reduces the size of the router buffer is impractical and raises the NoC packet latency considerably. They need to develop creative strategies such as congestion management to mitigate the crippling impact of self-similar traffic due to the strong volatility of LRD traffic.

In this paper [85] established an overview to the worst-case buffer requires study in time-constrained applications built for NoCs. Based on this study, two problem solved related to: reducing the overall buffer-size needed for application specific NoCs and information synthesis with buffer-size constraints applied. In all situations it ensure that the deadlines set and the requirements for the likelihood of delivery messages are achieved. It also claimed that traffic shaping is a effective tool for reducing buffer capacity. They suggested 2 powerful greedy sub-problems of connectivity planning and traffic shaping and reported experimental findings demonstrating the efficacy of the method.

---

## Synthetic Self-similar Traces based Optimization of Buffer-Size for Different Traffics for Networks-on-Chip

---



**Fig. 3.1** (a) 64 nodes Mesh Topology (b) 16 nodes Mesh interconnection network by OMNET++

### 3.1 Traffic Distribution Patterns

The multicore system architectures work are assessed on the 3 traffic distribution patterns to understand distribution of packets under space using metrics such as *latency*, *nodes*

*hop-count etc.* from source to destination. The selection of source-destination pair in *uniform traffic* [7] is determined by inserting the random function provide a number picked using a uniform distribution range between 1 and  $m - 1$ , where  $m$  is the topology size to each of the source core-id which ranges from 1 to  $m$ .

$$d_i = (s_i + int\_uniform(1, (\lceil m/2 \rceil - 1))) \text{ mod } m \quad (3.1)$$

where  $d_i$  and  $s_i$  are the terminal and origin core-id respectively, The function *int\_uniform* returns uniform distributed integer number created from the range  $[1, n - 1]$ . The direction most probable deliver to end destination balanced equally with very low load balancing using uniform traffic. The set of source-destination core-id in *tornado traffic* [7] is where the origin cores are added with  $\frac{m}{2}$  the network-size  $m$ .

$$d_i = s_i + (\lceil m/2 \rceil - 1) \text{ mod } m \quad (3.2)$$

The specified source-destination core-id pair is in the *complement traffic* [7] in which the terminal core focuses on the source core.

$$d_i = \neg s_i \text{ mod } m \quad (3.3)$$

## 3.2 Experimental Results

All 3 network trends won't offer the same core-id origin and terminal pair, that is the core that produces the data-packets would not return to its own core. That is the explanation for the option of choosing these 3 traffics, because the center does not transmit the packets to itself in the multicore architecture. The multiple cores have specific tasks to do, such that the intermediate effects are passed to other cores after processing the packets. For example, in multimedia device processing, encoding, decoding and shared functions are delegated to specific architecture cores utilizing certain specialized mapping algorithms. Such key perform the tasks and connect to other key for full application delivery. So we took these 3 traffic distribution to test the efficiency of the synthetic-traces on interconnection based many core architecture.

We took 2 instances of mesh topology for the results one is 64 nodes square topology in Fig. 3.1a and the other is 16 nodes square topology in Fig. 3.1b. The simulation framework is performed on the OMNET++ [86], where the simulation specifications are seen in the Table 5.5. In order to change the buffer-size, the adjustment is made in the maximum number of queued packets, where 8, 16, and 32 are allocated for the buffer sizes of 256, 512 and 102 bytes.

Within the Figs. 3.2 and 3.3, the average hop-distances for each center are plotted, the plot indicates identical trend of distance variance with specific traffic trends in both architectures. The overall mean hop-count for *Uniform-traffic* for 64 *nodes* mesh topology is 10 for core number 7, while the overall mean hop-count for *Tornado-traffic* and *Complement-traffic* is 15 for core number 1 and 8 respectively, and the max average hop-count for *Uniform-traffic* is 45 for core number 7 for 64 *nodes* mesh topology, while the actual average hop-count is 45 for core number 7 respectively.

In each architecture the mean end-to-end latency is plotted in the Fig. 3.4 and Fig. 3.5 for all three flow. The end-to-end latency is the period of time the packet takes from origin to terminal. For increasing traffic distribution the distribution of latency is very common across the 3 specific buffer-sizes. The analysis is performed on the 3 traffic conditions on *Uniform-traffic*, *Tornado-traffic* and *Complement-traffic*, in 3 buffer-sizes of 256, 512 and 1024 bytes.

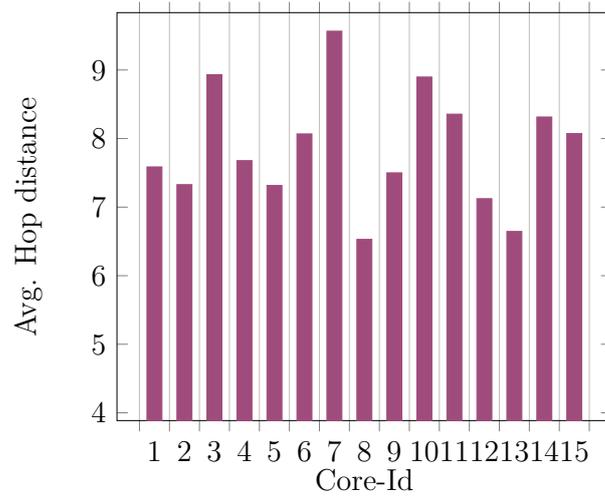
$$\psi_{e-to-e} = M [\psi_{trn} + \psi_{pro} + \psi_{pro} + \psi_q], \quad (3.4)$$

where  $\psi_{trn}$  is the delay in transmission,  $\psi_{pro}$  is the delay in processing,  $\psi_{pro}$  is the delay in propagation,  $\psi_q$  is the delay in queuing  $M$  is number of connections, which is only *number of routers*. The latencies for specific traffic reveal that the *complement-traffic* has higher latency and differs because the variance of the source-destination pair, while the latency for *tornado-traffic* is quite smooth and variable parameter very minimal, the variability of the source-destination pair is limited and thus the latency is weak, while for *Uniform-traffic* is zig-zag.

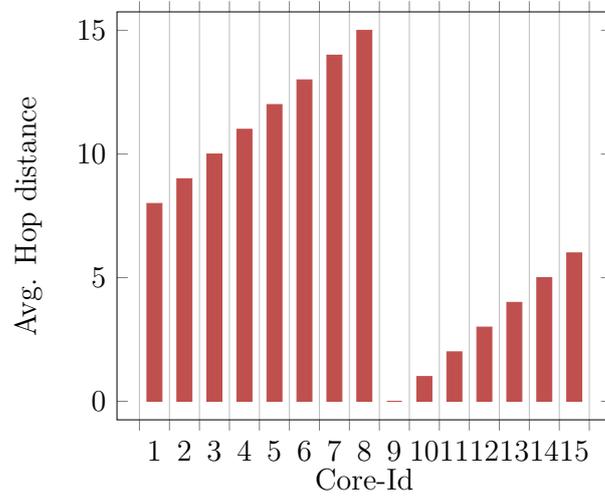
### 3.3 Conclusion

We provided a comparison of hop lengths, latency delay of end-to-end and the likelihood of packet-loss for the three traffic patterns Uniform, Tornado & Complement for 2 mesh topology architectures  $4 \times 4$  &  $8 \times 8$ . Our multicore architecture design builds on The self-similar traces created for multimedia applications are the statistical properties of the video systems are synthetically used. The versatile nature of trace produced in synthetics helps in a simple and precise simulation which is not feasible for application in real time.

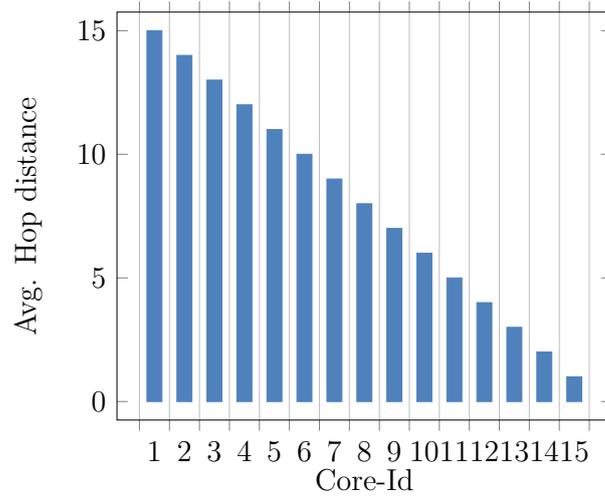
The above determined parameters assist in the early design of architecture and the selection of suitable choices resource connectivity. The probability of failure highlights the existence of buffer under various network condition systems and an optimal use of buffer makes them energy-saving by not losing packets and effective architecture by saving unexploited extra room for buffering.



(a)

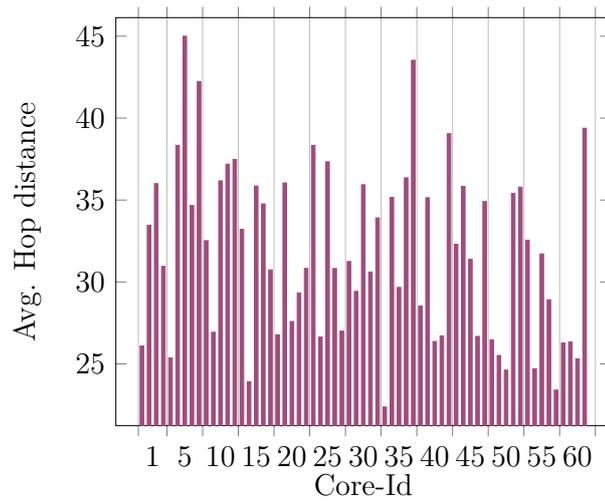


(b)

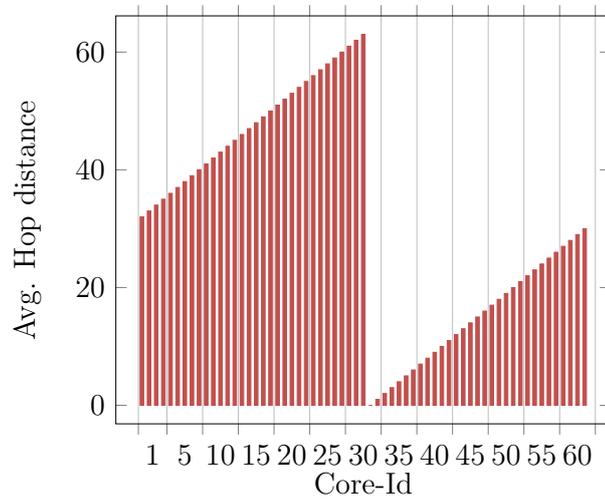


(c)

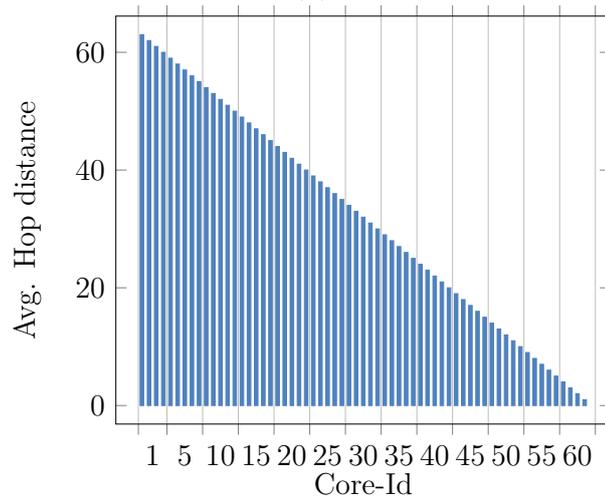
**Fig. 3.2**  $4 \times 4$  Mesh interconnection network's avg. hop-count for the (a) Uniform-traffic (b) Tornado-traffic (c) Complement-traffic



(a)

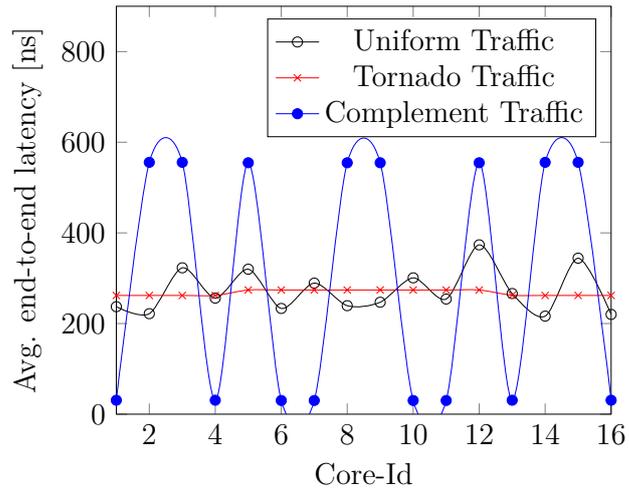


(b)

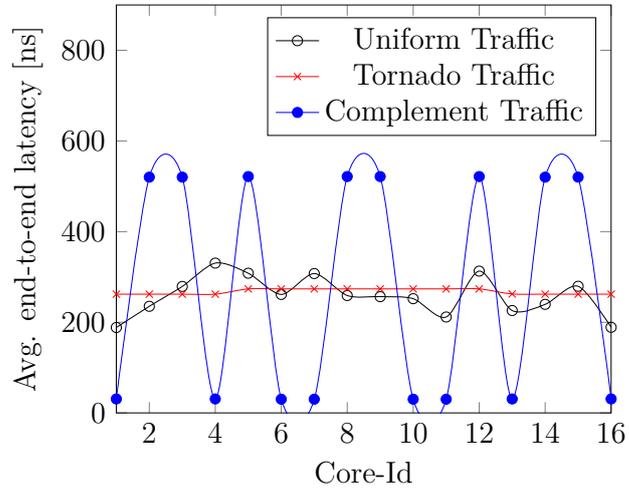


(c)

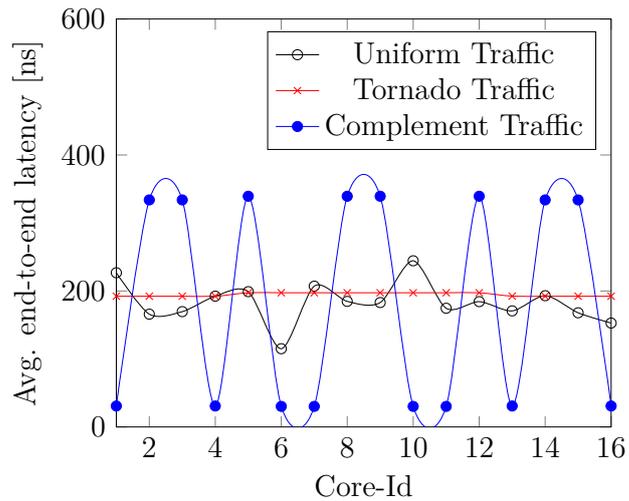
**Fig. 3.3**  $8 \times 8$  Mesh interconnection network's avg. hop-count for the traffic (a) Uniform-traffic (b) Tornado-traffic (c) Complement-traffic



(a)

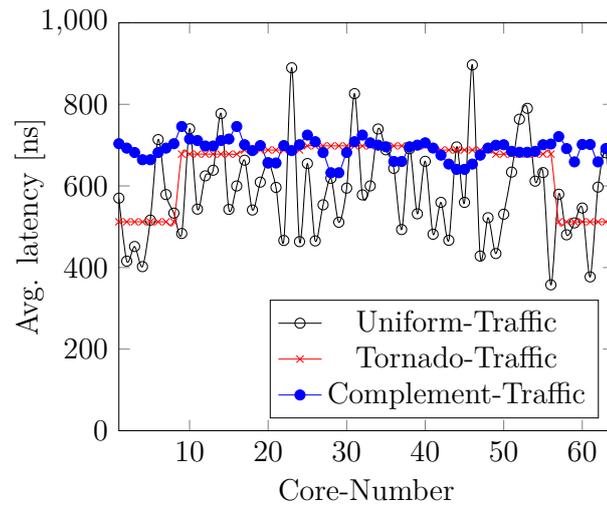


(b)

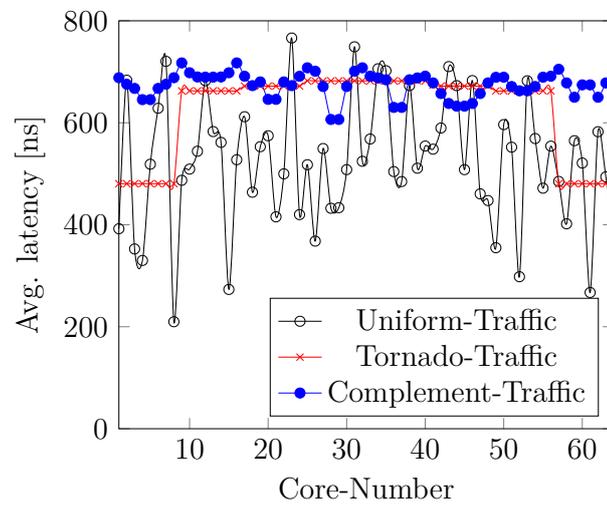


(c)

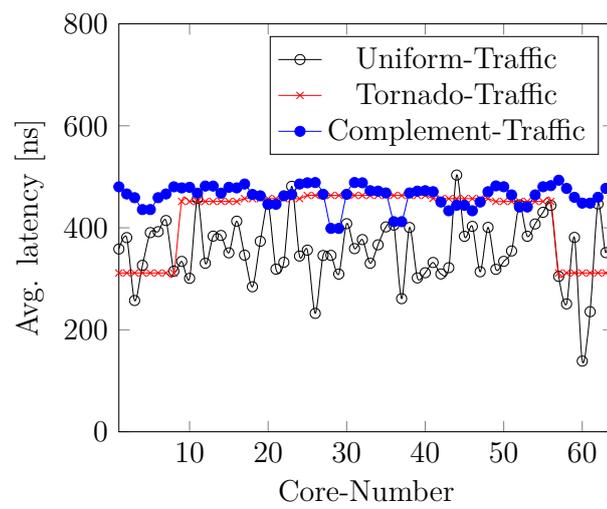
**Fig. 3.4**  $4 \times 4$  Mesh interconnection network's avg. end-to-end latency for distinct traffic for buffer-size (a) 1024 bits (b) 512 bits (c) 256 bits



(a)



(b)



(c)

**Fig. 3.5**  $8 \times 8$  Mesh interconnection network's avg. end-to-end latency for distinct traffic for buffer-size (a) 1024 bits (b) 512 bits (c) 256 bits

---

## Energy Efficient Application Traffic Modeling and Synthesis for MPEG4 based on networks-on-chip architecture

---

In this chapter the synthetic traffic is generated using the statistical properties of raw videos. The fast algorithm is designed for the generation of synthetic traffic with running cost of almost  $O(n \log n)$ . In order to validate the traffic is passed from the Norro's technique of loss-probability considering infinite buffer.

### 4.1 Long-Range Process Dependency

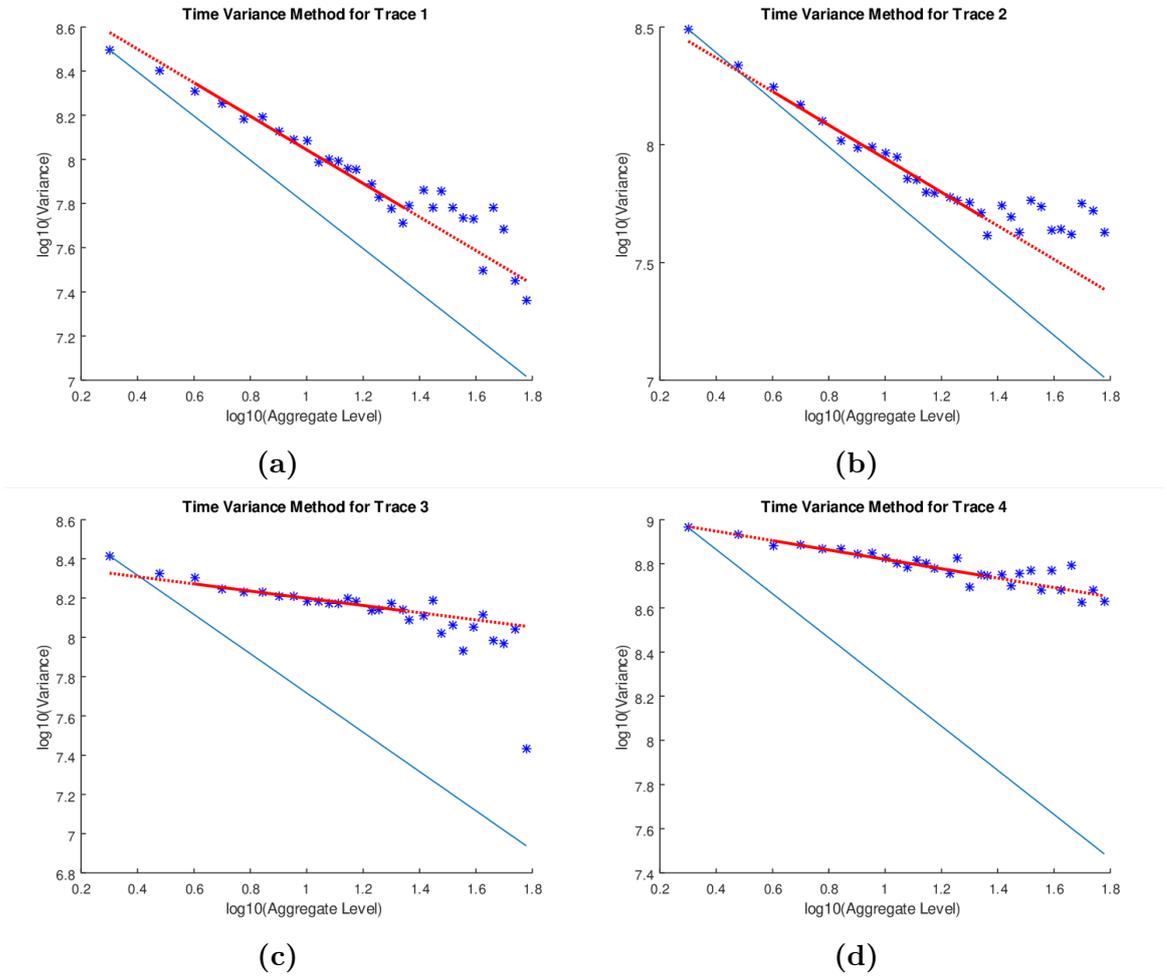
The stationary *LRD* process are stochastic process. In order to define *LRD* processes are represented by the equation as: let  $\mathcal{W} = (\mathcal{W}_i : i = 0, 1, \dots)$  with average as  $\mu$ , variance as  $\sigma^2$  and auto-correlation function represented as  $a\_corr(r)$  where  $r \geq 0$ .  $\mathcal{W}$  is known as to long-range process [87] with dependency if

$$a\_corr(\kappa) \sim r^{-\beta} \mathcal{X}_1(t) \quad \text{as } r \rightarrow \infty \quad (4.1)$$

where varying function  $\mathcal{X}_1(t)$  that gradually diverge. The exponent's representation as  $\beta$  ranges from  $(0, 1)$ . The equation  $\lim_{t \rightarrow \infty} \mathcal{X}_1(tz)/\mathcal{X}_1(t)$ , is equal to 1 for all  $z > 0$  and  $\sim$  refers to asymptotic proximity.

