Optimal buffer-size by Synthetic Self-similar traces for different traffics for NoC

Amit Chaurasia *
Department of Computer Science & Engineering
Jaypee University of Information Technology
Waknaghat, Solan 173234, H.P.
India
amit.chaurasia@mail.juit.ac.in

Vivek Kumar Sehgal †
Department of Computer Science & Engineering
Jaypee University of Information Technology
Waknaghat, Solan 173234, H.P.
India
vivekseh@acm.org

ABSTRACT
The importance of the traffic modeling in the field of communication became crucial for the optimization of the network resources used in the communication as the quality of service became the bottleneck in the early design of an architecture. In this paper we analyze different parameters for the quality of service by the multicore architecture using the synthetic generated traces for the multimedia applications. The parameters calculated for the multicore architectures with the help of the synthetic self-similar traces helps the designer to choose the optimal resources in the early design process due to the flexibility nature of the traces which is not possible with the real-time applications. The packet loss probability is calculated for different traffic patterns for different architecture against buffer sizes for different traffic patterns.

Categories and Subject Descriptors
C.4 [**PERFORMANCE OF SYSTEMS**]: Modeling techniques

General Terms
Performance

Keywords
Self-similarity, Traffic patterns, Loss-Probability

1. INTRODUCTION
Nowadays, the multicore architectures are used as systems for high performance and thus multicore system shows an increase in the number of processing elements. According to the International Technology Roadmap for Semiconductors (ITRS) 2013, the term *More Than Moore* has coined to the fact of scaling, integration of heterogeneous new functionalities into smart systems became a driving factor for the technology. This trend, diversification in conjunction with miniaturization, led to an increasing complexity in the multicore process estimating more than 1300 core elements[5]. For example, there exists a 1200 cores platform for the protein folding computation [18]. And therefore, the need of bus-based communication has been replaced by parallelism decreasing the power and latency and thereby increasing the throughput. But, it opens many challenges before the designer i.e. for instance the use of optimal network resources. For the better understanding of the system the application based simulation is necessary, the challenge of using the application is that they are not flexible and consumes a lot of

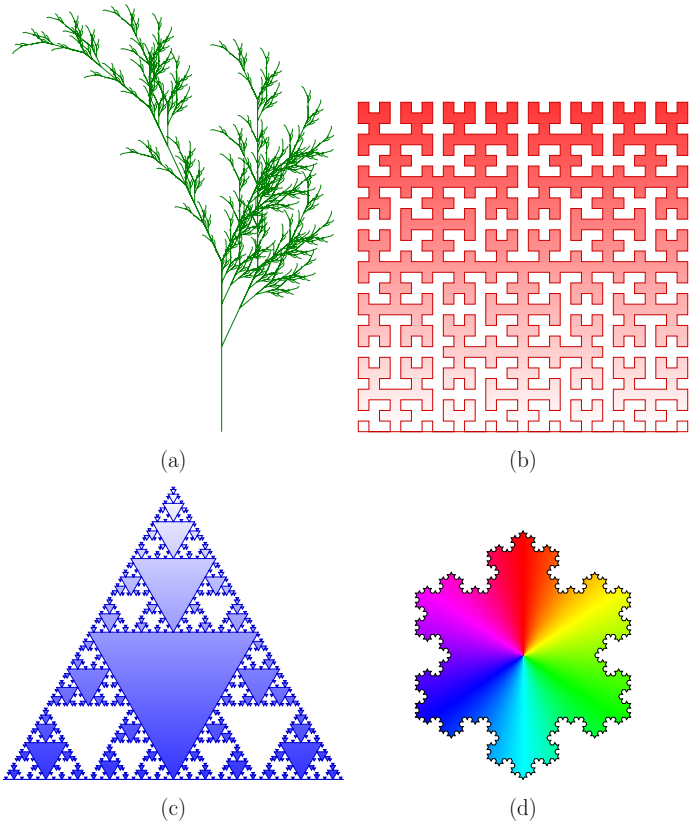


Figure 1: (a) Fractal plant (b) Hilbert curve (c) Sierpinski triangle (d) Koch curve

*He is the main author.
†He is the corresponding author.

time for computation and if traces is used with shorter the length of original trace will fast the simulation and therefore the requirement of traffic modeling comes into light.