When hyperbolic existence of autocorrelation function decay which is found polynomial decay can be seen from Eq. (4.1). Not like the exponential decay that is close to the

conventional Markov process, or *SRD*. For any time series  $\mathcal{W}$  we chose three methods to check the existence of LRD. The first two methods are using the graphical approach while the 3<sup>rd</sup> method is established on a periodogram. For the *long range dependent* (LRD) processes, the autocorrelation function on one axis and the lags number on another axis decreases very gradually, which is expressed by polynomial function.



**Fig. 4.1** Variance-Time representation for four Synthetic Traces (a) 1 (b) 2 (c) 3 (d) 4

## 4.2 Various Methods to calculate Hurst Parameter

Here various algorithm are discussed which are used for the calculation of Hurst exponent. There are limitations and advantages of each method discussed in the sections.

### 4.2.1 Variance-Time Method

Suppose  $\mathcal{V}$  a fixed time-series in stochastic form. Consider  $\mathcal{V}^q = \mathcal{V}_k^q$  developed by captivating the average over seminal time-series with  $k : k, q = 1, 2, \dots$ , denote the  $\mathcal{V}$  stochastic time-series with stationary increments over segments block-size of  $q$ , where every series of

segments does not overlap. That's represented by Eq. (4.2).

$$\mathcal{V}_k^q = \left(\frac{1}{q}\right) (\mathcal{V}_{kq-q+1} + \dots + \mathcal{V}_{kq}), \quad k > 0 \quad (4.2)$$

In variance-time method time series are plotted using log-log plots  $\mathcal{V}^q$  against  $q = 1, 2 \dots$  whose slope is negative and linear with magnitude equal against  $m$  for the *short-range dependent* processes. The nature of negative slope decreases slowly as value of  $m$  increases. The behaviour of *long-range dependent* variance-time method series plot is linear negative slope for large values of  $q$  with magnitude equal to 1 in log-log graphs, but the slope rather linear as it can be seen by the plotting of  $\log(\text{var}(\mathcal{V}^q))$  against  $\log(q)$  it is much more flatter. Thus the divergence  $\mathcal{V}^q$  of the time-series and its definition is commensurate to the auto-correlation's definition [88], which is defined as

$$\phi(\mathcal{V}^{(q)}) \sim cq^{-\alpha} \quad \text{as } q \rightarrow \infty \quad (4.3)$$

where  $\alpha \in [0, 1]$ , is interpreted as to auto-correlation stipulation is in Eq. (4.1), the magnitude of  $\alpha$  is the auto-correlation function. The nature of decay is hyperbolic function as the number of lag becomes larger. The plot from variance-time method shown in Fig. 4.1, for the different series. The red spots are accumulated series variation when the red line is projected on these lines, the  $\alpha$  is the equipped line's negative slope, thus *Hurst Parameter* from Eq. (4.4).

$$\mathcal{H} = 1 - \frac{\alpha}{2} \quad (4.4)$$

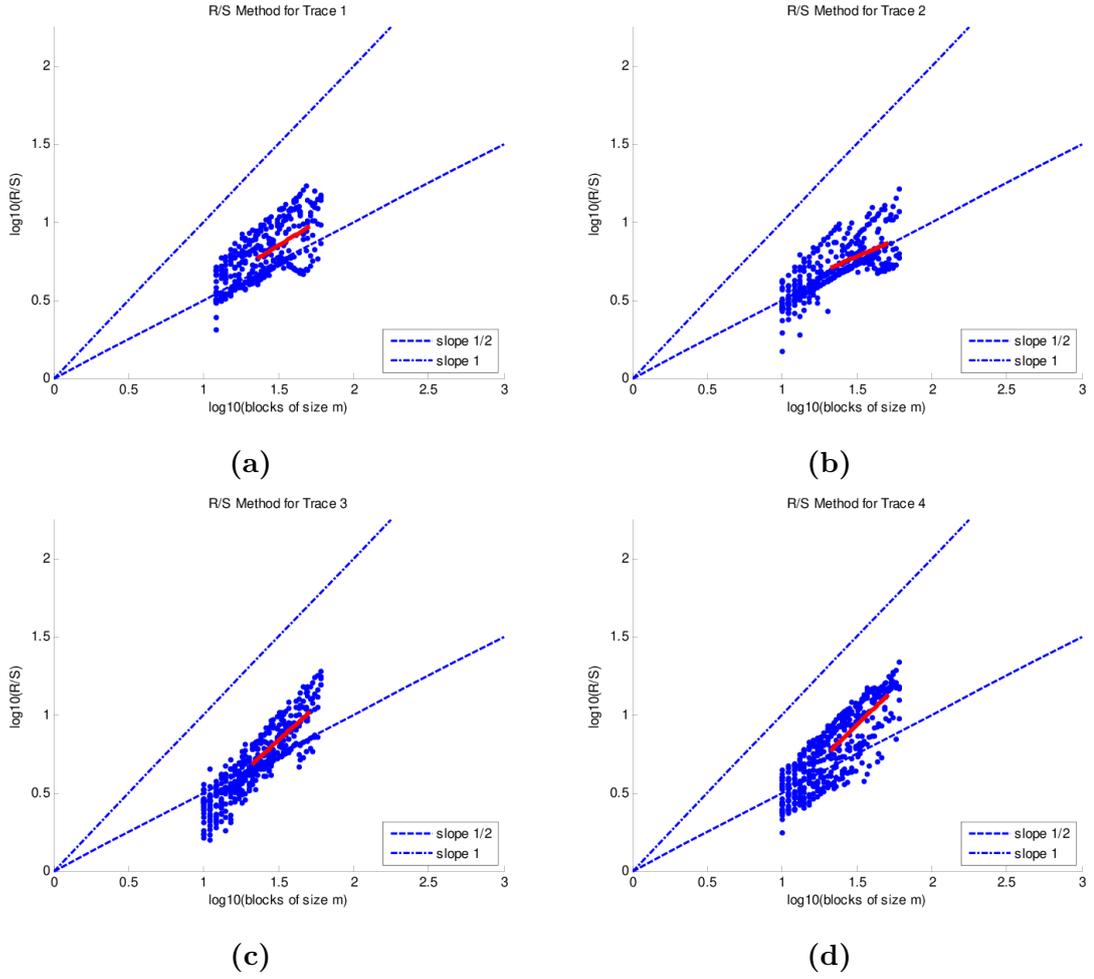
### 4.2.2 Rescaled Adjusted Range Analysis

The time series variability of the data is calculated by statistical based method to measure *Hurst exponent* [89], [90]. The principle behind to measure this value is the proportion between numerical mean of time-series range, and the standard deviation. The log-log curve of *rescaled range* with the amount of data is evaluated in this process, the plot slope is interpreted as *Hurst Exponent*.

Let us consider the observed time-series ( $\mathcal{X}_n : n = 1, \dots, k$ ) with its average equal to  $\mu(k)$  and variance as  $\phi^2(k)$  this statistic of *rescaled adjusted range* (represented as *RS* statistics). The equation to represent *RS* statistics is given in Eq. (4.5)

$$\frac{R(k)}{S(k)} = \frac{1}{S(k)} [\text{Max}(0, \mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_k) - \text{Min}(0, \mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_k)] \quad (4.5)$$

where  $\mathcal{X}_n = (\mathcal{W}_1, \mathcal{W}_2, \dots, \mathcal{W}_n) - n\mu(k)$ , where the range of  $n$  is  $1 \leq n \leq k$ . From the [51]



**Fig. 4.2** RS method representation for four Synthetic Traces (a) 1 (b) 2 (c) 3 (d) 4

we find that the *Hurst parameter* has historically existed in many documents which is shown by

$$\mathbb{E} \left\{ \frac{R(k)}{S(k)} \right\} \sim ck^H, \quad \text{as } k \rightarrow \infty \quad (4.6)$$

When the near value of  $H = \frac{1}{2}$ , therefore  $\mathcal{W}_n$  would be *pure Gaussian process*, this process will be the part of *short-range dependent process*. If  $H > \frac{1}{2}$ , then the  $\mathcal{W}_n$  process would be *long-range dependent process* and that is called the *Hurst impact*. The relationship is given by Eq. (4.4) between the Hurst parameter and the rate at which the auto-correlation function decays.

The *RS* method for measuring exponent Hurst plotted in Fig. 4.2 for the four images. The *Hurst parameter* can be interpreted by line's slope. The non-overlapping blocks  $m$  and  $\left[\frac{R}{S}\right]_m$  are represented as dotted blue dots, whereas the red sloped line is interpreted as *least-square fitted projection* over these spots of dot.

**Table 4.1** Hurst Exponent Using 3 Methods

Video Files	R-S Plotting Method	Variance_Time Method	Whittle's Estimation Method
<i>Appolo_Sixteen</i>	0.5924	0.6170	0.6450
<i>Change_Values</i>	0.4161	0.6438	0.5862
<i>Quicktime_Play</i>	0.8670	0.9085	0.6436
<i>Media_Power</i>	0.9294	0.8940	0.7818

Consider the process time series  $(\mathcal{W}_n : n = 1, 2, \dots, k)$ . The time series divide the process into  $k$ -block sizes. The non-overlapping structure of blocks computed for each of the generated new starting points the rescaled range, when  $t_1 = 1$ ,  $t_2 = \frac{N}{K} + 1$ , and  $t_3 = 2(\frac{N}{K}) + 1, \dots$  satisfies the condition  $(t_1 - 1) + s \leq N$  where  $s$  is the block-size. There are conditions to keep the value of  $s$  appropriate, such as for greater number of values keep  $s$  small and for large value of  $s$  we will get tinier no. of values for RS method with  $\mathcal{X}_{t_i+1} - \mathcal{X}_n$  and  $\phi^2(t_i, s)$  is the variance of the sample of  $\mathcal{W}_{t_i+1}, \mathcal{W}_{t_i+2}, \dots, \mathcal{W}_{t_i+s}$ . The model can be found on the [48].

### 4.2.3 Whittle's Estimation

The *Whittle's Estimation* theory is *semi-parametric technique* established on *periodogram* to approximate the Hurst exponent [91],[92], [93]. The method approximation of spectral density  $\{(\lambda)\}$  utilizes to approximate the Hurst exponent.

$$f_{c,H}(\lambda) = d\lambda^{1-2H} \quad (4.7)$$

for frequencies  $\lambda$ . The time-series  $(\mathcal{X}_k : k = 1, \dots, n)$  for the process is shown in Eq. (4.8)

$$\mathcal{I}_N(\lambda) = \frac{1}{2\pi n} \left| \sum_{t=1}^N \mathcal{X}_t e^{i\lambda t} \right|^2 \quad (4.8)$$

where frequencies  $\lambda$  are related to Fourier-like  $\lambda_{j,N} = \frac{2\pi j}{N}$ ,  $0 \leq j \leq [N/2]$ . The estimate of *Hurst Paramater*  $\mathcal{H}_{WE}(m)$  is estimated by the equation Eq. (4.9) with  $c$  and  $H$ , with  $f_{c,H}$ .

$$\sum_{j=1}^m \log f_{c,H}(\lambda_{j,N}) + \frac{\mathcal{I}_N(\lambda_{j,N})}{f_{c,H}(\lambda_{j,N})} \quad (4.9)$$

### 4.3 Synthetic Traffic Generation

The *Hurst exponent* stipulate video file properties, helps to develop fragments of shorter data process, and speeds up simulation by magnitude order. We need to consider the statistical features of the basic video before generating synthetic traces first. First and Second order statistics are important characteristics for determining the storage pattern for the buffer. The conservation of the initial traffic *Hurst parameter* is necessary to catch the gradually decaying long-range auto-correlation property.

We define the method Circular Embedding Technique Method (CIET) for generating the *fractional-Gaussian-noise* (FGN) process, the runtime complexity of which is  $O(n \log n)$ , which we define as the first method for generating the *fractional Gaussian noise*(FGN) process [94]. The fast approximation technique operation used in *Varatkar et. al.* to produce the FGN spectrum using function of sine, cosine and gamma. The FGN is studied with the closed form statement of  $f(\lambda, H)$  in [51], having approximation to the less complex articulation in [95]. CIET works within the range of  $[0.5, 1]$  in Hurst parameter. First, by using the Hurst parameter ( $H$ ) Eq. (4.10) to measure the covariance ( $c_{fgn}$ ) of  $n$  results, the next step is to embed the array of covariance with the first the other way making the length  $2k$ . Extricate the real positive element of a circular embedding sequence after taking the *Fast Fourier Transform* (FFT).

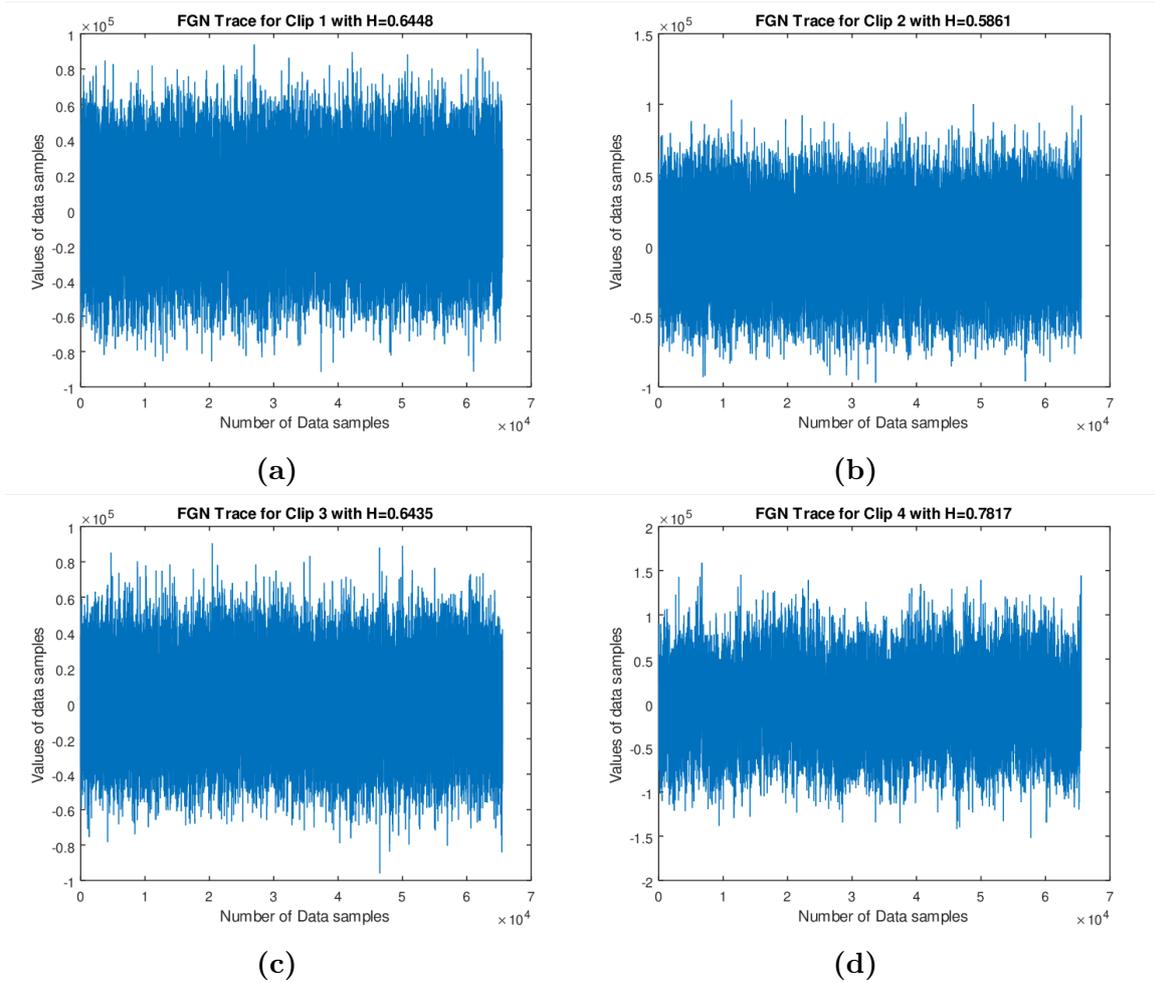
$$c_{fgn} = \frac{\sigma^2}{2} \left( |n-1|^{2H} - 2|n|^{2H} + \frac{|n+1|^{2H}}{2} \right) \quad (4.10)$$

where  $n$  ranges from  $[0, k-1]$  and  $\sigma$  the standard-deviation.

Within the Fig. 4.4 displays the steps to produce the FGN method using CIET. The 1<sup>st</sup> step is to create random complex-numbers of length  $k$ . The next step to take  $\sqrt{2}$  normalizing the first and last value only. Then embedded the series by finding the conjugate of complex\_numbers to render the array-size to  $2k$ . From the previous step's output take the complex array of the  $\sqrt{n}$  of the scale  $2k$ -array is normalized.

The final procedure to remove  $n$  data parameters from the complex-array before taking the real part of the FFT and normalizing it by  $\sqrt{4k}$ . The resulting  $k$  size process is the FGN process provided from CIET algorithm. The algorithm 4.1 and algorithm 4.2 takes the complexity of  $O(n \log n)$ , as the upper asymptotically bound for the FFT and the IFFT respectively. The linear dimension's length of the generated time-series is  $n$  with the total runtime complexity is constrained to  $O(n \log n)$ .

From Fig. 4.5 displays method of generating synthetic traffic or trace's length  $k$ , and with the help of  $H$ , the key ingredient to produce the scattered-spectrum at equivalent distance of  $k/2$  points in the range of  $(0, \pi)$  frequency-domain. The resultant spectrum consists of



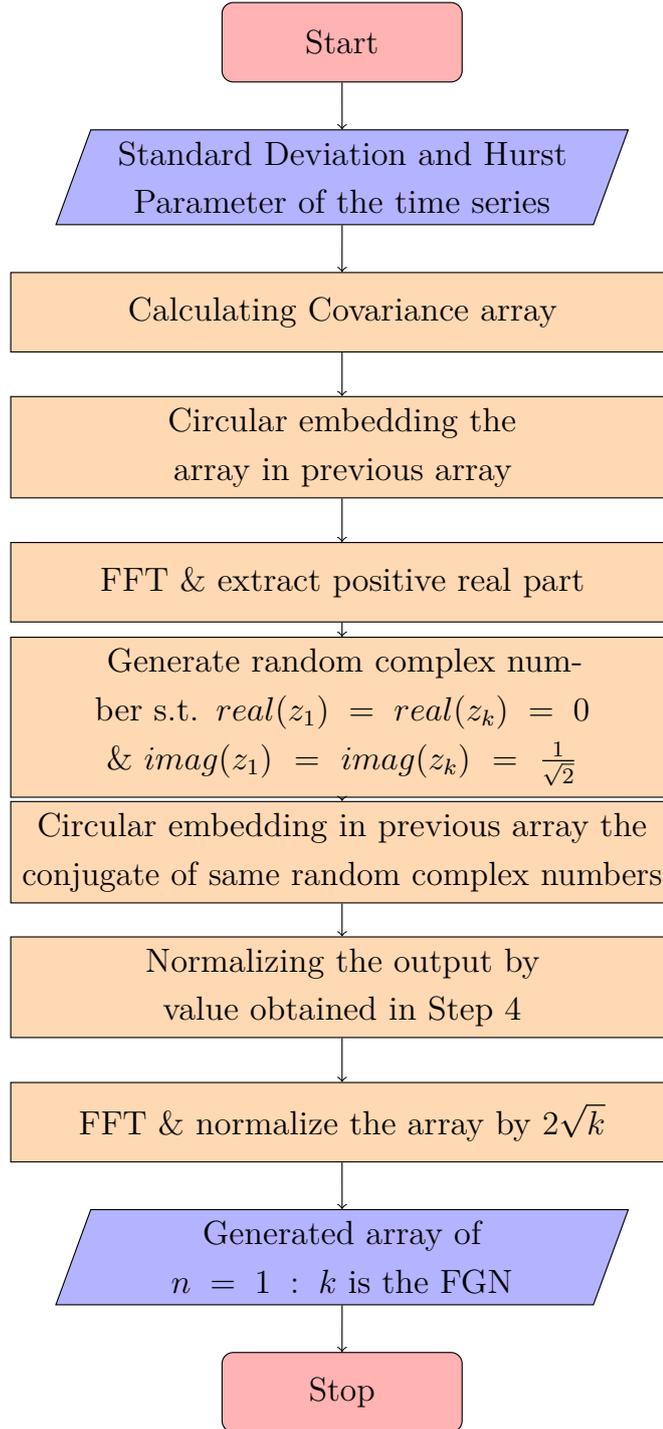
**Fig. 4.3** Fractional Gaussian Noise (FGN) for Synthetic Traffic (a) 1 (b) 2 (c) 3 (d) 4

complex-numbers. The sample path based on frequency-domain formed and the dynamic conjugates generating sample path completes the chain.

In Fig. 4.3 displays the generated FGN process using CIET procedure with the number of time series samples are produced by the statistical properties of the initial clips on the  $x$ -axis, while the  $y$ -axis indicates the statistical complexity of sample produced for the four separate clips.

Time series process models are assessed through the exponentially decaying functionality of their auto-correlation function, which earned the dynamic existence from inherent network-properties, the overwhelming process of time-series implementations gradually decay their auto-correlation function, which is why they are now called *LRD* when the former is called *SRD*. Such *LRD* provide a special division of self-similar process described by the word *Hurst Exponent* ( $H$ ) span from  $[0, 1]$ .

The packet loss-probability of the on-chip network's considering infinite buffer is measured



**Fig. 4.4** Fractional Gaussian Noise generation procedure

in [48]. But for the on-chip network, the finite buffer isn't determined. Although the loss likelihood analysis in [48] isn't as detailed. Multiplexing is such an aspect which can't be ignored, in order to serve various real-time applications for the on-chip infrastructure in higher speed networks where multiple devices share separate network resources, for example routers, bandwidth and buffer space, thus improving network performance.

---

**Algorithm 4.1: Algorithm for the generation of Fractional Gaussian Noise Using Circular Embedding Technique**

---

**Input:** Hurst Parameter, Standard Deviation of the Time Series

**Output:** Fractional Gaussian Noise

- 1 Calculate the covariance  $c_{fgn}$  using Hurst parameter  $H$  and standard deviation  $\sigma$ .

$$c_{fgn} = \sigma^2/2(|m-1|^{2H} - 2|m|^{2H} + |m+1|^{2H}/2), \quad (4.11)$$

where  $m = 0 : k - 1$ .