Traffic modeling is an important aspect where the optimal use of the resources used in the communication to guarantee the quality of service. During the early years of traditional communication where the number of calls can be represented as the Poisson process where the duration of traffic packet arrivals as exponential variable. But as the greater complexity is added in the communication such as multimedia traffic, which are bursty in nature, modeling of the traffic not only shows the trend of arrivals and numbers but also the variation of bandwidth during communication.

In [4] traffic modeling is done for the multimedia application using the adaptable neural-network architecture, which uses the recursive weight estimation for adapting the network with the original conditions. The first method of self-similar traffic modeling is found in [17] where the self similar traffic is generated for the Ethernet LAN. In [15] the self-similar traffic for the multimedia application for the multicore architecture is generated and the loss probability is calculated considering the infinite buffer system.

In [6] for the finite buffer system, the impact of the arrival process on the loss of the correlation becomes negligible as across the time scale. This paper shows the selection of traffic models such as Markovian or Self-similar under the assumption of correlation horizon. They also find that packet loss is dependent on the arrival rates of the packet. In [12] shows the limitations of ordinary resource allocation procedures due to the presence of self-similar traffic. In [13] the stochastic model used to compute the probabilities for real time allocation of single buffer, and energy saved from the inactive routers, and expected delay in multicore system.

In [1] the limitations of queueing theory and Markov chain methodology to predict the size of the buffer solved by using the characteristics of power-law distribution, the network flow exhibit the scaling and correlation properties, by the presence of energy level and parameter of packet injection rate the origin of self-similarity or long-range dependency can be inferred in NoC traffic.

In [2] presented a new method to capture the non-stationary and multifractal effects on the traffic of NoC, the impact of the packet injection rate over the overflow probability and latencies between node-to-node is calculated.

In this paper, we discuss the impact of self-similar traffic on the different traffic patterns for different architectures. The self-similar traffic is generated using the statistical properties of the multimedia applications [9] which shows the properties of burstiness. The *autocorrelation function* for the generated traffic will not be summable. The *autocorrelation function* decays so slowly that any limit of aggregation will not eliminate the *autocovariance* from the process. The sum of autocorrelation function is shown in eqn. 1 as

$$\sum_{-\infty}^{\infty} r(k) = \infty \tag{1}$$

where $r(k)$ is the *autocorrelation function* with k number of lags. The process showing this dependency also known as *Long Range Dependence process (LRD)*. If in the process *autocorrelation function* is summable then this dependency is known as *Short Range Dependence process (SRD)*.

According to [9], traffic modeling simulation and different benchmarking [14] in order to evaluate a NoC behavior and performance at an early stage in the design process. However, NoC performance estimation in the early design is highly dependent on the type of traffic patterns. Therefore, the traffic selection to present the similarities as for real applications, it is supposed to underline.

We have searched three different classes of traffic patterns [11] i.e. real application traffic patterns, synthetic traffic models & the last one extracts the statistical characteristics from real application for generating more simpler traffic models than the original application traffics.

2. SELF-SIMILARITY

Self-similarity [10] refers to the same characteristics shown at all the distribution at all possible scales. Different examples showing the self-similar are shown in Fig. 1, where four figures of *Fractal plant, Hilbert curve, Sierpinski triangle and Koch curve* [7] showing the self-similar fractals at the finer details of scaling where the finer details shows the same properties from which it is originated and this process is iterated recursively, then it will give the final system of its origin. In terms of traffic this would be analogous to 100ms network bins is aggregated into the 100s network bins and this 100s network bins is aggregated into 100min. network bins. So this is not similar to a Poisson process as the distribution increases which will smooth giving the flat line, i.e. no place for burstiness, whereas if the burstiness shown at the finer level, then the burstiness will appear in the whole distribution and thus it is the good method to model the multimedia applications.

The mathematical representation of the long-range dependence as: let $L = (L_t: t = 0,1,2,\dots)$ be a stationary stochastic process with mean q, and variance σ^2 and autocorrelation function $AC(k)$, $k \geq 0$. L is said to be long-range dependence if

$$AC\,(k) \sim k^{-\beta} L_1\,(t) \qquad as\ k \to\ \infty \tag{2}$$

where $0 < \beta < 1$, $L_1(t)$ is a slowly varying function, that is, $\lim_{t\to\infty} L_1(tw)/L_1(t) \ = 1$, for all $w > 0$ and $\sim$ denotes the condition of *asymptotically close*.

Self-similar processes are measured by the *Hurst parameter (H)* which ranges from $0 < H < 1$. Consider a traffic volume process $S(t)$ at the t-th time, its aggregated process $S^{(m)}$ of S at aggregation level m,

$$S^{(m)}(j) = \frac{1}{m} \sum_{m(j-1)+1}^{mj} S(t) \tag{3}$$

$$S^{(m)} = m^{H-1} S \tag{4}$$

$$S = m^{1-H} S^{(m)} \tag{5}$$

The $S(t)$ is partitioned into a nonoverlapping blocks of size m, where number of blocks is denoted by j and their values are averaged over these blocks. Let $\phi^{(m)}(k)$ denotes the *autocovariance function* for the process $S^{(m)}$.

Definition 2.1. (Second-Order Stationarity) A process is said to be *second-order stationary* if its *autocovariance function* ($\phi(a,b) = E[(S(t)-a)(S(t)-b)]$) is translation invariance i.e.,

$$\phi(a,b) = \phi(a+k, b+k), \tag{6}$$

where $a, b, k \epsilon \mathbb{Z}$.

Definition 2.2. (Second-Order Self-Similarity) A process is said to be *second-order self-similar* if its *autocovariance function* satisfies,

$$\phi(k) = \frac{\sigma^2}{2}\left((k-1)^{2H} - 2(k)^{2H} + (k+1)^{2H}\right) \tag{7}$$

and it is said to be *asymptotically second-order self-similar* if,

$$\lim_{m\to\infty} \phi^{(m)}(k) = \frac{\sigma^2}{2}\left((k-1)^{2H} - 2(k)^{2H} + (k+1)^{2H}\right), \tag{8}$$

where $k \geq 1$ and $0.5 \leq H \leq 1$

By the definition of the *Second-order stationary* we arrived to the definition of *Second-order self-similar*. *Second-order self-similar* is an important property for the network traffic modeling whether it is exact synthesis or asymptotic.

Consider the cumulative process $I(t)$ of the aggregated process $S(t)$ such that $S(t) = I(t) - I(t-1)$ which is itself a self-similar process.

$$I(t) = c^{-H} I(ct) \tag{9}$$

where c is the contraction factor which can dilate or stretch according to its value if $c < 1$ and $c > 1$ respectively, $I(ct)$ is the time scaled version of $I(t)$, and both follows the same distribution.

$$I(t) = t^H I(1) \tag{10}$$

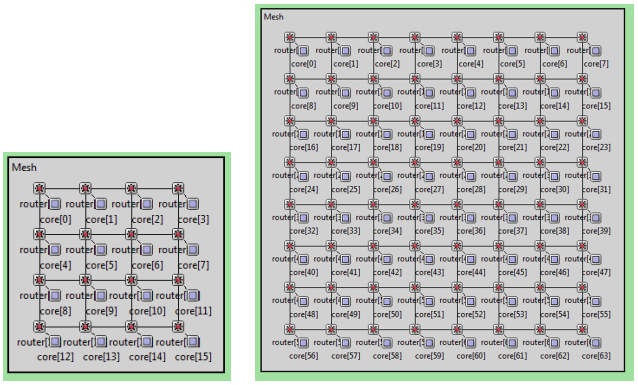
So variance of the aggregated S^m is derivable as,

$$var\left(S^{(m)}\right) = m^{2H-2}\sigma^2. \tag{11}$$

So $I(t)$ is said to be *fractional brownian motion (FBM)* whereas its cumulative process is called *fractional gaussian motion (FGN)* which is an important ingredient for traffic modeling.

3. TRAFFIC PATTERNS

The performance of the system is evaluated on the three traffic patterns to understand messages spatial distribution using metrics such as *end-to-end latency, hop count etc.*. In the *uniform traffic* [3] the destination node is calculated by adding the random number to its source id. The random number is generated using the uniform distribution ranges from 1 to $n-1$, where n is the size of the topology.