- 2 Concatenate the  $c_{fgn}$  by taking the elements of the same array and circularly embedding it making the size of the array as  $2k$  lets call it  $v_{fgn}$ .
  - 3 Extract the positive real part of *Fast Fourier Transform* of  $v_{fgn}$  lets call it  $g_{fgn}$ .
  - 4 Generate the random complex numbers  $z_1, z_2, \dots, z_k$  whose imaginary part of first and last element are zero and real part of first and last element is normalized by  $\sqrt{2}$  of the random array.
  - 5 Create an array by taking the conjugate of the same array  $(z_1, z_2, \dots, z_k)$  and circularly embedding it making the size of the array as  $2k$  lets call it  $conj_{fgn}$ .
  - 6 Normalize the  $conj_{fgn}$  array by  $g_{fgn}$  call it  $y_{fgn}$ .
  - 7 Calculate the *Fast Fourier Transform* of  $y_{fgn}$  and extracts the real part of it and normalize by  $\sqrt{4k}$  call it  $f_{fgn}$ .
  - 8 Generated array of  $f_{fgn}(m)$  where  $m = 1 : k$  are the resulted *Fractional Gaussian Noise*.
- 

More precisely, the encoder-decoder are the simple block functions of MPEG, in the 1<sup>st</sup> stage of encoder part which performs the coarse motion estimation from frame to frame. In the 2<sup>nd</sup> stage executes the comminuted motion estimation to accomplish data-compression. The quantization and discrete-cosine-transform (DCT) are performed in the 3<sup>rd</sup> stage. In comparison, from the decoding portion, the variable-length decoding of each data. The stagnant dataflow in the 1<sup>st</sup> step, conducts the second step subsequent inverse\_c cosine\_transformation (IDCT). The inverse quantization and motion compensation are performed in the 3<sup>rd</sup> stage. Finally, the final stage reconstruction.

The core portion that implements the encoding is shown in Fig. 4.6 with a red backdrop, while the blue is used for decoding. The green color indicates the encoder and the decoder jointly exchanging cores. In Table 4.3 shows different processes applying on the cores. When encoding the VLD, MEMD and ISC cores are isolated from the OVM and DB cores since no consistent information flow between the cores, they connect collectively with the aid of REC.

---

**Algorithm 4.2: Algorithm for the Trace generation**


---

**Input:** Hurst Parameter, Fractional Gaussian Noise

**Output:** Stochastic Process OR Trace

- 1 The progression of elements  $x_1, x_2, \dots, x_k$  where  $x_j = \tilde{x}(2j\pi/k; H)$  is constructed analogous to the sphere of an *Fractional Gaussian Noise (FGN)* process for frequencies ranging between  $2\pi/k$  to  $\pi$ .
- 2 Independently multiply each term of  $x_j$  with an exponential variable which is chosen randomly with mean is equal to 1 call it  $\hat{x}$ .
- 3 Construct  $(z_1, z_2, \dots, z_k)$ , complex number sequence so that  $|z_i| = \sqrt{\hat{x}}$  and the state of  $z_i$ , is uniformly dispersed in the range 0 and  $2\pi$ .

$$\mathbf{z}'_{\mathbf{g}} = \begin{cases} 0, & \text{if } g = 0. \\ z_g, & \text{if } 0 < g < k/2. \\ \bar{z}_{k-g}, & \text{if } n/2 < g < k. \end{cases} \quad (4.12)$$

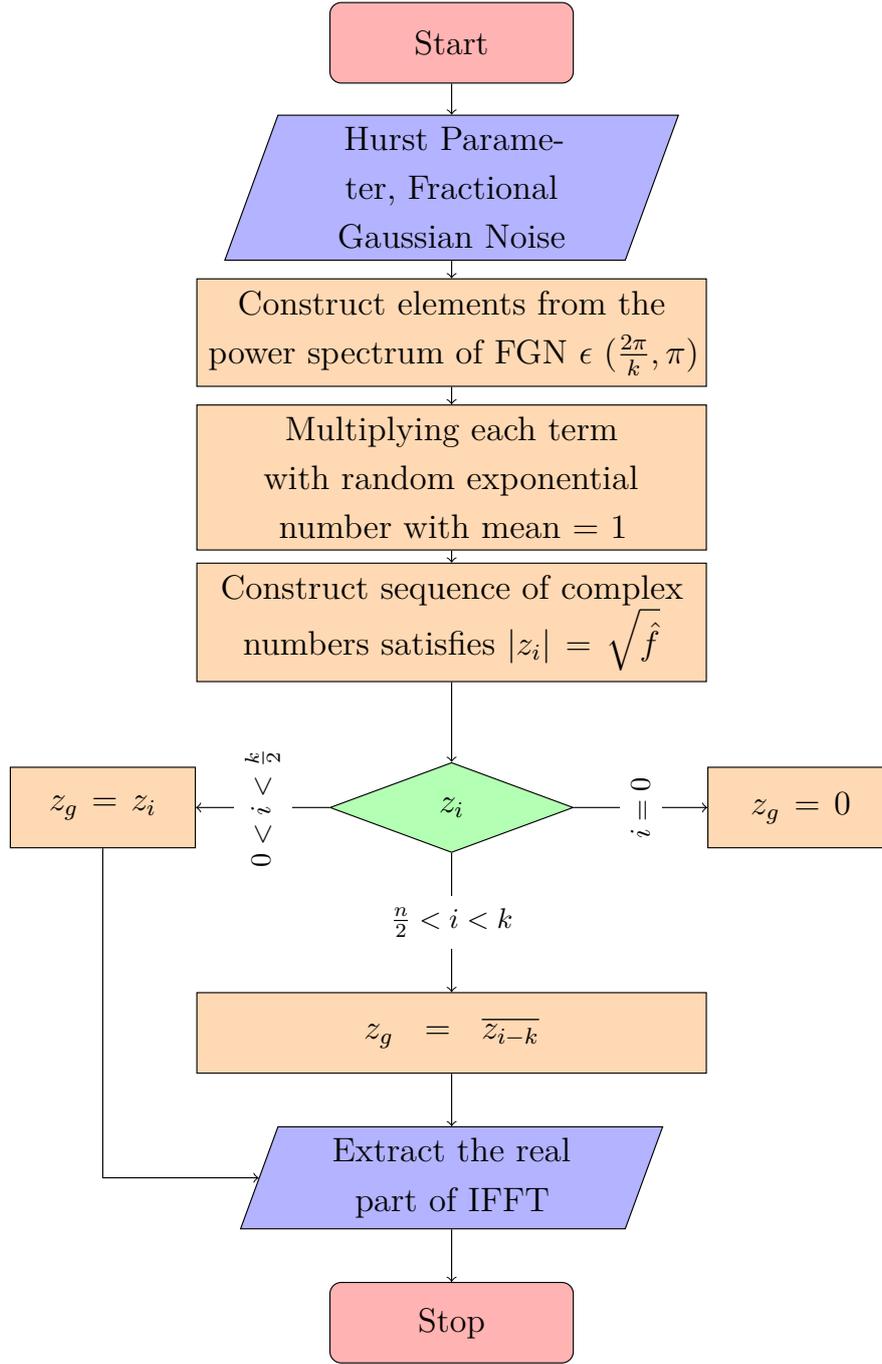
- 4 Extract the real part of *Inverse-Fourier Transform* of  $\mathbf{z}'_{\mathbf{g}}$ .
- 

**Table 4.2** Cores Implementing the Processes

<i>core</i>	<i>Description</i>
MEC	Motion estimator coarse
MEG	Motion estimator granular
MC	Motion compensation
MEMD	Decoder memory
MEME	Encoder memory
VLC	Variable length encoder
VLD	Variable length decoder
IVM	Input video module
OVM	Output video module
REC	Reconstruction
SC	Stream Creator
DB	Deblocking
QDCT	Quantization and discrete cosine transform
IQIDCT	Inverse quantization and inverse discrete cosine transform
RISC	32 bit RISC microprocessor
ISC	Input stream controller

## 4.4 Packet Buffer-Loss Probability

The probability of packet-loss in communication-protocol is an imperative Quality of Service (*QoS*) standard. In case the chances of overflow for a boundless buffer is found in [96],



**Fig. 4.5** Synthetic Trace generation procedure

[97]. The analysis for the bounded buffer is found in [98], [99]. The uncertain buffer-loss  $P_L(z)$ , for the bounded buffer-size of  $z$ , measures the percentage of data-packets missing to the packets that feed inside the router.

$$P_L(z) = \lim_{n \rightarrow \infty} \frac{\sum_{k=1}^N (Q_{k-1} + \lambda_k - c - z)^+}{\sum_{k=1}^N \lambda_k} \quad (4.13)$$

where  $(z)^+$  corresponds to maximum of either of  $(z, 0)$ . The *Tail\_probability* can be

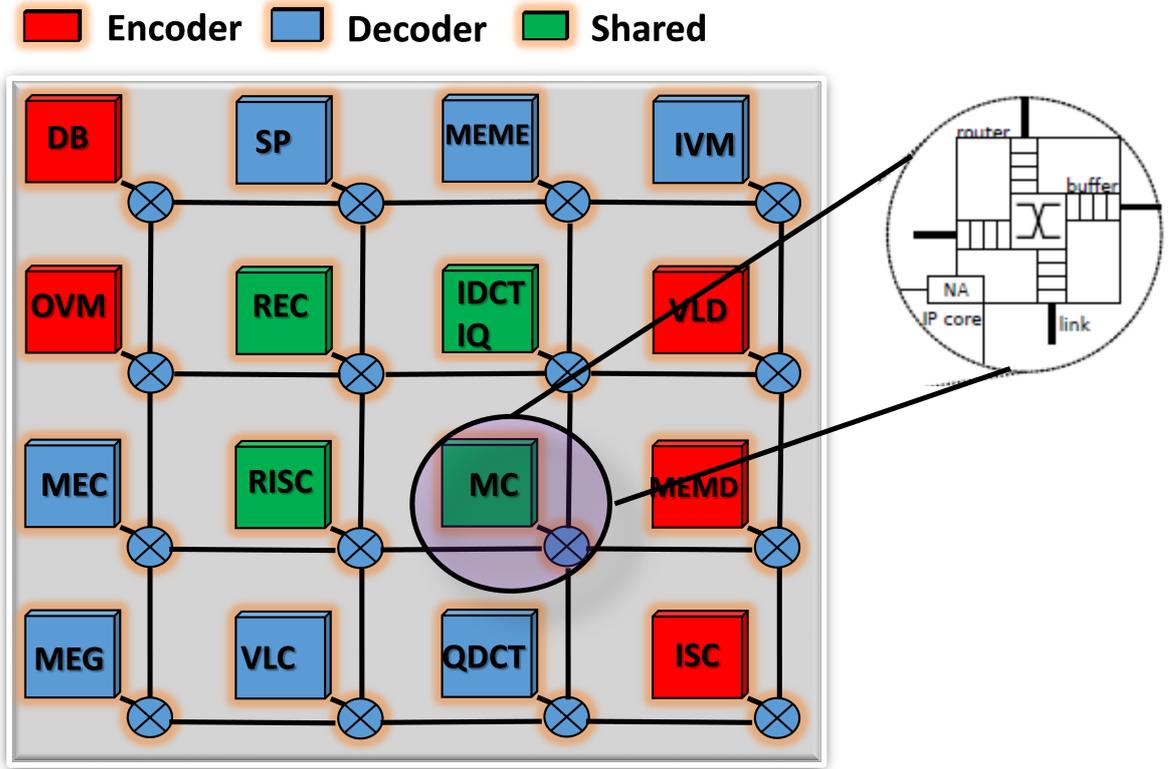


Fig. 4.6 Application mapping of the functions for the MPEG4

expressed by the expression  $P\{Q > z\}$  is portrayed by the amount of data-packets spent above point  $z$  by the cumulative time inside the gratuitous buffer space. The probability of packet-loss ( $P_L(z)$ ) is usually estimated to the probability of a tail given by

$$P\{Q > z\} \text{ as } P_L(z) \approx P\{Q > z\} \text{ i.e. } \frac{P\{Q > z\}}{P_L(z)} \quad (4.14)$$

subsidiary to a variable constant as  $z \rightarrow \infty$ .

So the likelihood of failure can be determined as

$$P_L(z) = \frac{P_L(a)}{\mathcal{P}\{Q > a\}} \mathcal{P}\{Q > z\} \quad (4.15)$$

The Lindley's equation based on the length of queue  $Q_n$  at time  $n$  can be represented by

$$Q_n = (Q_{n-1} + \lambda_n - c)^+. \quad (4.16)$$

the process can be defined as

$$Z_n = \sum_{k=1}^n \lambda_k - cn. \quad (4.17)$$

The process in Eq. (4.17) is stochastic process. Every  $x > 0$  the generalized variance  $\sigma_{z,n}^2$  can be represented as

$$\sigma_{z,n}^2 = \frac{Var\{Z_n\}}{(z - \mathbb{E}\{Z_n\})^2}. \quad (4.18)$$

Let  $m_z$  be the peak of  $\sigma_{z,n}^2$  complementary for a given  $z$ ,

$$m_z = \frac{1}{\max_{n \geq 1} \sigma_{z,n}^2} = \min_{n \geq 1} \frac{(z + kn)^2}{Var\{Z_n\}}. \quad (4.19)$$

where  $k = c - \bar{\lambda}$ .

where  $(z)^+$  corresponds to  $\max\{z, 0\}$  i.e. maximum of  $z$  and 0. The expression  $P\{Q > z\}$  is defined by the packet within length. We describe the  $nx$  the time point described by  $n$  at which the generalized variance is maximum. *Maximum Variance Asymptotic (MVA)* approximation,  $e^{-\frac{mx}{2}}$  estimate. It provides an asymptotic upper-limit, the studies in shows different articles that it is an estimation for small values of  $x$  [100],[101],[102],[103]. The boundless buffer space is over the  $z$  point separated by the complete timing.

Now  $P_L(0)$  is evaluated as

$$P_L(0) = \frac{1}{\bar{\lambda}\sqrt{2\pi}} \int_c^\infty (r - c) e^{-\frac{(r-\bar{\lambda})^2}{2\sigma^2}} dr. \quad (4.20)$$

and the  $P_L(z)$  can be used as

$$P_L(z) \approx \frac{P_L(0)}{e^{-\frac{m_0}{2}}} e^{-\frac{m_z}{2}} = \alpha e^{-\frac{m_z}{2}}, \quad (4.21)$$

where

$$\alpha = \frac{1}{\bar{\lambda}\sqrt{2\pi}} \exp\left(\frac{(c - \bar{\lambda})^2}{2\sigma^2}\right) \int_c^\infty (r - c) e^{-\frac{(r-\bar{\lambda})^2}{2\sigma^2}} dr.$$

## 4.5 Results

The approach used to model the traffic is based on time-series data. We take 4 random video files (*Change\_Value*, *Appolo\_Sixteen*, *Media\_Power*, and *Quicktime\_Play*) of distinct length from 31 *seconds* to 70 *seconds*. The varying frame rates are used as raw videos

**Table 4.3** Statistics for different Original Video files

Video original files	I frames	B frames	P frames	fps	Length (secs)
<i>Media_Power</i>	12	1	286	24	36
<i>Changing_Values</i>	13	0	288	24	38
<i>Quicktime_Play</i>	13	1	285	29	69
<i>Appolo_Sixteen</i>	12	0	287	25	31

**Table 4.4** Latency Comparison for different buffer length

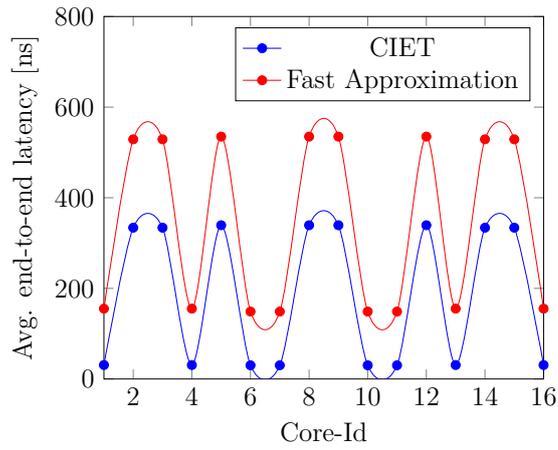
Buffer Length	Traffic generation Methods	Complement -pattern	Tornado -pattern	Uniform -pattern
<b>256 bits</b>	<i>Fast-Approximation Method</i>	342.011	235.069	257.065
	<i>CIET Algorithm</i>	183.576	194.830	183.214
	<b>Enhancement (%)</b>	<b>46.33</b>	<b>17.11</b>	<b>28.73</b>
<b>512 bits</b>	<i>Fast-Approximation Method</i>	559.430	427.330	421.450
	<i>CIET Algorithm</i>	275.680	267.830	258.503
	<b>Enhancement(%)</b>	<b>50.71</b>	<b>37.22</b>	<b>38.65</b>
<b>1024 bits</b>	<i>Fast-Approximation Method</i>	696.374	680.253	700.386
	<i>CIET Algorithm</i>	292.854	268.023	271.400
	<b>Enhancement (%)</b>	<b>57.94</b>	<b>60.60</b>	<b>61.25</b>

ranging from 23 *fps* to 31 *fps*. As shown in Table 4.3, in which the no. of *I frames*, *P frames* and *B frames* are investigated, we find 300 frames. The data sequences are long as they are information of binary bits representing data encoded in the MPEG4 video frame. Average latencies for 3 buffer lengths 1024, 256 and 512 bits are plotted, the end-to-end abeyance for 3 distinct traffic arrangement shown in Fig. 4.10.

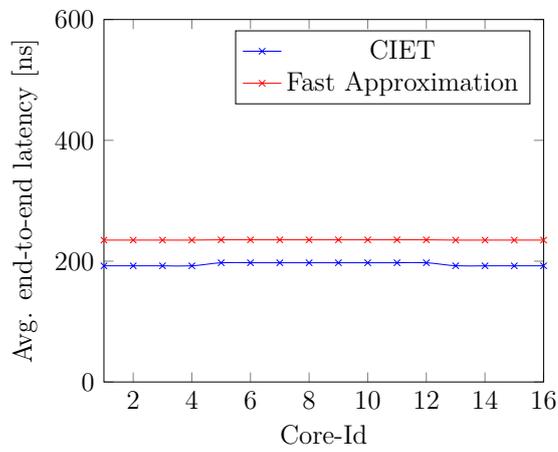
The power consumption for each tile is shown in the Fig. 4.14. The traces produced from the algorithm *CIET* and the technique *Fast Approximation*. In Table 4.6 indicates an increase in average power usage, points the output is 55.54%, 52.71%, and 50.73%, for the *Uniform Tornado* and *Complement* traffic respectively. The Table 4.5 displays the power used by components of router such as under various loads which serves as the basis for calculating core energy. The relation of the power consumption under specific capacity is seen in the Table 4.4. The mean-latencies are diminished by 51.67%, 38.31% and 42.90% for the *Complement-traffic*, *Tornado-traffic* and *Uniform-traffic* individually.

The computation of hurst exponent, we perform 3 evaluation.

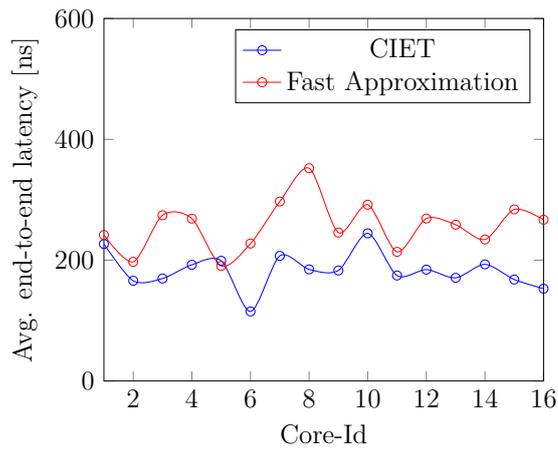
1. The form *variance-time* is the **first-method** used in Section 4.2.1. The plots obtained by plotting the  $\log(\text{var}(X_m))$  against the  $\log(m)$  are from the process of



(a)

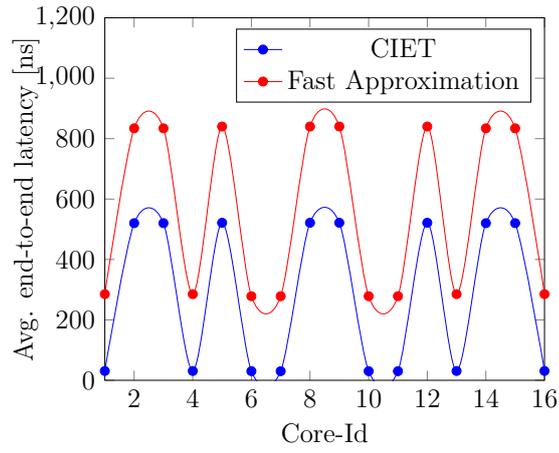


(b)

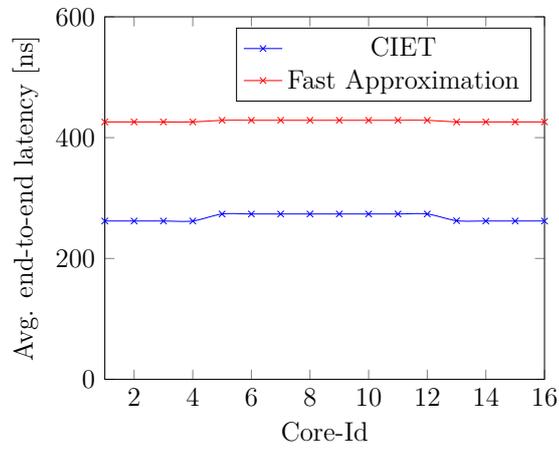


(c)

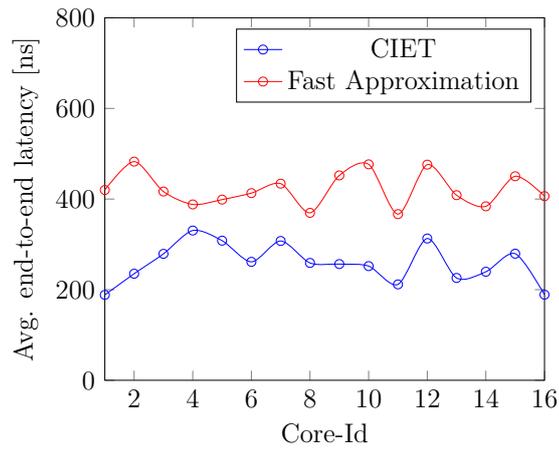
**Fig. 4.7**  $4 \times 4$  Mesh interconnection network's average end-to-end latency for distinct patterns (a) Uniform-traffic (b) Complement-traffic and (c) Tornado-traffic for buffer-size 32 bytes



(a)

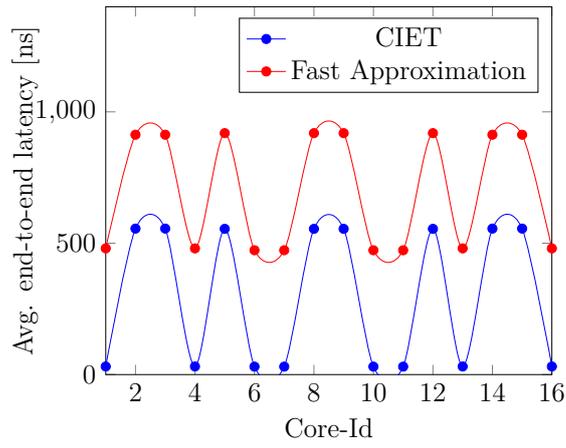


(b)

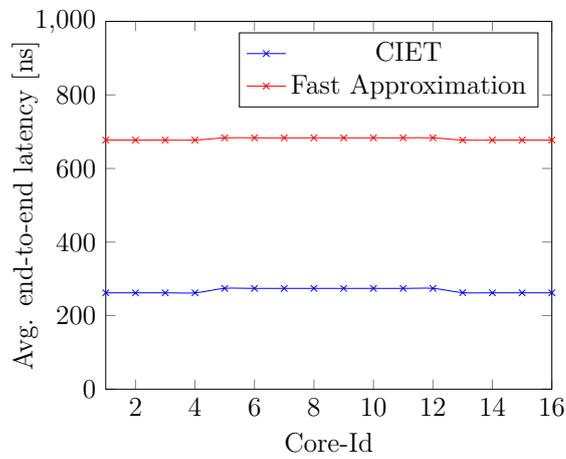


(c)

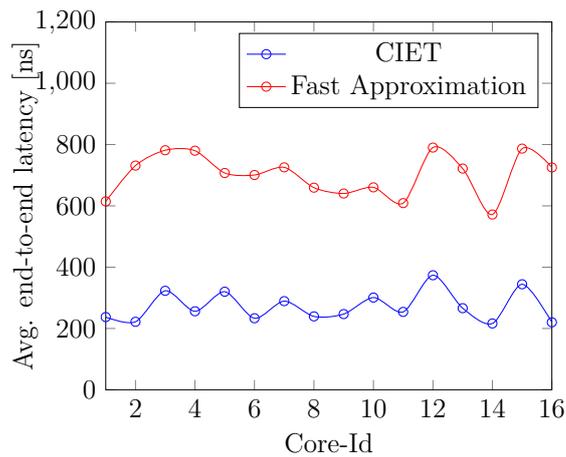
**Fig. 4.8**  $4 \times 4$  Mesh interconnection network's average end-to-end latency for distinct patterns (a) Uniform-traffic (b) Complement-traffic and (c) Tornado-traffic for buffer-size 64 bytes



(a)



(b)



(c)

**Fig. 4.9**  $4 \times 4$  Mesh interconnection network's average end-to-end latency for distinct patterns (a) Uniform-traffic (b) Complement-traffic and (c) Tornado-traffic for buffer-size 128 bytes

**Table 4.5** Router’s Components power under different fractional-load

Fractional-Load \ Router Components	Buffer -Power(mW)	Crossbar -Power(mW)	VC Allocator -Power(mW)	SW Allocator -Power(mW)	Clock -Power(mW)
0.20	4.1876	6.8415	1.3275	0.5333	4.923
0.40	7.2773	1.3557	1.4313	0.9441	4.923
0.60	10.368	20.27	1.5351	1.3550	4.923
0.80	13.455	26.985	1.6390	1.7659	4.923
1.00	16.545	33.701	1.7428	2.1768	4.923

variance-time. In Fig. 4.1 shows plots of 4 different images. The small values of  $m$  are disregarded. The line-climb is represented as the parameter  $\beta$ . With the support of  $\beta$  the Hurst exponent can be calculated by the equation  $H = (1 - \beta) \times 2^{-1}$ .

2. The 2<sup>nd</sup> **method** is specified as *RS Analysis* in the time series  $X$  segment centered on the rescaled modified spectrum. In Fig. 4.2 depicts plots of four different images.
3. Section 4.2.3 explores the 3<sup>rd</sup> **method** established on the periodogram by approximating the density of the time-series  $X$  spectrum.

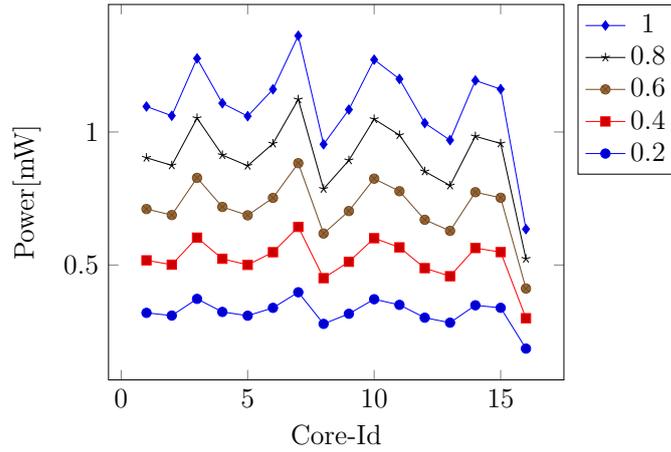
**Table 4.6** Power Estimation under different traffic pattern

Methods Implemented	Complement pattern	Tornado pattern	Uniform pattern
<i>Fast_Approx Method</i>	1.4780	1.4780	1.6061
<i>CIET-Method</i>	0.4159	0.6436	0.5862
<b><i>Enhancement (%)</i></b>	<b>50.74</b>	<b>52.52</b>	<b>55.57</b>

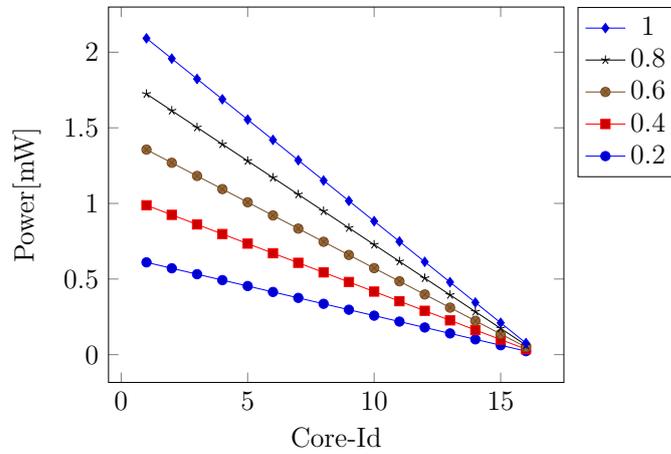
In Table 5.1 contains a table with the measured Hurst parameter. The value of  $H$  indicates the presence of *long-range dependency* in the range between 0.5 and 1. In all three approaches, the value in  $H$  is very close to each other, supports the existence of LRD.

It is noticed that the *Whittle’s Estimation* is the easiest of the 3 methods which are applied. Given this,  $H$  and the most stable of all three is never overestimated. Whereas, due to its complex theory, *RS-method* is very difficult for the application evaluation but provides the benefit of categorizing traffic-flow. The method *variance time* is not so consistent between all methods.

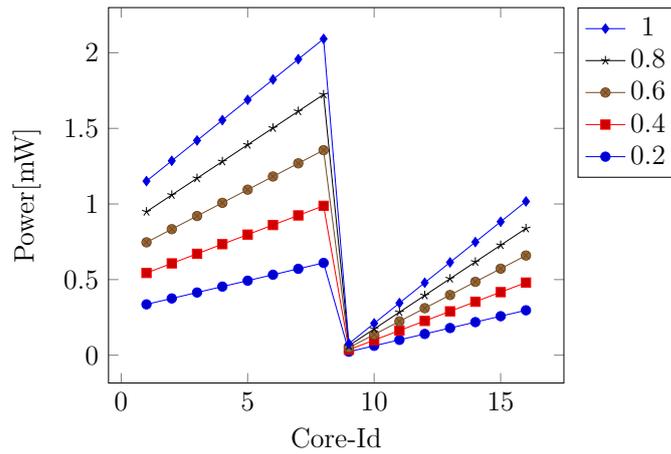
The *RS-method* is valid for deep observation of the traffic-flow particularly on end nodes. In order to monitor traffic function of nodes especially the end nodes by sending control information back to the final node. The process *Whittle’s Estimation* is a valid definition for installing or placing nodes in the network architecture keeping the neighbor into



(a)

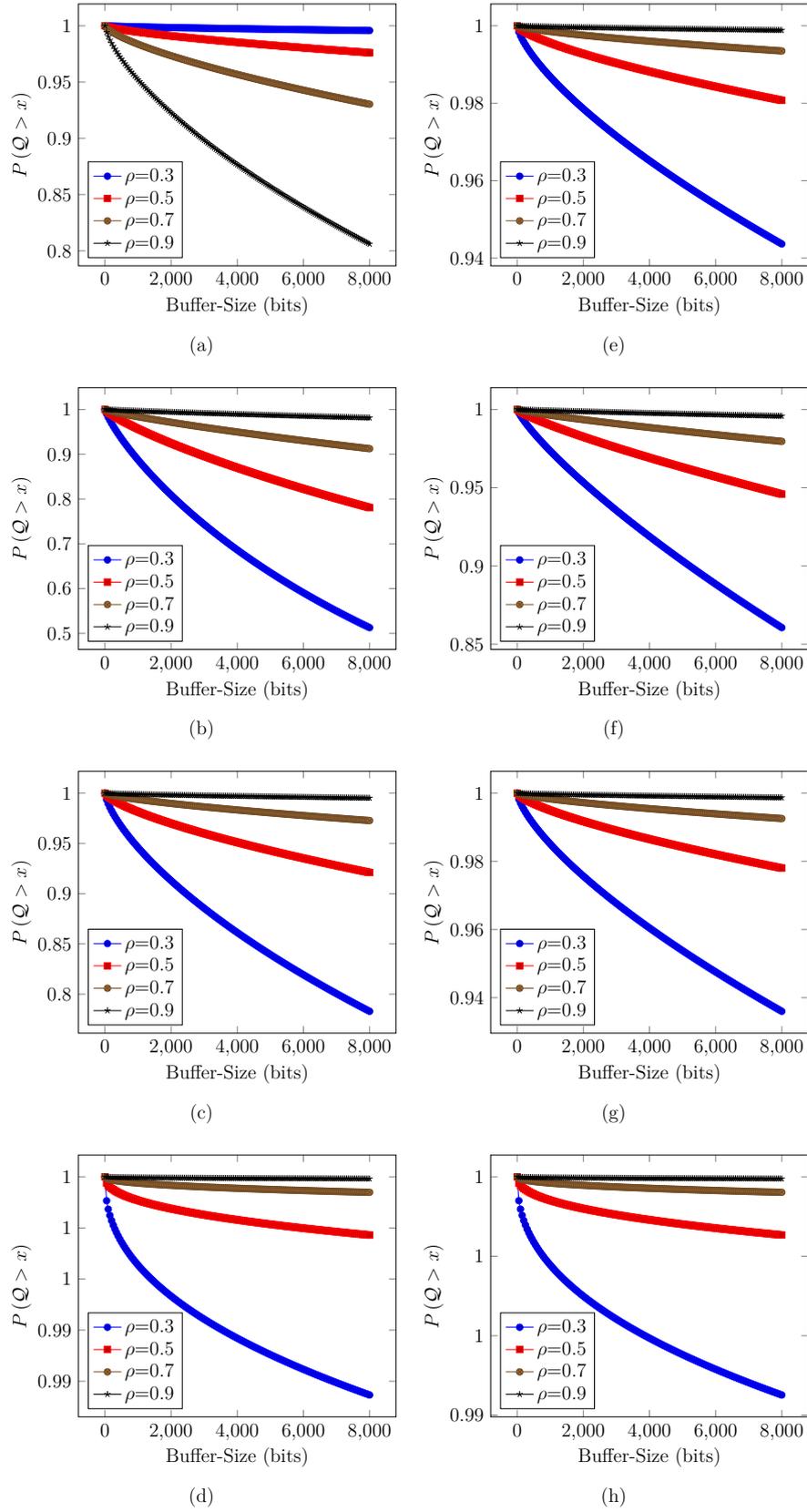


(b)

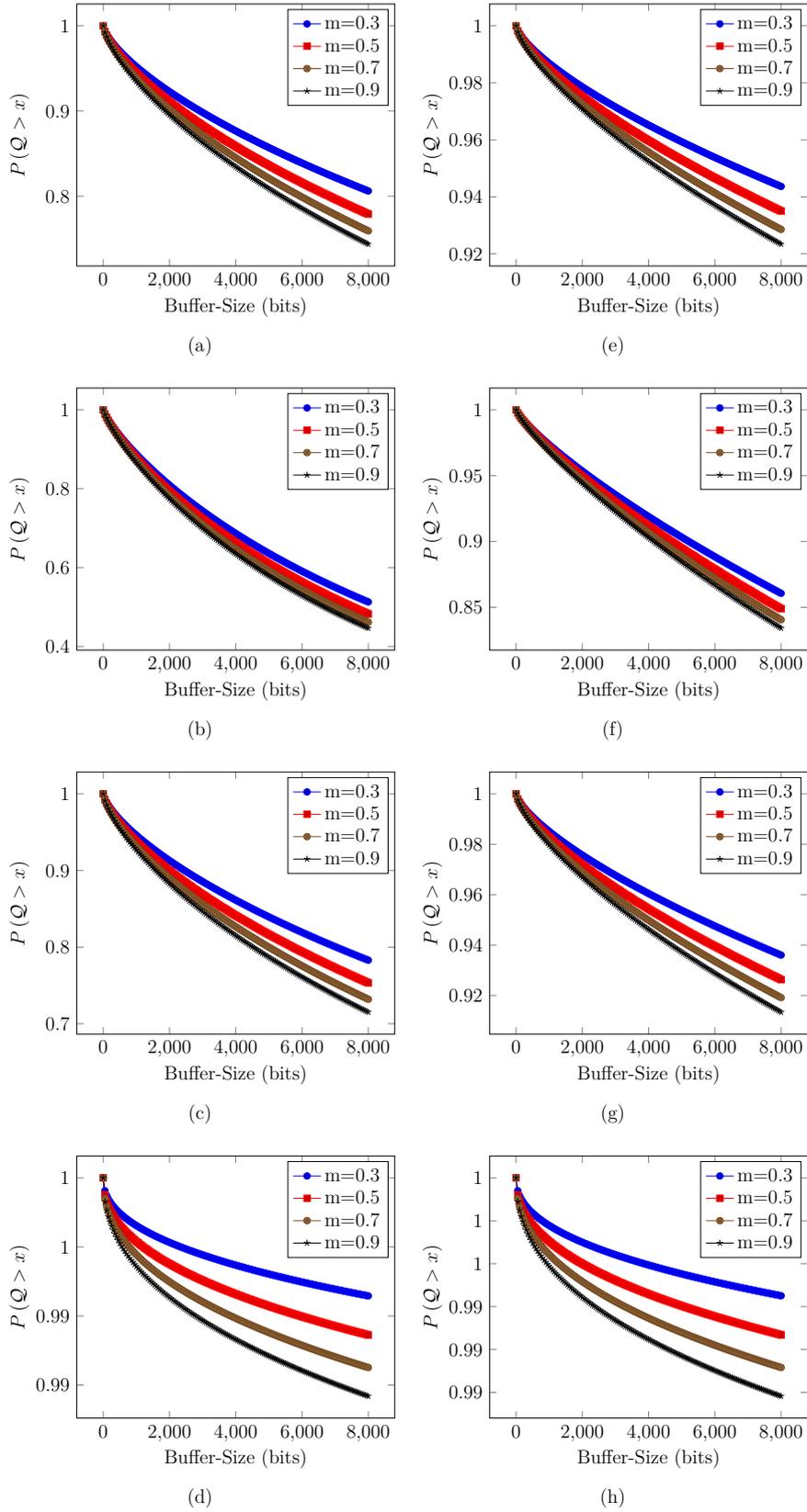


(c)

**Fig. 4.10**  $4 \times 4$  Mesh interconnection network under distinct load average power for distinct patterns (a) Uniform-traffic (b) Complement-traffic (c) Tornado-traffic



**Fig. 4.11** Analytical packet loss-probability with average input-rate ( $m$ ) fixed with varying utilization ( $\rho$ ) Original-files (a) 1 (b) 2 (c) 3 (d) 4 and Synthetic Traces (g) 1 (h) 2 (g) 3 (h) 4 Using Norros Technique



**Fig. 4.12** Analytical packet loss-probability with mean input-rate ( $m$ ) varying with fixed utilization ( $\rho$ ) Original-files (a) 1 (b) 2 (c) 3 (d) 4 and Synthetic TracesS (g) 1 (h) 2 (g) 3 (h) 4 Using Norros Technique

consideration. *Whittle's Estimation* is the strongest approach for specific by specifying traffic flow behavior, and by taking measures to adjust the  $H$  meaning, e.g. by modifying the node setbacks [49].

Data with thousands of steps in the time series are not normal in nature. Various models have distinctive functions analogous to the traffic model *SRD* which is used to measure the likelihood of buffer storage failure as the synthetic traffic model is developed with *LRD*.

In [104], Norros used the FBM model that exhibits the behavior of *LRD*, it evaluates the likelihood of *lower-bound* for  $\mathcal{Q}$  exceeding the trust of the  $x$  buffer level, assuming boundless buffering.

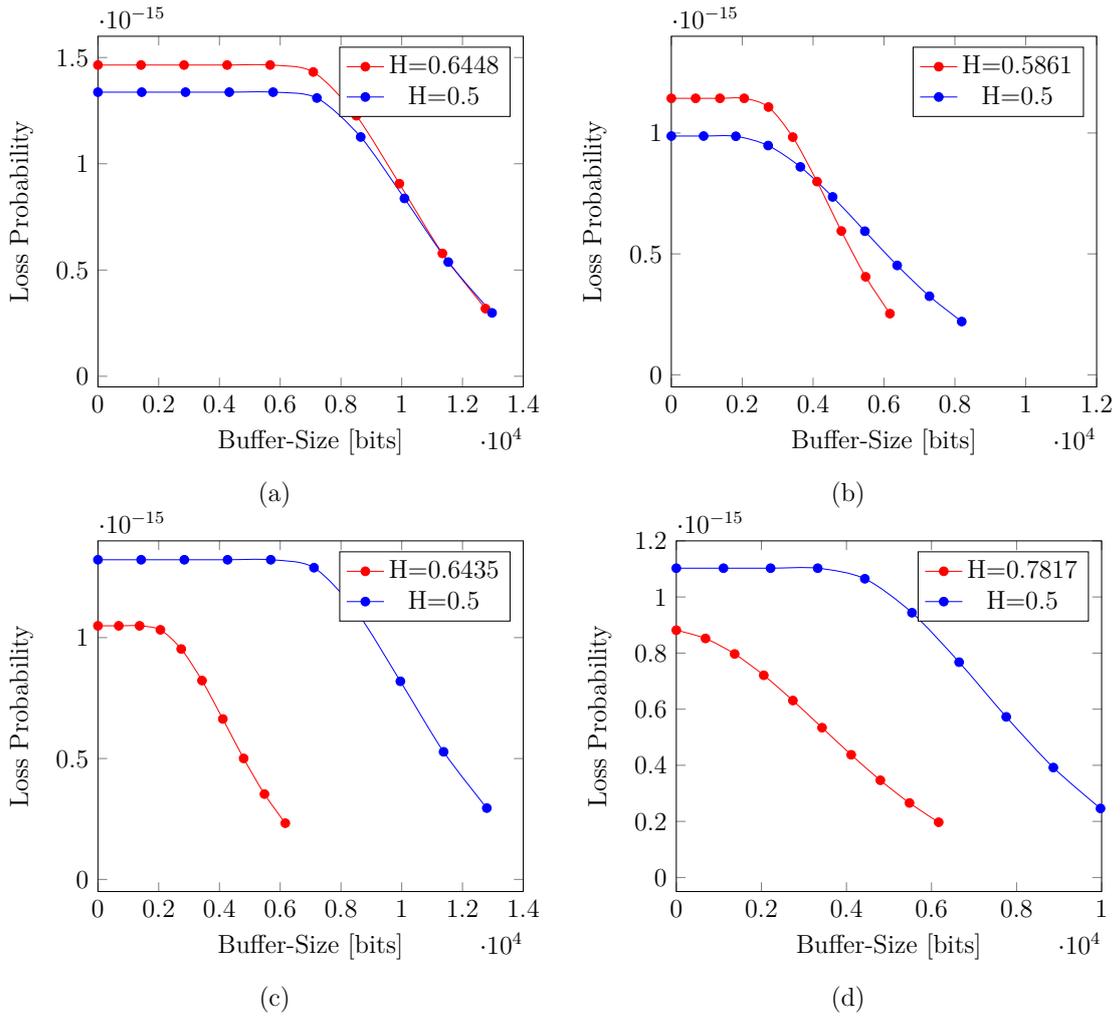
$$P(\mathcal{Q} > z) \sim \exp[-cz^{2-2H}] \quad (4.22)$$

$$c = \frac{m^{2H-1}}{2a} \times \left[ \left( \frac{H}{1-H} \right)^{1-H} + \left( \frac{1-H}{H} \right)^H \right]^2 \times \left( \frac{1-\rho}{\rho} \right) \quad (4.23)$$

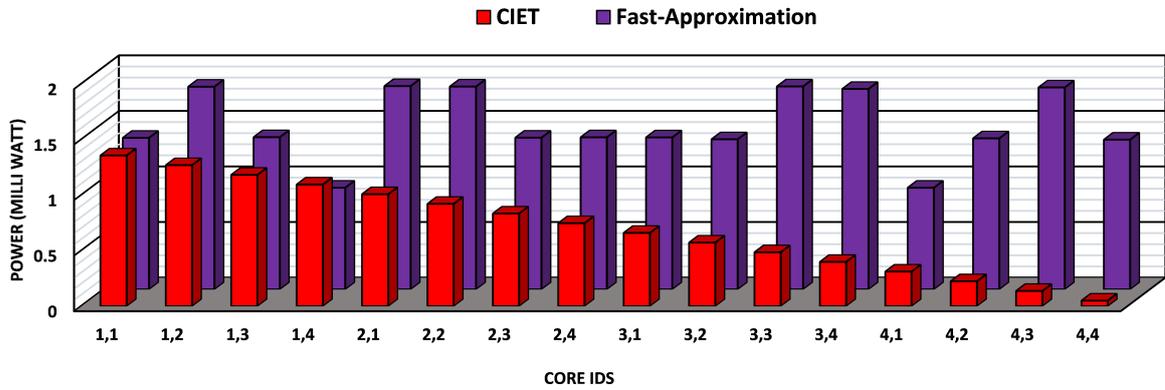
## 4.6 Conclusion

The on-chip network output indicators depending are its precision and impact. The likelihood of buffer packet-loss is depicted in Figs. 4.11 and 4.12, applying the infinite buffer using Norro's strategy. The queue utilization parameter  $\rho$ , the mean input factor  $m$ , the Hurst exponent  $H$  and  $a$  is the combination of variance and mean that derived from Figs. 4.1 and 4.2. The probability of packet-loss is measured against the bounded buffer in these graphs. The analysis is rational on assuming infinite buffer size when figures are plotted Figs. 4.11 and 4.12. In Fig. 4.13 are also theoretical approximations based on the approximation of MVA, when the finite size buffer is used. As the buffer storage is increased, the loss-probability in all traces produced is dramatically decreased. The probability of packet-loss for the synthetic traffics generated are plotted with Hurst exponent equal to  $\frac{1}{2}$ .