(a) (b)

Figure 2: (a) 4×4 Mesh Topology (b) 8×8 Mesh Topology by OMNET++

$$d_i = (s_i + intuniform(1, (\lceil n/2 \rceil - 1)))\ mod\ n \tag{12}$$

where d_i & s_i are the destination id and source id respectively, *intuniform* is a function which returns integer number from the uniform distribution generated from 1 to $n-1$. In *uniform traffic* the source most likely to send equally to each destination, it balances the load that has a very low load balancing. In *tornado traffic* [3] is the fixed source-destination pair where the source id is added with half of the size of the network.

$$d_i = s_i + (\lceil n/2 \rceil - 1)\ mod\ n \tag{13}$$

In the *complement traffic* [3] is another fixed source-destination pair in which the destination id depends on the id of the source id.

$$d_i = \neg\ s_i\ mod\ n \tag{14}$$

The destination id is the negation of the source id. All these three network patterns will not give the same source and destination, i.e. the core which is generating the packets will not send back to its own address. This is the reason behind for choosing these three traffics since in the multicore architecture the core will not send the packets to itself. The different cores have different functions to perform so after processing, the intermediate results are transferred to other cores. For example, in the multimedia application processing, the encoding, the decoding functions and the shared functions are assigned to different cores of the architecture by using some optimized mapping algorithm. These cores perform the functions and communicate to other cores for complete processing of the applications. Hence we have taken these three traffic patterns to measure the performance of the synthetic traces on the multicore architecture.

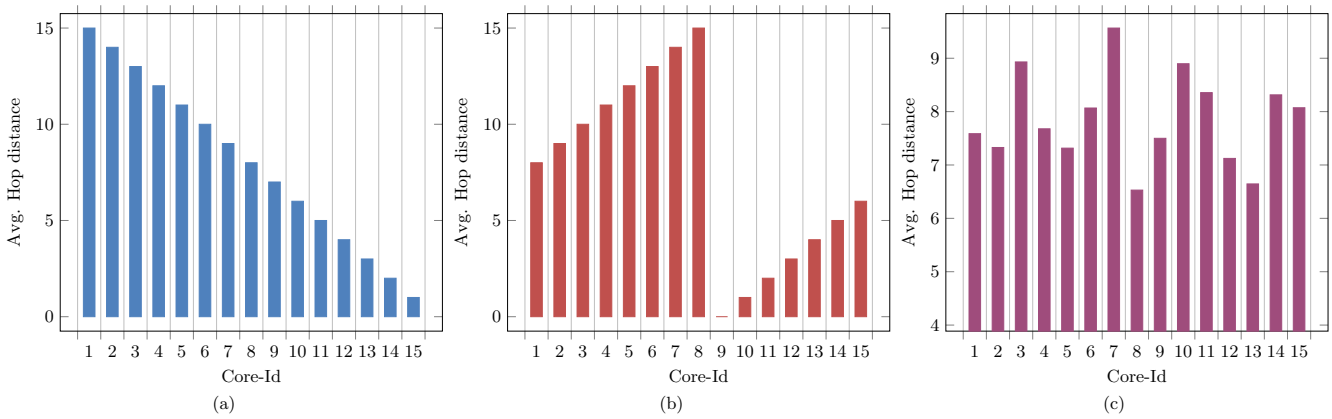


Figure 3: Average Hop distance for 4×4 Mesh network for the traffic (a) Complement (b) Tornado (c) Uniform