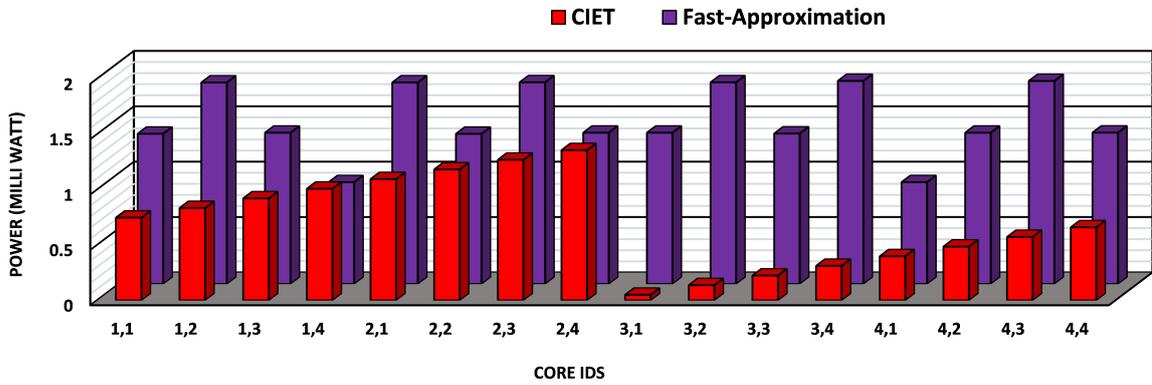
The curves in Fig. 4.11 where the mean input-rate were fixed and utilization factor is varied, the original traffic shows a rather strong statement as a buffer-size target. There are little differences based on parameter assessment, because the traffic-generation from original files uses the  $H$  that can be marginally exaggerate or undervalue. Yet the description of buffer packet-loss likelihood are similar in one and the other sections. In curve Fig. 4.12 curves are very complementary in which we set the utilization-factor ( $\rho$ ) is kept fixed and modified the input-rate ( $m$ ) the curves are produced, the specification calculation adjustments are of same trends as discussed above.



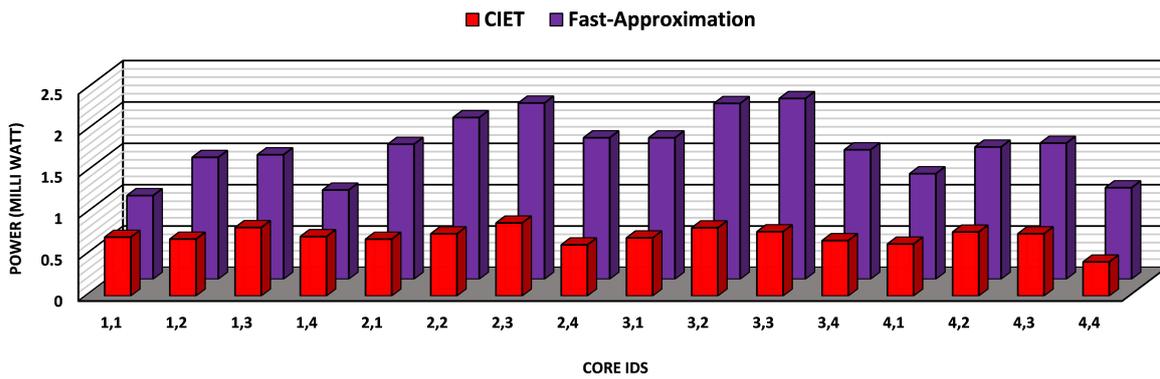
**Fig. 4.13** Packet loss-probability for Synthetic-Trace (a) 1 (b) 2 (c) 3 (d) 4



(a)



(b)



(c)

Fig. 4.14 Power [mW] for each cores by traces generated from *CIET*-algorithm and *Fast-Approximation* technique (a) Complement-traffic (b) Tornado-traffic (c) Uniform-traffic

---

## Synthetic Traffic Performance for Non-Gaussian and Gaussian based traffic on NoC

---

We took two parameters to differentiate between Gaussian and non-Gaussian flows, one being *Skewness* and other *Kurtosis* defined in Eq. (5.1) compute the order or irregularity of the bell-shaped distribution, if the result is positive, otherwise the bulk of the distribution on the right part of the mean is longer and on the left it is much flatter, then the distribution on the left of the mean is lengthy and flatter than on the left. *Kurtosis* is another *Skewness*-like parameter which measures the distribution peak expressed in Eq. (5.2),

$$s = \frac{\mu^3}{\sigma^3} = \frac{\mathbb{E}[(X - \mu)^3]}{(\mathbb{E}[(X - \mu)^2])^{\frac{3}{2}}} \quad (5.1)$$

$$k = \frac{\mu^4}{\sigma^4} = \frac{\mathbb{E}[(X - \mu)^4]}{(\mathbb{E}[(X - \mu)^2])^2} \quad (5.2)$$

the parameters are  $X$ ,  $\mu$  and  $\sigma$  are the process-distribution, *mean* and *standard-deviation* respectively.

The values associated with parameters defines the existence of Gaussian distribution if the magnitude of *Kurtosis* is 3, and *Skewness* is 0. The gaussianity of traffic dispersion if it follows the statement above otherwise it is non-gaussianity process showed in Table 5.2.

4 traces of both traffic for non-Gaussianity and Gaussianity are used, the *Skewness* for Gaussian traffic are nearly 0 and where the *Kurtosis* is nearly 3 then synthetic traffic is not perfectly Gaussian rather close of being called Gaussian. Whereas, traffic with non-Gaussianity the *Skewness* not near to 0 and similar situation for *Kurtosis*.

In *Mandelbrot et al.* [105] examined with respect to the self-similarity which is the idea under which, for instance, all data are held under terms of time or space scaling, for example the quintessence of the information continues as before whether the time and scale changes. The comparative sub-parts amplified then they duplicate the entire item's structure from which it starts. To clarify the idea of self-likeness in which iteration study was used to provide self-comparable fractals, a model is taken from Chapter 1 [106]. The zoomed piece of every one of the fractal sub-parts from the figure uncovers the likeness of the total adjusted picture coin, or clarifies the self-closeness word. Utilizing the self-comparable sub-object traffic hypothesis is displayed with an ON-OFF time arrangement hypothesis. The 1-D set lying on  $\mathcal{C} = [0,1]$  (state) is initiated, at that point the size of the underlying is scaled  $1/3^{rd}$ , at that point the sub-parts of rescaling line  $\mathcal{C}$  are put. The technique is monotonously named up to the infinity, at that point the subsequent set is 1D self-comparable. By anticipating the yield of recursively created batches on the flat time position, the grouping of the ON-OFF status are given which lays the foundation of information traffic.

Self-similarity based stochastic process, intruding the nonpassivity or non-determinism test may either be non-Gaussianity or Gaussianity as proposed by the evidence of traffic determined. The investigation of non-Gaussianity circulation is earliest broke down in [55] where the yield is determined utilizing the non-Gaussian self-comparative system traffic. The instability is caught by the auto-correlation capacity and second-request numbers, plotted against the time slack, diminishes *polynomially* rather than exponential, this nearness of auto-correlation is alluded to as LRD, as the auto-relationship work diminishes step by step as time slacks increase. When contrasted with the standard, short-extend based (or Markovian) model, the LRD forms have a much more slow auto-correlation highlight. LRD's presence recommends that the long-go contrasts are low, and their joined impact isn't zero.

## 5.1 Hermite Process of Non-Gaussian

There is an uncommon procedure having a place with the self-comparative procedure class *Hermite* which displays the property of LRD and has a place with the restrictions of *Non-Central limit theorems* (NCLT). This technique authored as *Rosenblatt* [63], while a similar investigation was done in [107] with the exception of the sub-partial movement. Gaussian *FBM* is the least complex *hermite* process, and the most straightforward non-Gaussian *hermite* class process is the *Rosenblatt* procedure. The requesting technique  $k \geq 1$  can be

composed as  $t \geq 1$  for each time as defined in Eq. (5.3).

$$\mathcal{Z}_H^k(t) = c(H, k) \times \int_{R^k} \left[ \int_0^t \left( \prod_{i=1}^k (q - y_i)_+^{-\left(\frac{1}{2} + \frac{1-H}{k}\right)} \right) dq \right] dW(y_1) \dots dW(y_k) \quad (5.3)$$

where the term  $c(H, k)$  is the relative constant. *Hurst exponent* is represented by  $H$  and is the closeness work of on-off chance that its worth lies in the  $(0, \frac{1}{2})$  set, at that point the procedure is known to be the *Short Range Dependence* process and on-off chance that its worth lies in the interval of  $(\frac{1}{2}, 1)$ , at that point it is LRD process. The importance  $(z)_+$  is set to  $\max(z, 0)$ . The *Wiener-Itô* integral essential for a two-sided Brownian movement  $(W(y)) y \in \mathbb{R}$ [108]. The estimation of  $k$  concludes the nature of the technique created if  $k = 1$  is Gaussian or fundamentally speaking to FBM, and for  $k = 2$  process is of non-Gaussianity in nature analogous to *Rosenblatt* in terms of speaking. All *Hermite*-class has crucial properties:

- The LRD, where the auto-correlation function degenerate very gradually with curve similar to polynomial, which means the summed value is  $\infty$ .
- they have the similar function of covariance

$$\mathbb{E} [\mathcal{Z}_H^q(a) \mathcal{Z}_H^q(b)] = \frac{1}{2} |[a^{2H} + b^{2H} - |a - b|^{2H}]|$$

- for this point on, the method is independent of all that has happened previously (by separate increments), and it always has the same distribution as the initial phase i.e.  $(\mathcal{Z}_H^k(v + g) - \mathcal{Z}_H^k(g))$  for  $v \geq 0$  does not change for  $g > 0$ .
- they have preserved property of self-similarity that is  $(\mathcal{Z}_H^k(cv))$  and  $(c^H \mathcal{Z}_H^k(v))$  for  $v \geq 0$  have similar distribution for value of  $c > 0$

## 5.2 Hurst Parameter or Exponent for non-Gaussian process

The Hurst exponent is determined using Whittle's Estimation [93], [91],[92], which takes the *semi-parametric* method to reduce feature. In Table 5.1 Hurst parameter is determined using Whittle's Estimation for 4 processes based on the property of non-Gaussian and Gaussian. This assumes approximation of the  $\nu(\lambda)$  spectral density of the phase

$$\nu_{c,H}(\lambda) = c \lambda^{1-2H} \quad (5.4)$$

for starting frequencies  $\lambda$ . The time-series  $(X_k)$  where  $k \in [1, n]$  represented as  $T_N(\lambda) =$

**Table 5.1** Hurst Exponent for Non-Gaussian & Gaussian traces

Video Fragement	Fragement-1	Fragement-2	Fragement-3	Fragement-4
<b>Gaussian system</b>	0.6449	0.5862	0.6436	0.7818
<b>Non-gaussian system</b>	0.9462	0.8691	0.9802	0.9701

**Table 5.2** Gaussian's & Non-gaussian's traces Skewness & Kurtosis

Methods		Fragment 1	Fragment 2	Fragment 3	Fragment 4
<b>Gaussian system</b>	<i>Skewness</i> (s)	-0.00078	0.0032	-0.0064	-0.0014
	<i>Kurtosis</i> (k)	2.9970	2.9960	2.9825	3.0203
<b>Non-gaussian system</b>	<i>Skewness</i> (s)	2.5834	3.0935	2.1723	1.032
	<i>Kurtosis</i> (k)	10.091	17.9433	8.0821	4.4452

$\frac{1}{2\pi n} \left| \sum_{t=1}^N X_t e^{i\lambda t} \right|^2$  where  $\lambda$  are the non-periodic frequencies such as  $\lambda_j, N = \frac{2\pi j}{N}$ ,  $0 \leq j \leq \lceil N/2 \rceil$ . The *Whittle Estimation* of the *Hurst Exponent*  $\mathcal{H}_{WE}(m)$  is calculated by minimize the equation

$$\sum_{j=1}^m \log \nu_{c,H}(\lambda_j, N) + \frac{T_N(\lambda_j, N)}{\nu_{c,H}(\lambda_j, N)} \quad (5.5)$$

as per the  $c$  and  $H$ , with  $\nu_{c,H}$ .

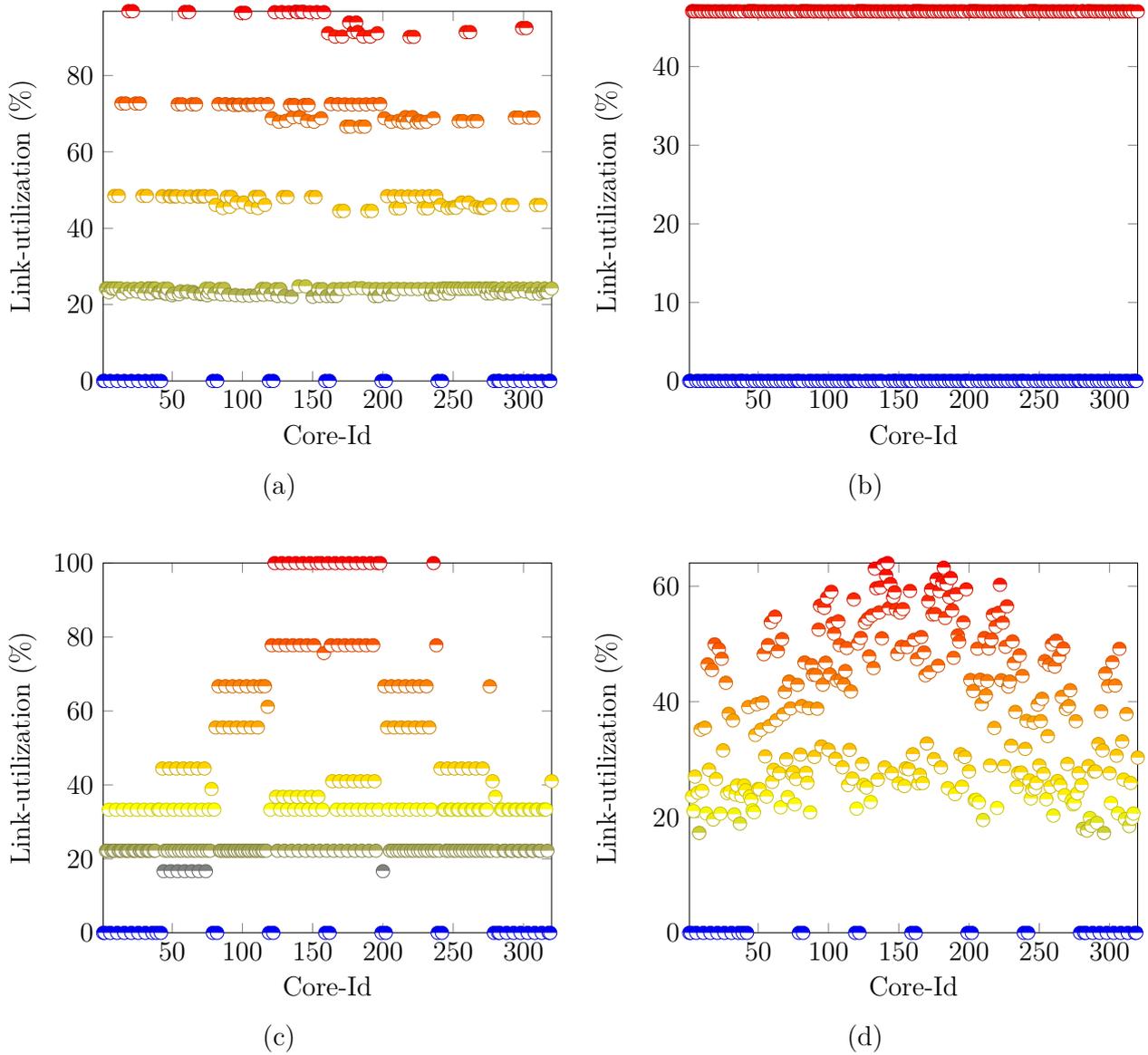
There are various methods to evaluate exponent  $H$  for instance *R/S Analysis* [109], *Periodogram regression* [110], *Variance-Aggregation* [111], and *Wavelet-Analysis* [112] and many more found in [113].

### 5.3 Experimental Results

The OMNET++ framework is utilized to develop an mesh interconnection network of  $8 \times 8$  with the 1.6 Gbps transmission capacity extend, and different parameters are appeared in Table 5.5. The buffer-size that is put on the switch is 1024 bytes with *maximum-queued packets* being 32, of every packet-size being 8 flits with each packet devouring 4 bytes of memory. The Gaussian made from the roundabout inserting strategy [94] which creates

**Table 5.3** Non-Gaussian & Gaussian traffic's link-utilization for traffic patterns

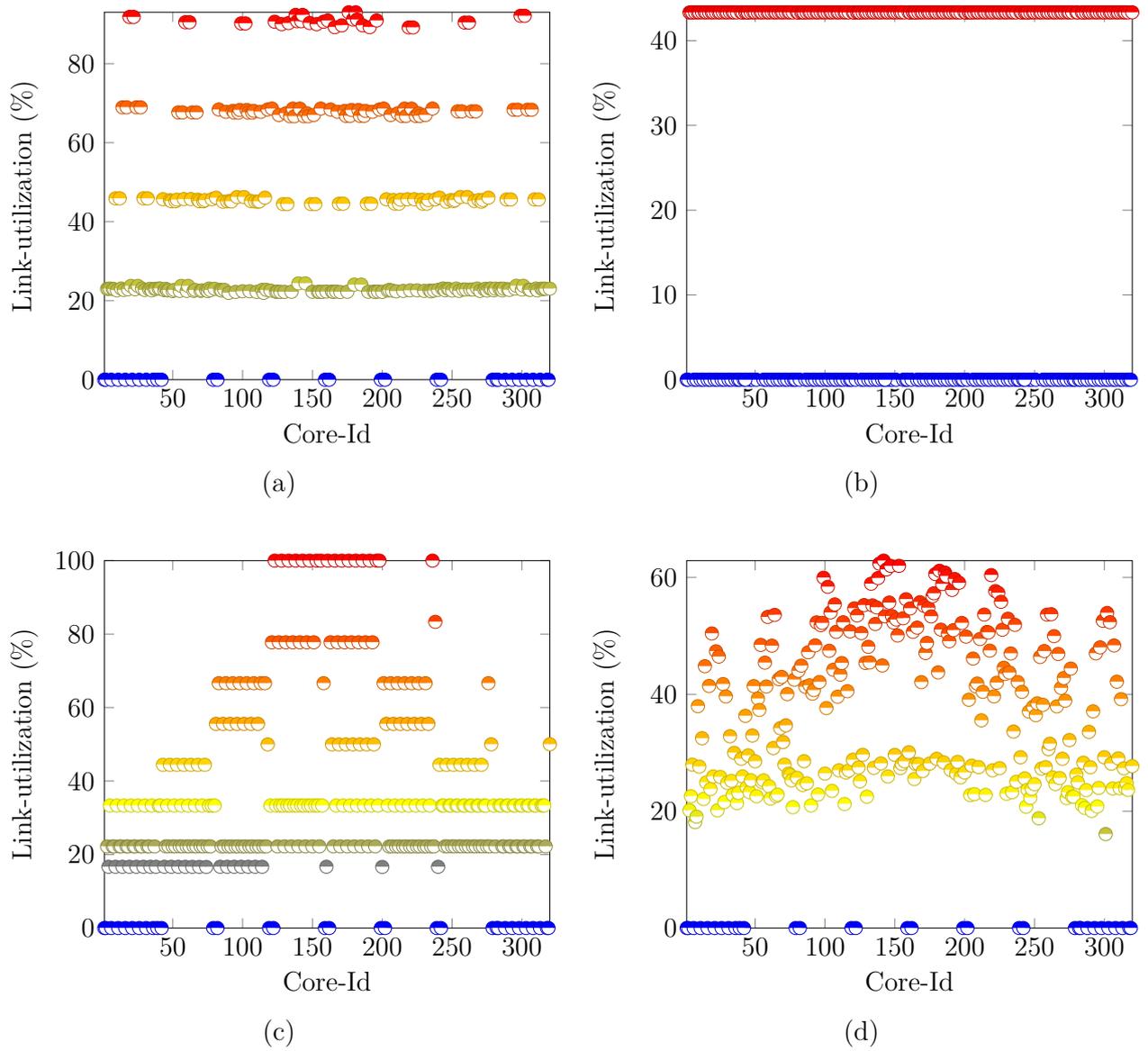
Traffic	Uniform	Tornado	Neighbourhood	Bit-complement
<b>Non-gaussian system</b>	37.847	38.77	43.34	45.358
<b>Gaussian system</b>	37.980	38.98	47.015	46.965
<b>Comparison (%)</b>	<b>0.350</b>	<b>0.54</b>	<b>7.81</b>	<b>3.42</b>



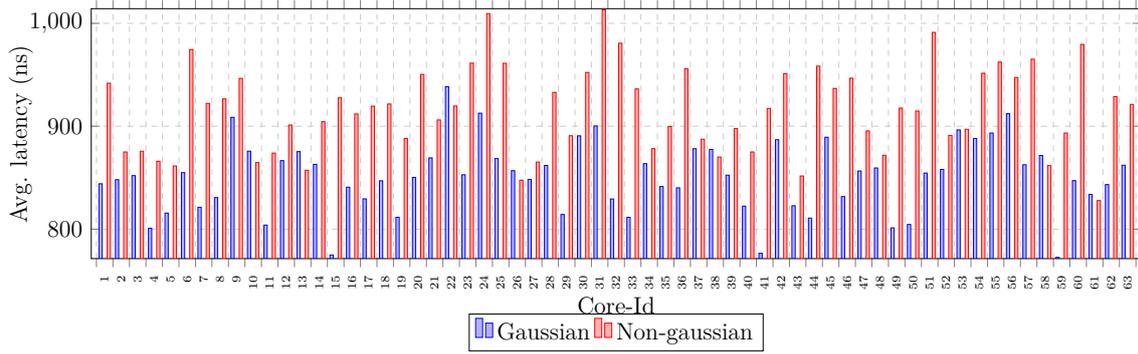
**Fig. 5.1** Gaussian Synthetic Traces average link-utilization for 64 nodes mesh interconnection network for (a) Bit-complement-traffic (b) Neighbourhood-traffic (c) Tornado-traffic (d) Uniform-traffic

**Table 5.4** Non-Gaussian & Gaussian traffic's avg. latency for traffic patterns

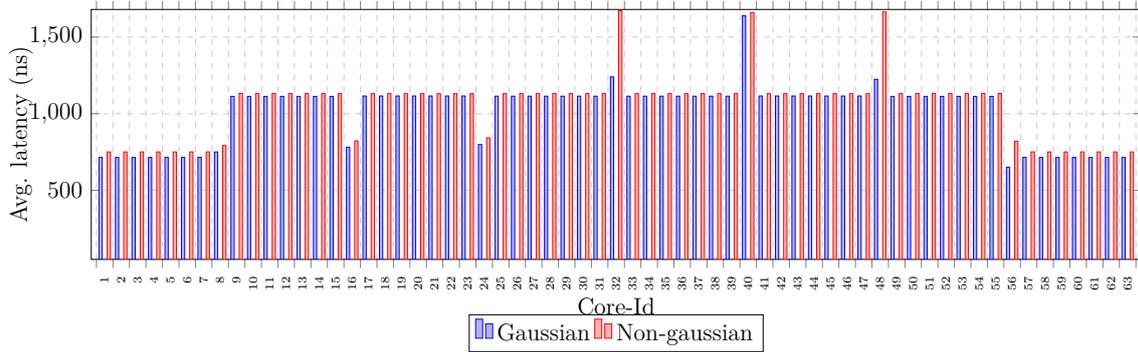
Traffic	Uniform	Tornado	Neighbourhood	Bit-complement
<b>Non-gaussian system</b>	916.30	1042.60	14.94	1116.20
<b>Gaussian system</b>	850.46	1022.85	14.92	1071.20
<b>Comparison(%)</b>	<b>4.032</b>	<b>0.067</b>	<b>1.904</b>	<b>7.185</b>



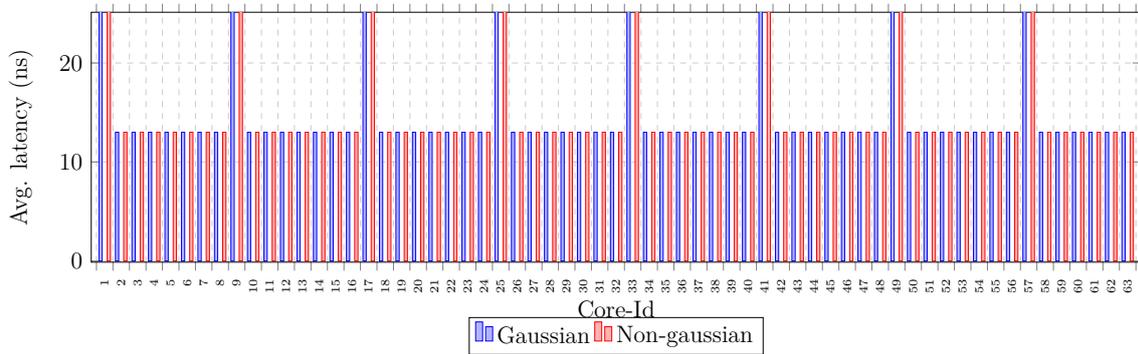
**Fig. 5.2** Non-Gaussian Synthetic Traces average link-utilization for 64 nodes mesh interconnection network for (a) Bit-complement-traffic (b) Neighbourhood-traffic (c) Tornado-traffic (d) Uniform-traffic



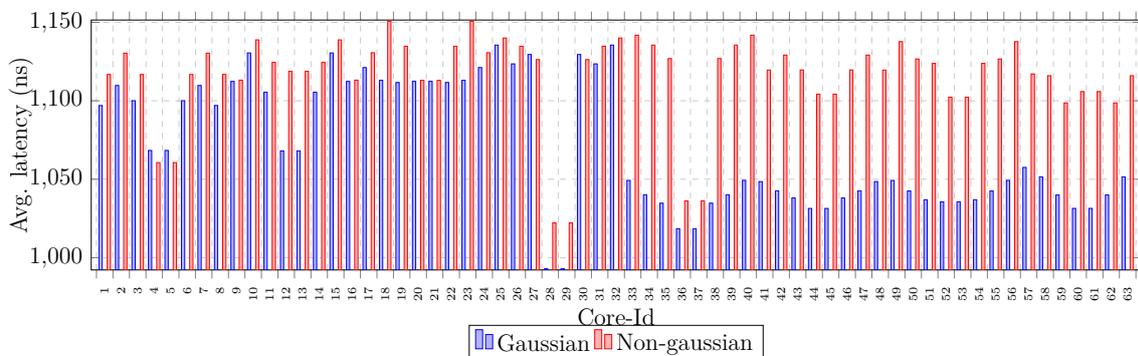
(a)



(b)

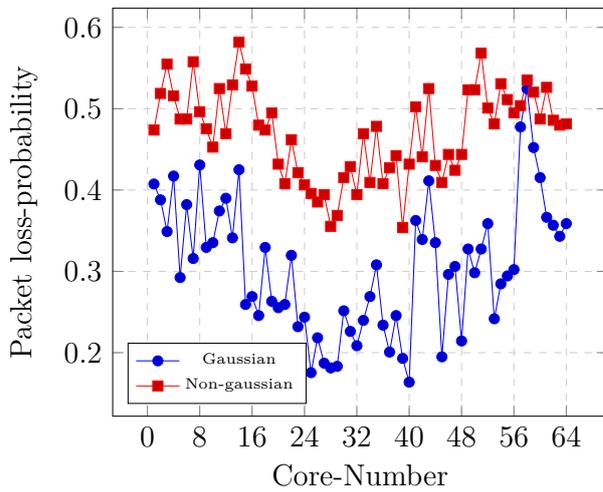


(c)

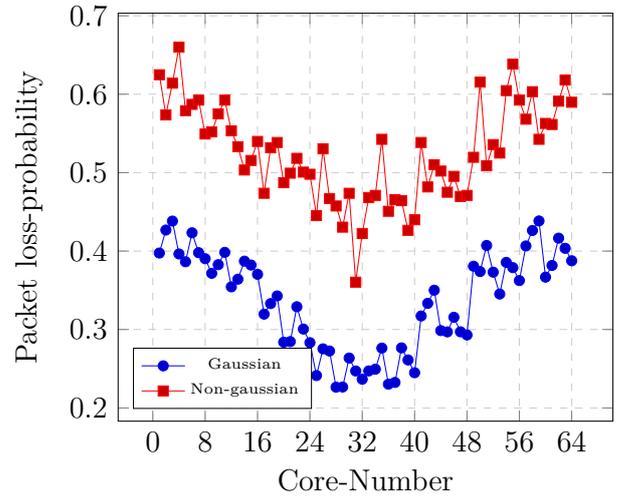


(d)

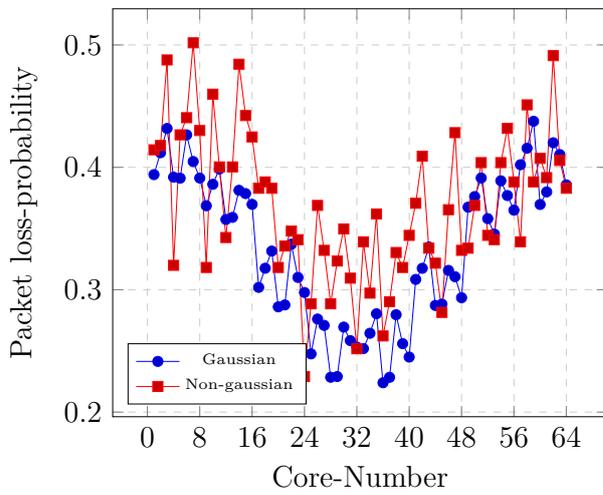
**Fig. 5.3** Non-Gaussian and Gaussian Synthetic Traces avg. latency for  $8 \times 8$  architecture for (a)Uniform (b) Tornado (c) Neighbourhood (d) Bit-Complement traffic



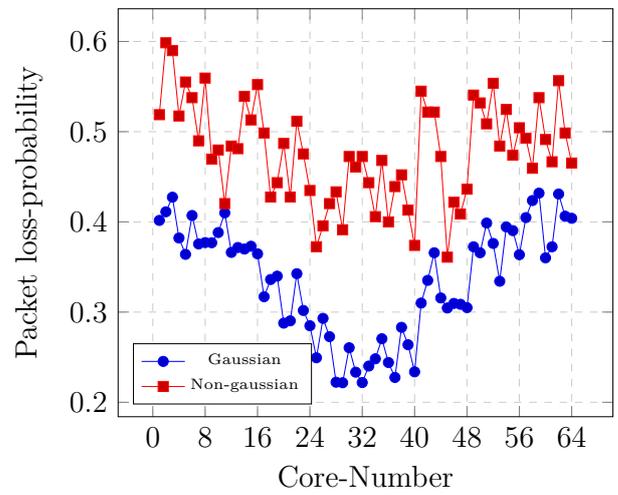
(a)



(b)



(c)



(d)

**Fig. 5.4** Non-Gaussian and Gaussian Synthetic Traces average packet Loss Probability for  $8 \times 8$  mesh architecture (a) Trace 1 (b) Trace 2 (c) Trace 3 (d) Trace 4

**Table 5.5** Parameters used for Simulation on OMNET++

<i>Parameters</i>	<i>Value</i>
No. of Virtual Channel (VC)	2
Flit Size	4 bytes
Start time	1ns
Message Length	4 pkts
Packet Length	8 flits
Flits per VC	1
Arbitration	false
Routing	XY routing
Simulation duration	40ms

the FGN and to follow non-Gaussianity are delivered from the technique characterized by [48] utilizing the brisk estimate. We completed the quest for hints of 4each for non-Gaussian and Gaussian traffic on various traffic designs *Uniform*, *Neighborhood*, *Tornado* and *bit-complement*. In both non-Gaussian and Gaussian traffics, start to finish activity is resolved for each core. The start to finish uncovers the bursty multifaceted nature of non-Gaussian traffic with a higher activity esteem contrasted with the Gaussian for the entire of the traffic patterns recorded in the figures *Bit-complement* Fig. 5.3(a), *Neighborhood*, Fig. 5.3(b), in *Tornado* Fig. 5.3(c), in & *Uniform*, (d) Fig. 5.3.

The qualification is made based on the normal start to finish activity execution for the distinct traffic features in Table 5.4 traffic for non-Gaussianity and Gaussianity uncovers the disparities between *bit-complement* latencies at 4.032%, *Neighborhood* at 0.067%, *Tornado* at 1.904%, and *Uniform* at 7.185%.

The other parameter used to decide traffic effectiveness is normal connection use by each port of the switch. There are 5 ports in NoC design, 4 ports interfacing with the other adjoining core ports in 4 ways for example *east*, *west*, *south* & *north*, and 1 port is associated with the system interface, and there are 320 ports for a  $8 \times 8$  architecture. The normal connection use in Fig. 5.1 and Fig. 5.1 is plotted against each 320 ports for 4 non-Gaussian and Gaussian traffic designs, individually. The blue plots show the router's port was not utilized for packet's transfer correspondence occurrences, while the other shading demonstrates the assortment of different connection utilization.

In Table 5.3 depicts the connection between various traffic drifts among non-Gaussian and Gaussian traffic, the outcome uncovers that later traffic is higher in association use contrasted with non-Gaussianity traffic *bit-complement* is 3.42%, *Neighborhood* is 7.

The last measurement for traffic examination is the packet loss-probability, which is the proportion of the quantity of packets missing to the quantity of packets took care of into the switch. Packet loss-probability is likewise used to arrange the buffer-sizes forced on the switch, which during the early structure stage is a significant issue for the on-chip organize manufacturer. In Fig. 6.4 the packet loss-probability is decided for the 4 traffic hints of non-Gaussianity and Gaussianity, the plots uncover that the packet loss-probability of the packets traffic for non-Gaussian is more noteworthy than that of Gaussian.

## 5.4 Conclusion

We actualized multicore engineering reenactment with non-Gaussian and Gaussian traffic making senders and beneficiaries dissemination reliant on explicit traffic designs. Indeed, even as non-Gaussian system traffic debases productivity comparative with Gaussian traffic, the NoC uncovers comparative examples wherein non-Gaussian traffic turns out to be additionally rushing in nature for gadget traffic. We additionally analyzed the outcomes for various follows for both non-Gaussian and Gaussian traffic uncovers Gaussian traffic effectiveness has a bit of leeway over its counterpart i.e. Non-Gaussian.

In any case, it is indicated that ongoing traffic has the minor non-Gaussian conveyance in nature, so for the exactness of the outcomes organize planner plays out the investigation traffic of the non-Gaussian conduct which helps in the choice of optical correspondence assets before an engineering's initial structure stage. In any case, we think our execution made ready for more work to refine the devices for making better models and suggesting strategies for compelling activity.

---

Non-Gaussian Traffic Modelling for Multicore Architecture Using  
Wavelet Based Rosenblatt Process

---

### 6.1 Rosenblatt Process

A stationary Gaussian sequence process  $(\xi_n)_{n \in \mathbb{Z}}$  with zero mean and unity variance with its function correlation represented as  $\mathcal{C}(n) = \mathbb{E}(\xi_0 \xi_n) = n^{\frac{2H-2}{k}} S(n)$ , with hurst exponent  $H \in (\frac{1}{2}, 1)$  and  $S$  is a slowly varying function which lags very slowly at infinity. Let the Hermite rank of the process  $k$  and  $f$  be the function defined as

$$f(x) = \sum_{i \geq 0} c_i H_i(x), c_i = \frac{1}{i!} \mathbb{E}(f(\xi_0) H_i(\xi_0)), \quad (6.1)$$

where  $H_i(x)$  is the Hermite polynomial of degree  $i$  and  $k = \min\{i | c_i \neq 0\} \geq 1$ . The Hermite polynomial is defined as  $H_i(x) = (-1)^i e^{\frac{x^2}{2}} \frac{d^i}{dx^i} e^{-\frac{x^2}{2}}$ . According to the Non-Central Limit Theorem (NCLT),  $\frac{1}{n^H} \sum_{i=1}^{[nt]} f(\xi_i)$  converges as  $n \rightarrow \infty$  in the meaning of finite-dimension distributions to the process in Eq. (6.2).

Rosenblatt process is an instance of the non-Gaussian class whose increments are fixed and self-similar. The covariance of the Rosenblatt process and the fractional Brownian motion (FBM) are identical in that FBM is also a self-similar process but in essence Gaussian. For generating network traffic or econometrics data, the self-similar stochastic process has always attracted the researcher. The method of Rosenblatt is a kind of Hermite class of processes, which is then normalized the sum of LRD random variables. Hermite Processes

is the simpler FBM are self-similar. The Rosenblatt process  $Z_k(t)$  can be expressed as

$$Z_H^k(t) = c(H, k) \int_{R^k} \int_0^t \left( \prod_{i=1}^k (s - \varphi_i)_+^{-\left(\frac{1}{2} + \frac{1-H}{k}\right)} \right) ds dB(\varphi_1) dB(\varphi_2) \quad (6.2)$$

where  $c_{(H,k)}$  is the normalized non-negative constant to satisfy  $\mathbb{E} Z_k(1)^2 = 1$ ,  $\int_{R^2}$  is the double Wiener-Itó multiple integral of order  $k$ ,  $x_+ = \max(x, 0)$  for  $x \in R$ , whereas  $B(\varphi)$  it is a standard FBM. The process is defined by  $k \in \left(\frac{1}{4}, \frac{1}{2}\right)$  with fixed increment, to specification it is also known by *fractional Rosenblatt motion* (fRm). The term  $k$  is considered as the Hurst parameter, used to denote the processes in the Eq. (6.2). It is calculated by various method which is defined in [48], [114] and [115]. The  $Z_k$  is denoted in terms of self-similarity index or Hurst parameter by the following equation

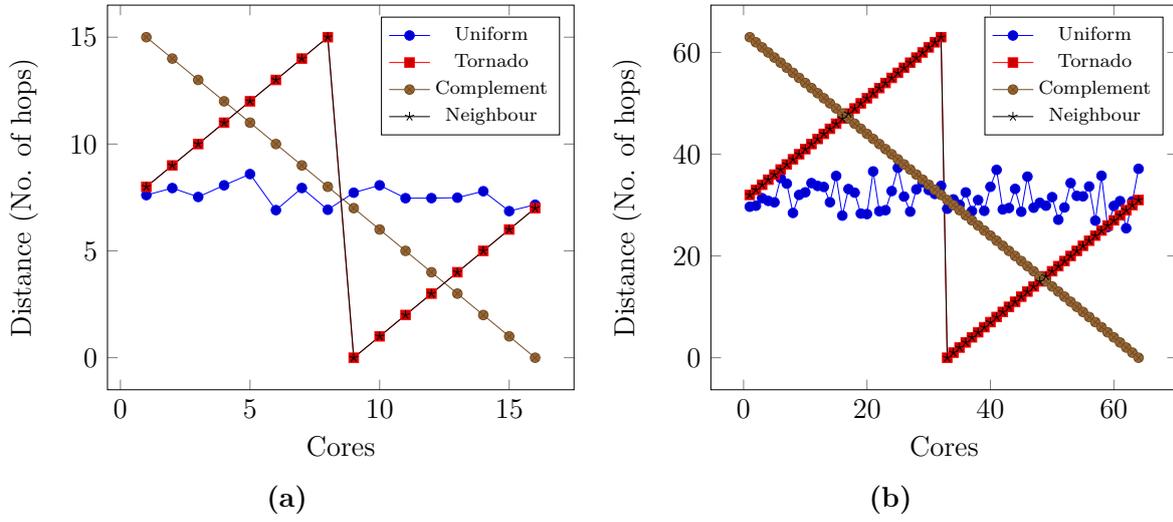
$$\{Z_k(ck)\}_{t \in R} \stackrel{m}{=} \{c^{2k} Z_k(t)\}_{t \in R} \quad (6.3)$$

where  $c \geq 0$  and the  $m$  equality is based on finite-dimensional distribution, the example can be found in Chapter 7 in [116]. The *fRm* process distribution is non-Gaussian and the tails are much heavier than the Gaussian distribution. The moments of fRm are finite makes it different from fractional Brownian motion (fBm) although both are having the stationary increments but *fBm* are Gaussian in nature whereas *fRm* is non-Gaussian. Rosenblatt process is introduced by Rosenblatt in [117] and further studies can be found in [61], [118], [56]. The Rosenblatt process in Eq. (6.2) is double Wiener-Itó integral where it can be generalized if it is single integral then it is a simple Hermite process known as *fractional Brownian motion* (fBm) and if it is a double integral then it is a second-order Hermite process also known as by *fractional Rosenblatt motion* (fRm). Let  $B_H, H \in Z$  be a zero mean time-series with stationary having long-range dependency, there covariance in denoted by  $\mathbb{E}(B_H B_0) = Y(k) H^{\alpha-1}$  where  $Y$  is gradually changing function as  $H \rightarrow \infty$  with  $\alpha \in (0, 1)$ .

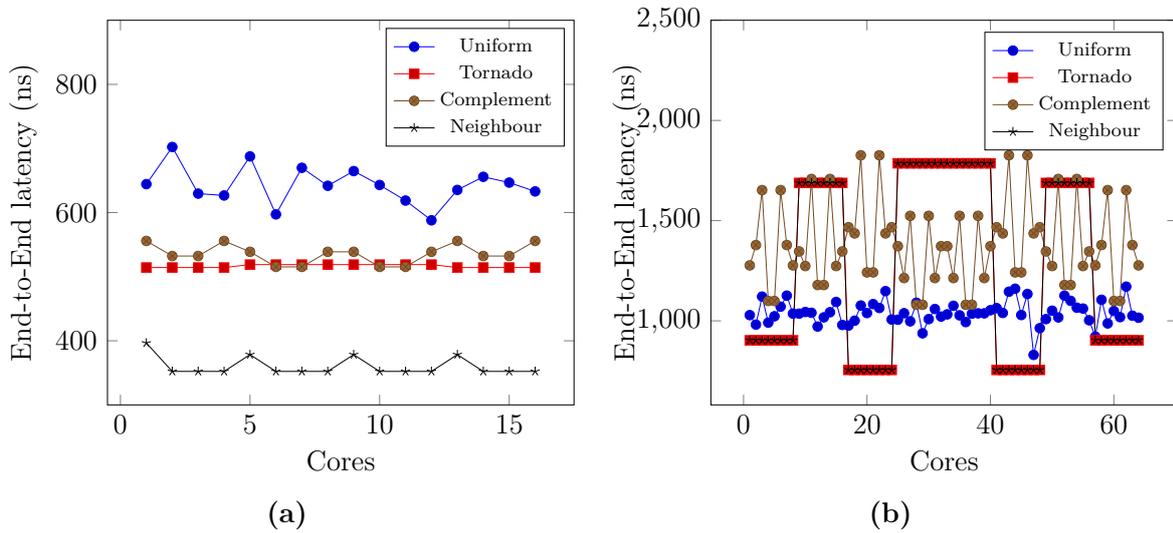
## 6.2 Algorithm for generation of Fractional Rosenblatt Process

For generating the process we use the wavelet-based FARIMA process, which uses the Hurst parameter for the initial phase of the process, to construct the Rosenblatt process. The generation is based on the approximation of *fRm*. In order to generate the fractional filter, the initial filters are used as  $\mu^0$  and  $\omega^0$  where they are available from [119] page no. 196. The fractional filters can be calculated by the following equation

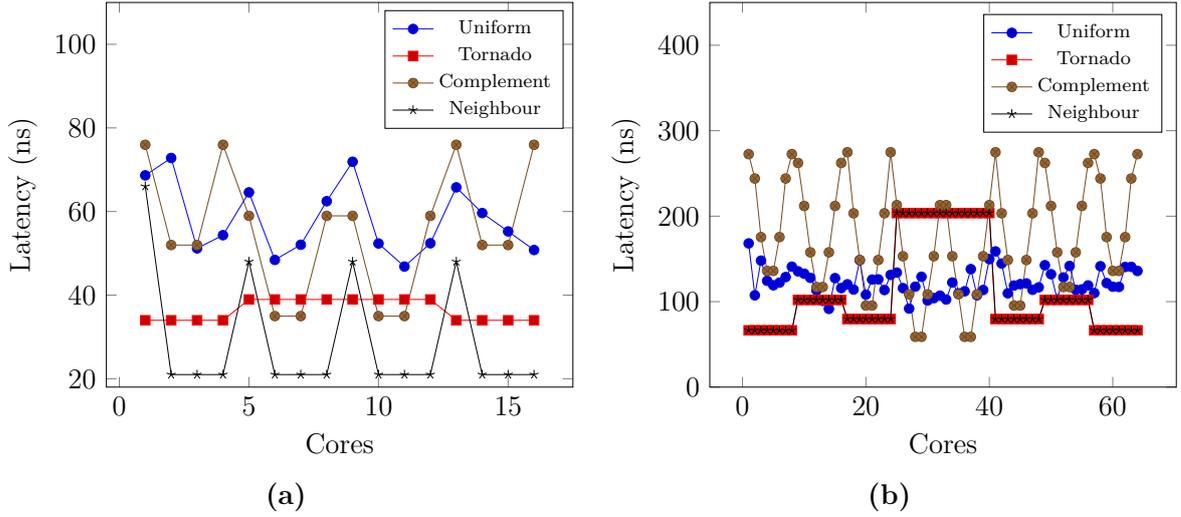
$$\mu^g(z) = f^{(N+g)} \mu^0(z) \quad (6.4)$$



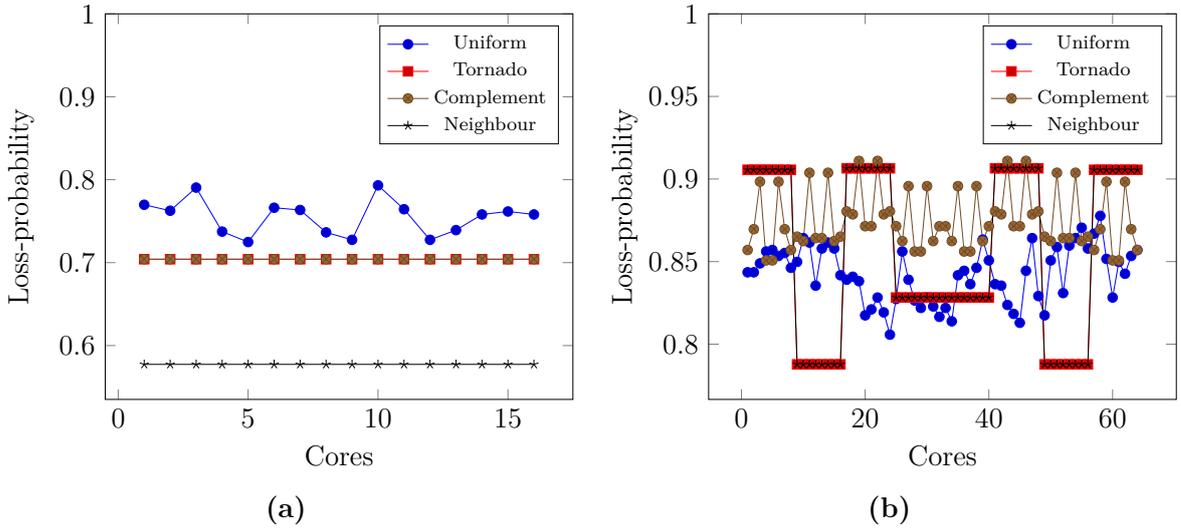
**Fig. 6.1** Average Distance(number of hops) for (a) 16 (b) 64 nodes Mesh topology for Complement, Tornado, Neighbor & Uniform traffic