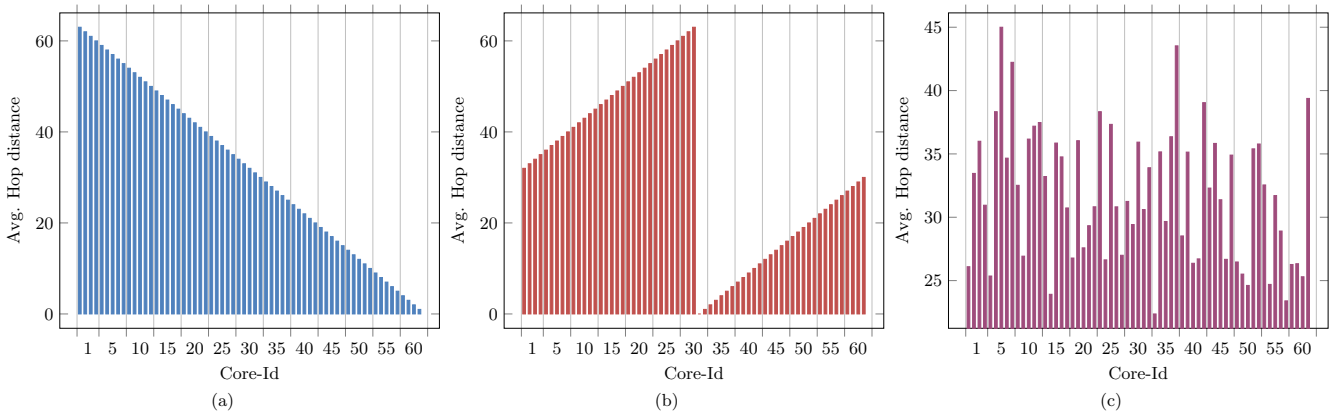


Figure 4: Average hop distance for 8×8 Mesh network for the traffic (a) Complement (b) Tornado (c) Uniform

4. RESULTS

For the performance we have taken two instances of Mesh topology one is 4×4 (see Fig. 2a) and the other is 8×8 (see Fig. 2b) network. The platform for the simulation is done on the OMNET++ [16] where the parameters used for simulation is shown in Table 1. For changing the buffer size, the change is done in the maximum number of queued packets, where 8, 16, & 32 are assigned for 256, 512 & 1024 bytes buffer sizes respectively.

The analytical overflow probability of the packets is calculated using well-known Empty Buffer Approximation (EBA) method which model queueing systems with traffic flows which can help in making the performance modeling problem tractable [8].

$$P\{Q_k > x\} \approx \frac{e^{-\frac{1}{2}U_k(t_k)}}{\sqrt[4]{2\Pi\left(1+\sqrt{U_k\left(t_k\right)}\right)^2}}, \qquad (15)$$

where

$$U_k\left(t_k\right) = \frac{\left(-x+\left(c_k-m_k\right)t\right)^2}{a_k * m_k * t^{2H}}$$

Parameters	*Value*
No. of Virtual Channel (VC)	2
Flit Size	4 bytes
Start time	1ns
Message Length	4 pkts
Packet Length	8 flits
Flit arrival delay	2ns
Flits per VC	1
Arbitration	false
Routing	XY routing
Simulation duration	2μs

Table 1: Parameters for Simulation

The c_k is the service capacity whereas m_k is the mean arrival rates of the services, x denotes the buffer size, a_k is the variance of the cumulative series process, H is the hurst parameter & t_k denotes the cumulative time of traffic flows at time t. In Fig. 7 the analytical overflow probability is plotted against the buffer size using the eqn. 15.

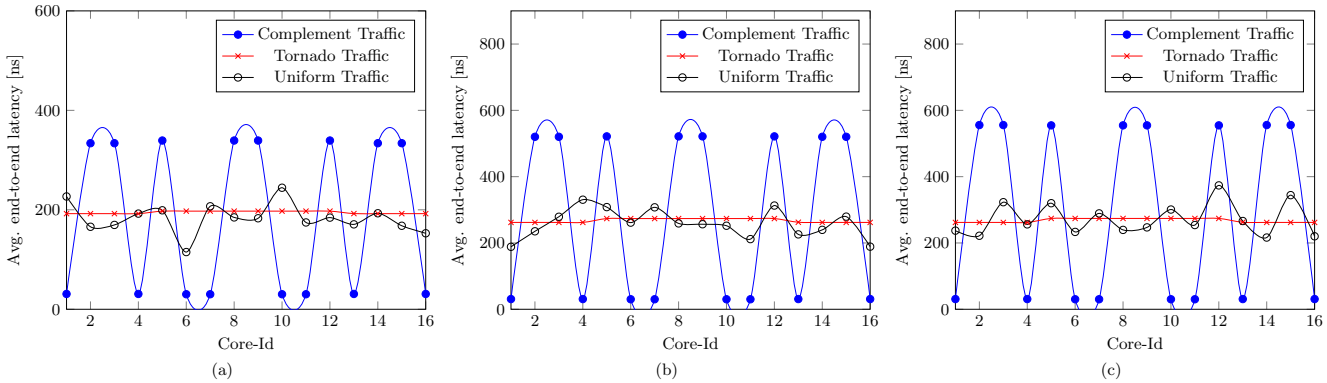


Figure 5: Average End-to-End latency for different traffic for 4×4 Mesh network for buffer size (a) 256 (b) 512 (c) 1024