**Fig. 6.2** Average estimation of End-to-End latency(ns) for (a) 16 (b) 64 nodes Mesh topology for Complement, Tornado, Neighbor & Uniform traffic



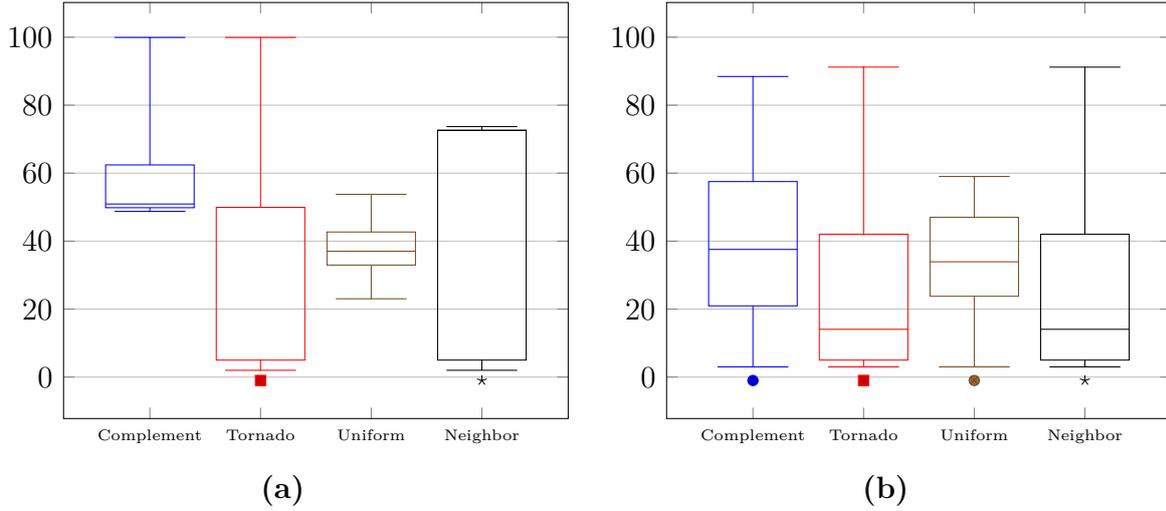
**Fig. 6.3** Average Network latency(ns) for (a) 16 (b) 64 nodes Mesh topology for Complement, Tornado, Neighbor & Uniform traffic



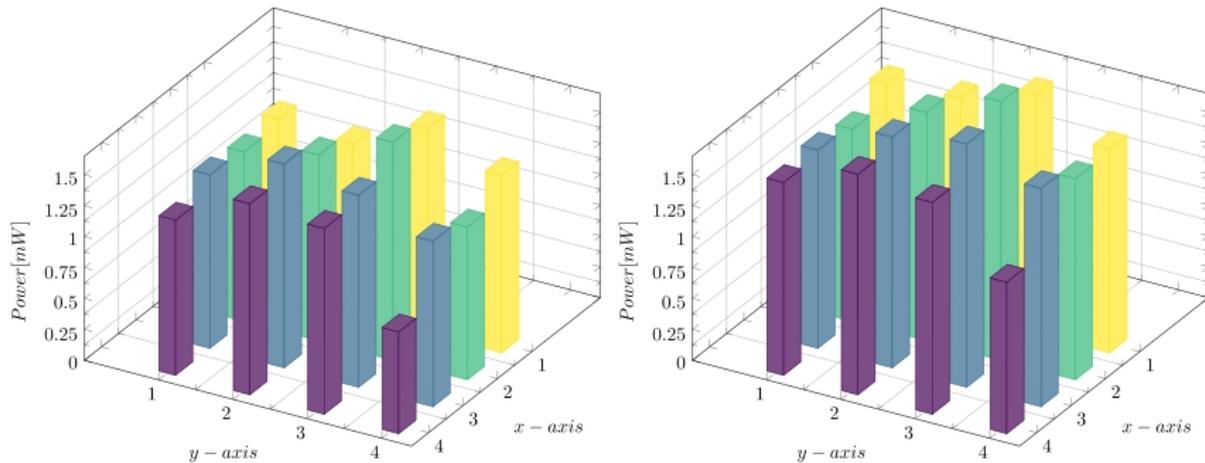
**Fig. 6.4** Average Loss probability for (a) 16 (b) 64 nodes Mesh topology for Complement, Tornado, Neighbor & Uniform traffic

as the value of  $N$  increases  $f^{(N+g)}$  and  $h^{(g-N)}$  filters decay a lot faster than usual  $\mu^g(z)$  and  $\omega^g(z)$  for further the relation between truncated off series and  $N$  zero moments can be found in [120].

In Step 1 the parameters used for the generation for initial *FARIMA* processes are collected i.e. the Hurst parameter, zero moments of multiresolution orthogonal wavelet basis, length of the process and a scaling parameter. In the next step, another parameter used  $L$  which lie between  $-M$  to  $J$  (which a choose parameter in the previous step). The initial filters are chosen for both low and high pass filters which are infinite so need to be truncated. In the next step recursively applying the Circular Embedding Technique (CIET) on initial



**Fig. 6.5** Boxplot for Average Link Utilization(%) for (a) 16 (b) 64 nodes Mesh topology for Complement, Tornado, Neighbor & Uniform traffic



**Fig. 6.6** Power estimation for each tile of  $4 \times 4$  Mesh network (a) Fractional Rosenblatt process (b) Simple Hermite process

*fractional Brownian motion* (fBm) [121]. The fast Fourier transform is applied on this process generated in the previous sub-step calling it  $\aleph^{(g)}$ . In the last sub-step of Step 2 the partial sum of  $\aleph^{(g)}$  is generated finally generating the initial *FARIMA* process calling it *fractional Rosenbaltt process* (fRm). In the Step 3 approximation of *fRm* is applied using Eq. (6.5).

### 6.3 Experimental Results

For the results, we have used OMNeT++ simulator, where two interconnections architecture is designed for a mesh network of size  $4 \times 4$  and  $8 \times 8$ . The simulation parameters are the same taken in the Table 5.5. The traffic generated from the above algorithm is used to simulate the traffic on these two mesh network. For source-destination pair the synthetic

---

**Algorithm 6.1: Algorithm for Fractional Rosenblatt Motion Process generation using Circular Embedding Technique (CIET)**

---

**Input:** Hurst Parameter, Fractional Brownian Motion

**Output:** Fractional Rosenblatt Motion

Step<sub>1</sub> : Select a parameter  $H \in (\frac{1}{4}, \frac{1}{2})$  defining fRm.

Select the multiresolution orthogonal wavelet basis, these orthogonal wavelets have zero moments  $N$

Time length for the fRm  $T = 2M$  Choosing the scale  $2^{-J}$

Step<sub>2</sub> : function initial\_famira(M, J, fbm)

Choosing the value of  $L$  such that  $-M \geq L \geq J$

Choosing the high and low pass fractional filter  $\mu^g$  and  $\omega^g$ , these filters need to be truncated at  $r$  since it is infinite.

for  $k = 0, 1, \dots, r + 2^{(M+L)}$

$\aleph^g = CIET(fBm)$

end

for  $k = 0, 1, \dots, J - L$

$\aleph^{(g)} = FFT(\aleph^g)$

end

for  $k = 0, 1, \dots, 2^{(M+L)}$

$\tilde{W}_{g,0}^{(k,2)} = \sum_{0 \leq j \leq k} \left( (\aleph^{(g)})^2 - E(\aleph^{(g)})^2 \right)$

end

Step<sub>3</sub> : Generate the sequence using approximation of fRm using following equation

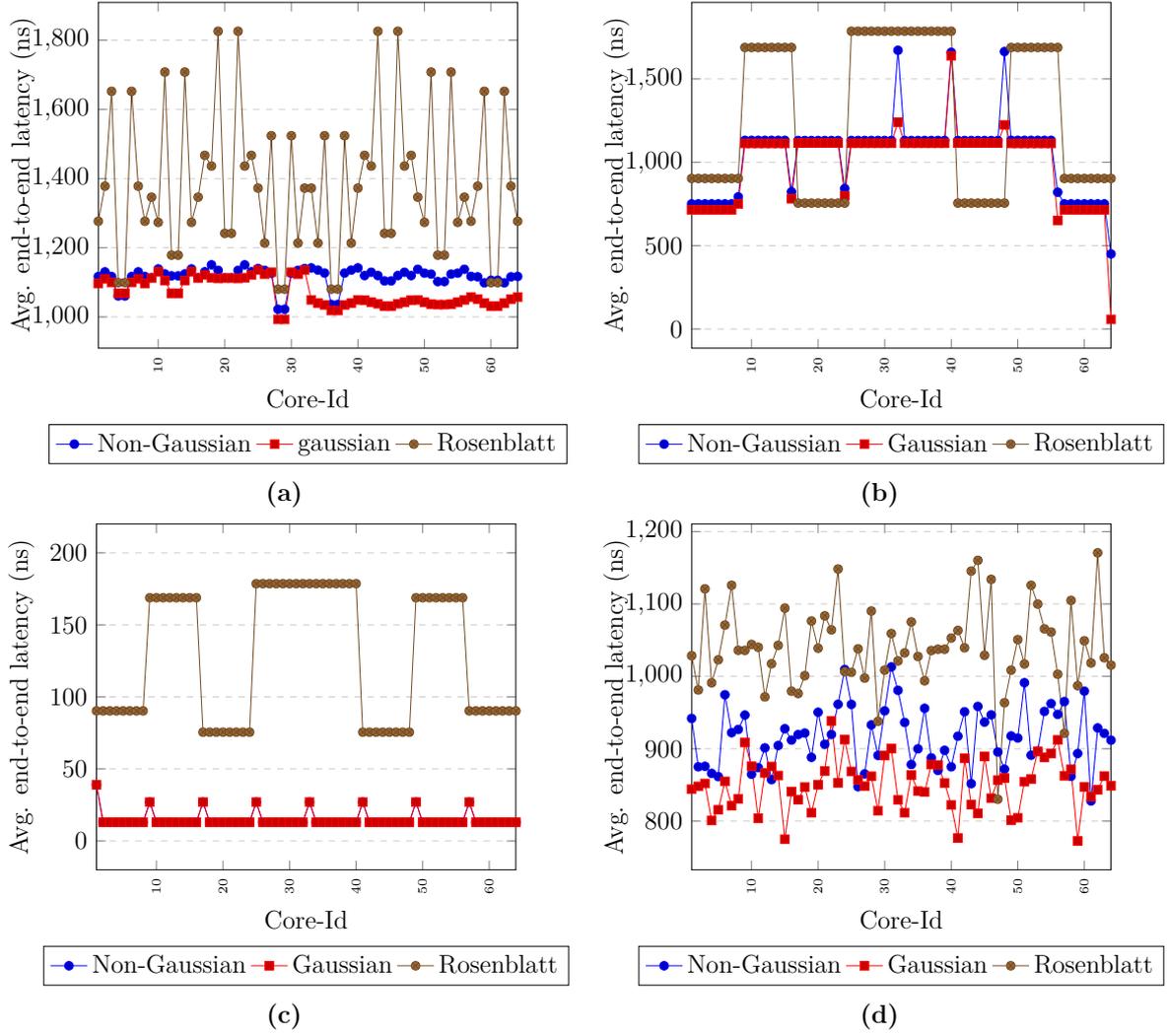
$$Z_{k,2} \left( L, J, t = g^{2^{-J}} \right) = C_k 2^{-2kJ} \tilde{W}_{g,0}^{(k,2)} \quad (6.5)$$

where  $C_k = \frac{\Gamma(k)\Gamma(1-k)\sqrt{4(k-1)k}}{\Gamma(1-2k)}$

return  $Z_{k,2}$  in vector

---

traffic we used are four traffic patterns i.e. Complement, Neighbor, Tornado and Uniform. In Fig. 6.1 the Average distance (number of hops) is represented against each core, the graph shows the overlapping of curve in complement and neighbor traffic since its average distance will be the same. The network and end-to-end latency is plotted in Fig. 6.2 and Fig. 6.3 respectively, where the range of latencies vary in both, in the network latency it is based on average latency between hop to hop whereas end-to-end latency is based on source to destination. The packet loss-probability is shown in Fig. 6.4 for  $4 \times 4$  and  $8 \times 8$  respectively, the average packet-loss is  $8 \times 8$  architecture is higher as compared to  $4 \times 4$  networks.



**Fig. 6.7** Average latency estimation of End\_to\_End (ns) for 64 nodes Mesh topology for non-Gaussian, Gaussian and Rosenblatt process for (a) Complement-traffic (b) Tornado-traffic (c) Neighbor-traffic (d) Uniform-traffic

In Fig. 6.7 all four traffic patterns have an average end-to-end latency for non-Gaussian, Gaussian and Rosenblatt process, as per the plot it can be interpreted that Rosenblatt process is burstier than non-Gaussian and Gaussian processes. The algorithm for generating synthetic traffic for non-Gaussian and Gaussian processes can be found in [122].

The parameter used for the simulation as mentioned above we have used  $4 \times 4$  and  $8 \times 8$  mesh topology, where message length size is of 4 packets and packet length is of size 8 flits where the flit size is of 4 bytes. The size of the buffer is maintained by maximum queued packets which is 16.

In Fig. 6.2 plotted mean end-to-end latencies for all 4 traffic patterns considering the three processes as Gaussian, non-Gaussian and Rosenblatt process. As in all the four cases,

**Table 6.1** Average Packet-Loss Probability for comparison for 64 mesh architecture nodes

Methods	Complement -Traffic	Tornado -Traffic	Uniform -Traffic	Neighbor -Traffic
<i>Non-Gaussian</i>	0.4704	0.3688	0.5254	0.4781
<i>Gaussian</i>	0.3051	0.3387	0.3357	0.3349
<i>Rosenblatt</i>	0.8748	0.8570	0.8422	0.8570

**Table 6.2** Average end-to-end latency [ns] comparison for 64 mesh architecture nodes

Methods	Complement -Traffic	Tornado -Traffic	Uniform -Traffic	Neighbor -Traffic
<i>Non-Gaussian</i>	1116.02	1042.59	916.29	14.93
<i>Gaussian</i>	1071.21	999.02	850.46	14.93
<i>Rosenblatt</i>	1379.67	1283.12	1039.69	1283.91

the Rosenblatt process shows more latency as comparison to non-Gaussian and Gaussian. In Table Tables 6.1 and 6.2 simulated parameters for complement, tornado, uniform and neighbor traffic are showing mean end-to-end latency and packet-loss probability for Gaussian, non-Gaussian and Rosenblatt process.

In Table 6.2 it is seen that the traffic generated from Rosenblatt process is 19.10% for complement, 18.84% for tornado, 11.86% for uniform and 88.37% for neighbor traffic burstier than non-Gaussian processes. If it is compared with Gaussian process it is found that 22.33% for complement, 22.14% for tornado, 18.20% for uniform and 88.37% for neighbor traffic burstier. In comparison with the packet-loss probability it is found that generated Rosenblatt process is 46.22% for complement, 56.96% for tornado, 37.61% for uniform and 44.21% for neighbor is more in losing packets from the buffer as comparison to non-Gaussian, whereas in case of Gaussian it is found to be more skeptical to packet-loss which is 69.12% for complement, 60.47% for tornado, 60.14% for uniform and 60.92% for neighbor. We considered a standardized flow of traffic in which average power for Rosenblatt process is 0.71 *mW* whereas for the simple Hermite process it is 0.79 *mW* an improvement of 11.29%.

## 6.4 Conclusion

All these results show that the non-Gaussian based Rosenblatt process is more data-driven traffic, which is better analogous to the real-time traffic which comprises of gaming, high-quality videos and other high processing data streams whose inter-arrival time of packet arrival cannot be judged by Gaussian distribution.

So the need for non-Gaussian process becomes the need of traffic modelling in order to capture the true nature of the network resources that are placed inside the multicore architecture. It will be helpful for the network designers to choose optimally network resources inside the router, virtual channels or arbitrator, wrong selection and quantity of these network resources can lead to the problem of buffer overflow, loss of process cycle, huge energy dissipation and network contention.

---

## Conclusions and Future Work's Scope

---

### 7.1 Conclusions

In this thesis, the synthetic traffic generation is studied and categorized. Work aims to address the role of synthetic traffic in the fields of multicore architecture. The primary objective of this thesis is to address the complexity of currently available synthetic traffic generation algorithm and the synthetic traffic generated is of old class of MPEG videos. The communication management is generally managed by the passive monitoring system. The generation of realistic traffics from the statistical properties of the real data flows are made from the self-similarity principle. The traffics generated are realistic as compared to real videos.

As the synthetic traffic consumes fewer simulation hours and also aids the network-designer to choose optimal network-resources which are used for the designing of on-chip architecture. We have generated the Synthetic traffic with an algorithm whose running complexity is low as compared to other available algorithms for new real bursty data. Our result shows that our algorithm is better in achieving resource usage optimization with different parameters such as packet-loss probability network and end\_to\_end latency & link utilization. We have designed an algorithm non-Gaussian based synthetic traffic generation and showed the requirement of the non-Gaussian traffic for the simulation as Gaussian traffic are not the best feature to capture the true features of the multicore architecture.

The role of machine learning is introduced for the synthetic traffic's generation apart from

the statistical properties of realistic data. The use of machine learning technology improves the quality parameters of synthetic traffic and helps in the optimization of communication resources.

## **7.2 Scope for Future Work**

In this thesis we have addressed the issue of better Synthetic traffic generation algorithm for the new class of MPEG data set, we have the problem of Gaussian traffic as compared to the non-Gaussian traffic. In future, other machine learning with incorporation with the statistical method to generate realistic based Synthetic traffic generation. To optimize the network resources traffic is generated with faster running time and fewer simulation hours requirement.

# Appendices

## .1 Validation of Rosenblatt-process of Hermite Class

The Hermite class of  $2^{nd}$ -order process is Rosenblatt-process  $\mathcal{Z}(t), t \geq 0$  has stationary increments with stable path then

$$\mathcal{Z}_H^k(t+h) - \mathcal{Z}_H^k(t) \quad (1)$$

for  $t \in [0, T]$  is independent of  $h$ .

The function of covariance of the Eq. (1) is represented by

$$\mathbb{E}(\mathcal{Z}_H^k(t)\mathcal{Z}_H^k(s)) = \frac{1}{2} \left( t^{2H_E} + s^{2H_E} - |t-s|^{2H_E} \right) \quad (2)$$

for consequently all  $s, t \in [0, T]$

$$\mathbb{E} |\mathcal{Z}_H^k(t) - \mathcal{Z}_H^k(s)|^2 = |t-s|^{2H_E} \quad (3)$$

where  $H_E$  is the Hurst Exponent.

---

## Bibliography

---

- [1] P. Gargigni. Available online at <http://www.itrs.net/>.
- [2] Y. Xue, Z. Qian, P. Bogdan, F. Ye, and C.-Y. Tsui, “Disease diagnosis-on-a-chip: Large scale networks-on-chip based multicore platform for protein folding analysis,” in *Design Automation Conference (DAC), 2014 51st ACM/EDAC/IEEE*, pp. 1–6, IEEE, 2014.
- [3] R. Marculescu, U. Y. Ogras, L.-S. Peh, N. E. Jerger, and Y. Hoskote, “Outstanding research problems in noc design: system, microarchitecture, and circuit perspectives,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 28, no. 1, pp. 3–21, 2009.
- [4] V. Soteriou, H. Wang, and L.-S. Peh, “A statistical traffic model for on-chip interconnection networks,” in *Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2006. MASCOTS 2006. 14th IEEE International Symposium on*, pp. 104–116, IEEE, 2006.
- [5] R. Prolonge and F. Clermidy, “Network-on-chip traffic modeling for data flow applications,” in *Proceedings of the 2013 Workshop on Rapid Simulation and Performance Evaluation: Methods and Tools*, p. 2, ACM, 2013.
- [6] K. Park and W. Willinger, “Self-similar network traffic: An overview,” *Self-similar network traffic and performance evaluation*, pp. 1–38, 2000.
- [7] W. J. Dally and B. P. Towles, *Principles and practices of interconnection networks*. Elsevier, 2004.

- [8] J. Duato, S. Yalamanchili, and L. M. Ni, *Interconnection networks: An engineering approach*. Morgan Kaufmann, 2003.
- [9] R. Marculescu, P. Bogdan, *et al.*, “The chip is the network: Toward a science of network-on-chip design,” *Foundations and Trends® in Electronic Design Automation*, vol. 2, no. 4, pp. 371–461, 2009.
- [10] A. Jantsch, H. Tenhunen, *et al.*, *Networks on chip*, vol. 396. Springer, 2003.
- [11] L. Benini and G. De Micheli, “Networks on chips: A new soc paradigm,” *computer*, vol. 35, no. 1, pp. 70–78, 2002.
- [12] W. J. Dally and B. Towles, “Route packets, not wires: on-chip interconnection networks,” in *Proceedings of the 38th annual Design Automation Conference*, pp. 684–689, 2001.
- [13] M. Sgroi, M. Sheets, A. Mihal, K. Keutzer, S. Malik, J. Rabaey, and A. Sangiovanni-Vincentelli, “Addressing the system-on-a-chip interconnect woes through communication-based design,” in *Proceedings of the 38th Design Automation Conference (IEEE Cat. No. 01CH37232)*, pp. 667–672, IEEE, 2001.
- [14] Z. Qian, P. Bogdan, C.-Y. Tsui, and R. Marculescu, “Performance evaluation of noc-based multicore systems: From traffic analysis to noc latency modeling,” *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 21, no. 3, pp. 1–38, 2016.
- [15] K. Lahiri, A. Raghunathan, and S. Dey, “Evaluation of the traffic-performance characteristics of system-on-chip communication architectures,” in *VLSI Design 2001. Fourteenth International Conference on VLSI Design*, pp. 29–35, IEEE, 2001.
- [16] T. Bjerregaard and S. Mahadevan, “A survey of research and practices of network-on-chip,” *ACM Computing Surveys (CSUR)*, vol. 38, no. 1, pp. 1–es, 2006.
- [17] M. Dai, Y. Zhang, and D. Loguinov, “A unified traffic model for mpeg-4 and h. 264 video traces,” *IEEE Transactions on Multimedia*, vol. 11, no. 5, pp. 1010–1023, 2009.
- [18] W. Liu, J. Xu, X. Wu, Y. Ye, X. Wang, W. Zhang, M. Nikdast, and Z. Wang, “A noc traffic suite based on real applications,” in *2011 IEEE computer society annual symposium on VLSI*, pp. 66–71, IEEE, 2011.
- [19] P. Bogdan, M. Kas, R. Marculescu, and O. Mutlu, “Quale: A quantum-leap inspired model for non-stationary analysis of noc traffic in chip multi-processors,” in *2010 Fourth ACM/IEEE International Symposium on Networks-on-Chip*, pp. 241–248, IEEE, 2010.

- [20] P. Doukhan, G. Oppenheim, and M. Taqqu, *Theory and applications of long-range dependence*. Springer Science & Business Media, 2002.
- [21] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, “On the self-similar nature of ethernet traffic (extended version),” *IEEE/ACM Transactions on networking*, vol. 2, no. 1, pp. 1–15, 1994.
- [22] A. Balachandran, G. M. Voelker, P. Bahl, and P. V. Rangan, “Characterizing user behavior and network performance in a public wireless lan,” in *Proceedings of the 2002 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pp. 195–205, 2002.
- [23] S. W. Turner, *Performance analysis of multiprocessor interconnection networks using a burst-traffic model*. PhD thesis, University of Illinois at Urbana-Champaign, 1995.
- [24] V. Paxson and S. Floyd, “Wide area traffic: the failure of poisson modeling,” *IEEE/ACM Transactions on networking*, vol. 3, no. 3, pp. 226–244, 1995.
- [25] C. Grecu, A. Ivanov, P. Pande, A. Jantsch, E. Salminen, and R. Marculescu, “An initiative towards open network-on-chip benchmarks,” *OCI-IP White Paper*, 2007.
- [26] J. H. Bahn and N. Bagherzadeh, “A generic traffic model for on-chip interconnection networks,” *Network on Chip Architectures*, p. 22, 2008.
- [27] L. Tedesco, A. Mello, D. Garibotti, N. Calazans, and F. Moraes, “Traffic generation and performance evaluation for mesh-based nocs,” in *Proceedings of the 18th annual symposium on Integrated circuits and system design*, pp. 184–189, 2005.
- [28] I. Cohen, O. Rottenstreich, and I. Keslassy, “Statistical approach to networks-on-chip,” *IEEE Transactions on Computers*, vol. 59, no. 6, pp. 748–761, 2010.
- [29] S. Suboh, M. Bakhouya, J. Gaber, and T. El-Ghazawi, “Analytical modeling and evaluation of network-on-chip architectures,” in *2010 International Conference on High Performance Computing & Simulation*, pp. 615–622, IEEE, 2010.
- [30] W. Dai and N. E. Jerger, “Sampling-based approaches to accelerate network-on-chip simulation,” in *2014 Eighth IEEE/ACM International Symposium on Networks-on-Chip (NoCS)*, pp. 41–48, IEEE, 2014.
- [31] Y. Zhang, B. Ozisikyilmaz, G. Memik, J. Kim, and A. Choudhary, “Analyzing the impact of on-chip network traffic on program phases for cmps,” in *2009 IEEE International Symposium on Performance Analysis of Systems and Software*, pp. 218–226, IEEE, 2009.

- [32] M. Badr and N. E. Jerger, “Synfull: Synthetic traffic models capturing cache coherent behaviour,” *ACM SIGARCH Computer Architecture News*, vol. 42, no. 3, pp. 109–120, 2014.
- [33] P. Bogdan and R. Marculescu, “Non-stationary traffic analysis and its implications on multicore platform design,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 4, pp. 508–519, 2011.
- [34] P. Bogdan and R. Marculescu, “Quantum-like effects in network-on-chip buffers behavior,” in *Proceedings of the 44th annual Design Automation Conference*, pp. 266–267, 2007.
- [35] P. Bogdan and R. Marculescu, “Statistical physics approaches for network-on-chip traffic characterization,” in *Proceedings of the 7th IEEE/ACM international conference on Hardware/software codesign and system synthesis*, pp. 461–470, 2009.
- [36] H. Hossain, M. Ahmed, A. Al-Nayeem, T. Z. Islam, and M. M. Akbar, “Gpnocsim-a general purpose simulator for network-on-chip,” in *2007 International Conference on Information and Communication Technology*, pp. 254–257, IEEE, 2007.
- [37] J. Xi and P. Zhong, “A system-level network-on-chip simulation framework integrated with low-level analytical models,” in *2006 International Conference on Computer Design*, pp. 383–388, IEEE, 2006.
- [38] C. Grecu, A. Ivanov, R. Saleh, C. Rusu, L. Anghel, P. P. Pande, and V. Nuca, “A flexible network-on-chip simulator for early design space exploration,” in *2008 1st Microsystems and Nanoelectronics Research Conference*, pp. 33–36, IEEE, 2008.
- [39] P. T. Wolkotte, P. K. Holzenspies, and G. J. Smit, “Fast, accurate and detailed noc simulations,” in *First International Symposium on Networks-on-Chip (NOCS’07)*, pp. 323–332, IEEE, 2007.
- [40] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta, “The splash-2 programs: Characterization and methodological considerations,” *ACM SIGARCH computer architecture news*, vol. 23, no. 2, pp. 24–36, 1995.
- [41] C. Bienia and K. Li, *Benchmarking modern multiprocessors*. Princeton University Princeton, NJ, 2011.
- [42] R. Dick, “Embedded system synthesis benchmarks suite (e3s), 2010,” *URL: <http://ziyang.eecs.umich.edu/~dickrp/e3s>*, 2009.
- [43] J. Hestness, B. Grot, and S. W. Keckler, “Netrace: dependency-driven trace-based network-on-chip simulation,” in *Proceedings of the Third International Workshop on*

*Network on Chip Architectures*, pp. 31–36, 2010.

- [44] L. Eeckhout, K. De Bosschere, and H. Neefs, “Performance analysis through synthetic trace generation,” in *2000 IEEE International Symposium on Performance Analysis of Systems and Software. ISPASS (Cat. No. 00EX422)*, pp. 1–6, IEEE, 2000.
- [45] P. Gratz and S. W. Keckler, “Realistic workload characterization and analysis for networks-on-chip design,” in *The 4th workshop on chip multiprocessor memory systems and interconnects (CMP-MSI)*, pp. 1–10, 2010.
- [46] D. Burger, S. W. Keckler, K. S. McKinley, M. Dahlin, L. K. John, C. Lin, C. R. Moore, J. Burrill, R. G. McDonald, and W. Yoder, “Scaling to the end of silicon with edge architectures,” *Computer*, vol. 37, no. 7, pp. 44–55, 2004.
- [47] P. Gratz, C. Kim, R. McDonald, S. W. Keckler, and D. Burger, “Implementation and evaluation of on-chip network architectures,” in *2006 International Conference on Computer Design*, pp. 477–484, IEEE, 2006.
- [48] G. V. Varatkar and R. Marculescu, “On-chip traffic modeling and synthesis for mpeg-2 video applications,” *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 12, no. 1, pp. 108–119, 2004.
- [49] J.-C. Cano and P. Manzoni, “On the use and calculation of the hurst parameter with mpeg videos data traffic,” in *Euromicro Conference, 2000. Proceedings of the 26th*, vol. 1, pp. 448–455, IEEE, 2000.
- [50] A. Lozhkovskiy, “Simplified calculation of the hurst coefficient by the r/s-analysis method: The coefficient h is equal to the ratio of the relative displacement of two points of the line along the x and y axes,” in *2017 2nd International Conference on Advanced Information and Communication Technologies (AICT)*, pp. 254–257, IEEE, 2017.
- [51] J. Beran, *Statistics for long-memory processes*, vol. 61. CRC Press, 1994.
- [52] W. Willinger, M. S. Taqqu, and D. V. Wilson, “Lessons from“ on the self-similar nature of ethernet traffic”,” *ACM SIGCOMM Computer Communication Review*, vol. 49, no. 5, pp. 56–62, 2019.
- [53] T. Mori, R. Kawahara, and S. Naito, “Analysis of non-gaussian nature of network traffic and its implication on network performance,” *arXiv preprint cs/0209004*, 2002.
- [54] V. J. Ribeiro, R. H. Riedi, M. S. Crouse, and R. G. Baraniuk, “Simulation of nongaussian long-range-dependent traffic using wavelets,” 1999.

- [55] A. Karasaridis and D. Hatzinakos, “A non-gaussian self-similar process for broadband heavy traffic modeling,” in *Global Telecommunications Conference, 1998. GLOBECOM 1998. The Bridge to Global Integration. IEEE*, vol. 5, pp. 2995–3000, IEEE, 1998.
- [56] M. Clausel, F. Roueff, M. S. Taqqu, and C. Tudor, “Wavelet estimation of the long memory parameter for hermite polynomial of gaussian processes\*\*\*,” *ESAIM: Probability and Statistics*, vol. 18, pp. 42–76, 2014.
- [57] A. Lawrance and N. Kottegoda, “Stochastic modelling of riverflow time series,” *Journal of the Royal Statistical Society: Series A (General)*, vol. 140, no. 1, pp. 1–31, 1977.
- [58] M. S. Taqqu, “A representation for self-similar processes,” *Stochastic Processes and their Applications*, vol. 7, no. 1, pp. 55–64, 1978.
- [59] A. Scherrer, *Analyses statistiques des communications sur puce*. PhD thesis, 2006.
- [60] P. Abry and V. Pipiras, “Wavelet-based synthesis of the rosenblatt process,” *Signal processing*, vol. 86, no. 9, pp. 2326–2339, 2006.
- [61] M. S. Taqqu, “Convergence of integrated processes of arbitrary hermite rank,” *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, vol. 50, no. 1, pp. 53–83, 1979.
- [62] J.-C. Breton, I. Nourdin, *et al.*, “Error bounds on the non-normal approximation of hermite power variations of fractional brownian motion,” *Electronic Communications in Probability*, vol. 13, pp. 482–493, 2008.
- [63] C. A. Tudor, F. G. Viens, *et al.*, “Variations and estimators for self-similarity parameters via malliavin calculus,” *The Annals of Probability*, vol. 37, no. 6, pp. 2093–2134, 2009.
- [64] P. Embrechts, *Selfsimilar processes*, vol. 21. Princeton University Press, 2009.
- [65] C. A. Tudor, “Analysis of the rosenblatt process,” *ESAIM: Probability and statistics*, vol. 12, pp. 230–257, 2008.
- [66] J.-M. Bardet and C. Tudor, “A wavelet analysis of the rosenblatt process: chaos expansion and estimation of the self-similarity parameter,” *Stochastic Processes and their Applications*, vol. 120, no. 12, pp. 2331–2362, 2010.
- [67] R. A. Davis, K.-S. Lii, and D. N. Politis, “The rosenblatt process,” in *Selected Works of Murray Rosenblatt*, pp. 29–45, Springer, 2011.

- [68] U. Y. Ogras and R. Marculescu, “Prediction-based flow control for network-on-chip traffic,” in *Proceedings of the 43rd annual design automation conference*, pp. 839–844, 2006.
- [69] E. Nilsson, M. Millberg, J. Oberg, and A. Jantsch, “Load distribution with the proximity congestion awareness in a network on chip,” in *2003 Design, Automation and Test in Europe Conference and Exhibition*, pp. 1126–1127, IEEE, 2003.
- [70] J. Hu, U. Y. Ogras, and R. Marculescu, “System-level buffer allocation for application-specific networks-on-chip router design,” *IEEE Transactions on Computer-Aided Design of integrated circuits and systems*, vol. 25, no. 12, pp. 2919–2933, 2006.
- [71] E. A. da Silva, M. E. Kreutz, and C. A. Zeferino, “Redscarf: an open-source multi-platform simulation environment for performance evaluation of networks-on-chip,” *Journal of Systems Architecture*, vol. 99, p. 101633, 2019.
- [72] Y. J. Yoon, P. Mantovani, and L. P. Carloni, “System-level design of networks-on-chip for heterogeneous systems-on-chip,” in *2017 Eleventh IEEE/ACM International Symposium on Networks-on-Chip (NOCS)*, pp. 1–6, IEEE, 2017.
- [73] J. Harbin and L. S. Indrusiak, “Comparative performance evaluation of latency and link dynamic power consumption modelling algorithms in wormhole switching networks on chip,” *Journal of Systems Architecture*, vol. 63, pp. 33–47, 2016.
- [74] T. T. Le, R. Ning, D. Zhao, H. Wu, and M. Bayoumi, “Optimizing the heterogeneous network on-chip design in manycore architectures,” in *2017 30th IEEE International System-on-Chip Conference (SOCC)*, pp. 184–189, IEEE, 2017.
- [75] S. Khan, S. Anjum, U. A. Gulzari, and F. S. Torres, “Comparative analysis of network-on-chip simulation tools,” *IET Computers & Digital Techniques*, vol. 12, no. 1, pp. 30–38, 2017.
- [76] A. E. Kiasari, Z. Lu, and A. Jantsch, “An analytical latency model for networks-on-chip,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 1, pp. 113–123, 2012.
- [77] V. Catania, A. Mineo, S. Monteleone, M. Palesi, and D. Patti, “Noxim: An open, extensible and cycle-accurate network on chip simulator,” in *2015 IEEE 26th international conference on application-specific systems, architectures and processors (ASAP)*, pp. 162–163, IEEE, 2015.
- [78] X. Bai and A. Shami, “Modeling self-similar traffic for network simulation,” *arXiv preprint arXiv:1308.3842*, 2013.

- [79] C. L. Franzke, T. Graves, N. W. Watkins, R. B. Gramacy, and C. Hughes, “Robustness of estimators of long-range dependence and self-similarity under non-gaussianity,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 370, no. 1962, pp. 1250–1267, 2012.
- [80] A.-M. Rahmani, A. Afzali-Kusha, and M. Pedram, “A novel synthetic traffic pattern for power/performance analysis of network-on-chips using negative exponential distribution,” *Journal of Low Power Electronics*, vol. 5, no. 3, pp. 396–405, 2009.
- [81] E. Pekkarinen, L. Lehtonen, E. Salminen, and T. D. Hämäläinen, “A set of traffic models for network-on-chip benchmarking,” in *2011 International Symposium on System on Chip (SoC)*, pp. 78–81, IEEE, 2011.
- [82] A. Jantsch, “Models of computation for networks on chip,” in *Sixth International Conference on Application of Concurrency to System Design (ACSD’06)*, pp. 165–178, IEEE, 2006.
- [83] L. Tedesco, A. Mello, L. Giacomet, N. Calazans, and F. Moraes, “Application driven traffic modeling for nocs,” in *Proceedings of the 19th annual symposium on Integrated circuits and systems design*, pp. 62–67, 2006.
- [84] A. Khonsari, M. R. Aghajani, A. Tavakkol, and M. S. Talebi, “Mathematical analysis of buffer sizing for network-on-chips under multimedia traffic,” in *2008 IEEE International Conference on Computer Design*, pp. 150–155, IEEE, 2008.
- [85] S. Manolache, P. Eles, and Z. Peng, “Buffer space optimisation with communication synthesis and traffic shaping for nocs,” in *Proceedings of the Design Automation & Test in Europe Conference*, vol. 1, pp. 1–6, IEEE, 2006.
- [86] A. Varga, “Network simulators,” January 2010. Available online at <http://www.omnetpp.org/>.
- [87] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, “On the self-similar nature of ethernet traffic,” in *ACM SIGCOMM Computer Communication Review*, vol. 23-4, pp. 183–193, ACM, 1993.
- [88] D. Cox and L.-R. Dependence, “a review,” *Statistics: An Appraisal, HA David and HT David (Eds.), The Iowa State University Press, Ames, Iowa*, pp. 55–74, 1984.
- [89] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger, “Long-range dependence in variable-bit-rate video traffic,” *Communications, IEEE Transactions on*, vol. 43, no. 234, pp. 1566–1579, 1995.
- [90] B. Mandelbrot and M. Taqqu, “Robust r/s analysis of long run serial correlation.

42nd internat,” *Statist. Inst., Manila*, pp. 1–38, 1979.

- [91] M. S. Taqqu and V. Teverovsky, “Semi-parametric graphical estimation techniques for long-memory data,” in *Athens Conference on Applied Probability and Time Series Analysis*, pp. 420–432, Springer, 1996.
- [92] M. S. Taqqu and V. Teverovsky, “Robustness of whittle-type estimators for time series with long-range dependence,” *Communications in statistics. Stochastic models*, vol. 13, no. 4, pp. 723–757, 1997.
- [93] R. Fox and M. S. Taqqu, “Large-sample properties of parameter estimates for strongly dependent stationary gaussian time series,” *The Annals of Statistics*, pp. 517–532, 1986.
- [94] H. Helgason, V. Pipiras, and P. Abry, “Fast and exact synthesis of stationary multivariate gaussian time series using circulant embedding,” *Signal Processing*, vol. 91, no. 5, pp. 1123–1133, 2011.
- [95] V. Paxson, “Fast, approximate synthesis of fractional gaussian noise for generating self-similar network traffic,” *ACM SIGCOMM Computer Communication Review*, vol. 27, no. 5, pp. 5–18, 1997.
- [96] R. G. Addie and M. Zukerman, “An approximation for performance evaluation of stationary single server queues,” *Communications, IEEE Transactions on*, vol. 42, no. 12, pp. 3150–3160, 1994.
- [97] N. G. Duffield and N. O’connell, “Large deviations and overflow probabilities for the general single-server queue, with applications,” in *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 118, pp. 363–374, Cambridge Univ Press, 1995.
- [98] H. S. Kim and N. B. Shroff, “Loss probability calculations and asymptotic analysis for finite buffer multiplexers,” *IEEE/ACM Transactions on Networking (TON)*, vol. 9, no. 6, pp. 755–768, 2001.
- [99] Y. Cheng, W. Zhuang, and L. Wang, “Calculation of loss probability in a finite size partitioned buffer for quantitative assured service,” *Communications, IEEE Transactions on*, vol. 55, no. 9, pp. 1757–1771, 2007.
- [100] J. Choe and N. B. Shroff, “A central-limit-theorem-based approach for analyzing queue behavior in atm networks,” in *Proceedings of the 15th International Teletraffic Congress*, pp. 1129–1138, ACM, 1997.
- [101] J. Choe and N. B. Shroff, “A central-limit-theorem-based approach for analyzing queue behavior in high-speed networks,” *Networking, IEEE/ACM Transactions on*,

vol. 6, no. 5, pp. 659–671, 1998.

- [102] J. Choe and N. B. Shroff, “Queueing analysis of high-speed multiplexers including long-range dependent arrival processes,” in *INFOCOM’99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, pp. 617–624, IEEE, 1999.
- [103] J. Choe and N. B. Shroff, “Use of the supremum distribution of gaussian processes in queueing analysis with long-range dependence and self-similarity,” *Stochastic Models*, vol. 16, no. 2, pp. 209–231, 2000.
- [104] I. Norros, “On the use of fractional brownian motion in the theory of connectionless networks,” *Selected Areas in Communications, IEEE Journal on*, vol. 13, no. 6, pp. 953–962, 1995.
- [105] B. B. Mandelbrot and J. R. Wallis, “Computer experiments with fractional gaussian noises: Part 2, rescaled ranges and spectra,” *Water resources research*, vol. 5, no. 1, pp. 242–259, 1969.
- [106] K. Park and W. Willinger, “Self-similar network traffic: An overview,” *Self-similar network traffic and performance evaluation*, pp. 1–38, 2000.
- [107] J. Liu, D. Tang, and Y. Cang, “Variations and estimators for self-similarity parameter of sub-fractional brownian motion via malliavin calculus,” *Communications in Statistics-Theory and Methods*, no. just-accepted, pp. 00–00, 2014.
- [108] D. Nualart, *The Malliavin calculus and related topics*, vol. 1995. Springer, 2006.
- [109] J. Feder, *Fractals*. Springer Science & Business Media, 2013.
- [110] D. Brillinger, P. Caines, J. Geweke, E. Parzen, M. Rosenblatt, and M. Taqqu, *New directions in time series analysis*, vol. 46. Springer Science & Business Media, 2012.
- [111] J. Beran, *Long-Memory Processes*. Wiley Online Library, 2013.
- [112] V. Gadre, U. B. Desai, *et al.*, *Multifractal based network traffic modeling*. Springer Science & Business Media, 2012.
- [113] T. Karagiannis, M. Faloutsos, and R. H. Riedi, “Long-range dependence: now you see it, now you don’t!,” in *Global Telecommunications Conference, 2002. GLOBECOM’02. IEEE*, vol. 3, pp. 2165–2169, IEEE, 2002.
- [114] J. Barunik and L. Kristoufek, “On hurst exponent estimation under heavy-tailed distributions,” *Physica A: Statistical Mechanics and its Applications*, vol. 389, no. 18,

pp. 3844–3855, 2010.

- [115] M. Dlask, J. Kukal, and O. Vysata, “Bayesian approach to hurst exponent estimation,” *Methodology and Computing in Applied Probability*, vol. 19, no. 3, pp. 973–983, 2017.
- [116] G. Samorodnitsky, M. S. Taqqu, and R. Linde, “Stable non-gaussian random processes: stochastic models with infinite variance,” *Bulletin of the London Mathematical Society*, vol. 28, no. 134, pp. 554–555, 1996.
- [117] M. Rosenblatt, “Independence and dependence,” in *Proc. 4th Berkeley sympos. math. statist. and prob.*, vol. 2, pp. 431–443, 1961.
- [118] C. Tudor, *Analysis of variations for self-similar processes: A stochastic calculus approach*. Springer Science & Business Media, 2013.
- [119] I. Daubechies, *Ten lectures on wavelets*, vol. 61. Siam, 1992.
- [120] V. Pipiras, “Wavelet-based simulation of fractional brownian motion revisited,” *Applied and Computational Harmonic Analysis*, vol. 19, no. 1, pp. 49–60, 2005.
- [121] A. Chaurasia and V. K. Sehgal, “The mpeg-4 based energy efficient application traffic modelling and synthesis for network-on-chip architecture,” *Sustainable Computing: Informatics and Systems*, vol. 23, pp. 67–79, 2019.
- [122] A. Chaurasia and V. K. Sehgal, “Performance of gaussian and non-gaussian synthetic traffic on networks-on-chip,” *International Journal of Multimedia Data Engineering and Management (IJMDEM)*, vol. 8, no. 2, pp. 33–42, 2017.

---

## List of Publications

---

### Conferences Papers

1. Chaurasia, A., and Sehgal, V. K. (2016, December). Performance of gaussian and non-gaussian synthetic traffic on networks-on-chip. In *2nd International Conference on Computers and Management (ICCM)* (pp. 1-10).
2. Chaurasia A. (2019, June). Performance of Synthetic Rosenblatt Process under Multicore Architecture. In *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)* (pp. 377-381). IEEE.

### Journal Papers

1. Chaurasia, A., and Sehgal, V. K. (2015). Optimal buffer-size by synthetic self-similar traces for different traffics for NoC. *ACM SIGBED Review*, **12(3)**, 6-12.
2. Chaurasia, A., and Sehgal, V. K. (2017). Performance of gaussian and non-gaussian synthetic traffic on networks-on-chip. *International Journal of Multimedia Data Engineering and Management (IJMDEM)*, **8(2)**, 33-42.
3. Chaurasia, A., and Sehgal, V. K. (2019). The MPEG-4 based energy efficient application traffic modelling and synthesis for network-on-chip architecture. *Sustainable Computing: Informatics and Systems*, **23**, 67-79.
4. Chaurasia, A., and Sehgal, V. K. (2019). "Synthetic Traffic Generation for Wavelet based Rosenblatt Process using LSTM for Multicore Architecture". *International Journal of Control & Automation (IJCA)*, **12(5)**, 607-617.
5. Chaurasia, A., and Sehgal, V. K. (2020). Non-Gaussian Traffic Modelling for Multicore Architecture Using Wavelet Based Rosenblatt Process". *IEEE Computer Architecture Letters, IEEE*. (Under review).