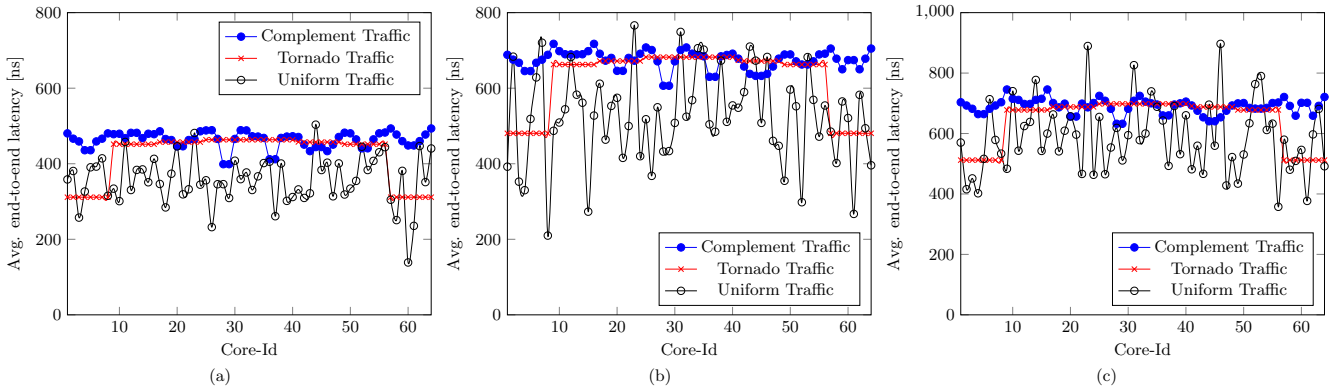


Figure 6: Average End-to-End latency for different traffic for 8×8 Mesh network for buffer size (a) 256 (b) 512 (c) 1024

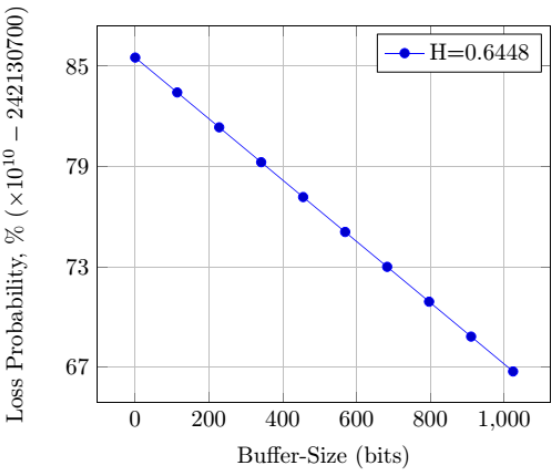


Figure 7: Analytical Buffer-Loss Probability for 4×4 Mesh network

In the Fig. 3 & 4, the average hop distances is plotted for each core, the plot shows similar pattern of variation of the distance in both architectures for similar traffic patterns. For 4×4 mesh topology the maximum average hop count for *Uniform traffic* is 10 for the core-id 7, whereas the maximum average hop count for *Tornado traffic* & *Complement traffic* is 15 for the core-ids 1 & 8 respectively, and for 8×8 mesh topology the maximum average hop count for *Uniform traffic* is 45 for the core-id 7, whereas the maximum average hop count for *Tornado traffic* & *Complement traffic* is 63 for the core-ids 1 & 32 respectively.

The average end-to-end latency is plotted for both the architecture for all three traffic in the Fig. 5 and 6. The end-to-end latency is the time taken by the packet from source to destination. In all three different buffer sizes, the pattern of latency is quite similar for each traffic pattern. The simulation is done on three buffer sizes of 256, 512 & 1024 bytes on the three traffic pattern on *Uniform traffic*, *Tornado traffic* & *Complement traffic*.

$$\psi_{end-to-end} = N\left[\psi_{trns} + \psi_{proce} + \psi_{propag} + \psi_{queue}\right], \tag{16}$$

where ψ_{trns} is the transmission delay, ψ_{proce} is the processing delay, ψ_{propag} is the propagation delay, ψ_{queue} queuing delay & N is the number of link, which is nothing but *number of routers*. The latencies for different traffic shows that the *complement traffic* shows higher latency and varying because the variation of the source-destination pair, whereas the latency for *tornado traffic* is quite smooth and varying parameter quite small, the variability of the source-destination pair is small and hence the latency is low, whereas for the *Uniform traffic* shows hip-hop variation not as like in the

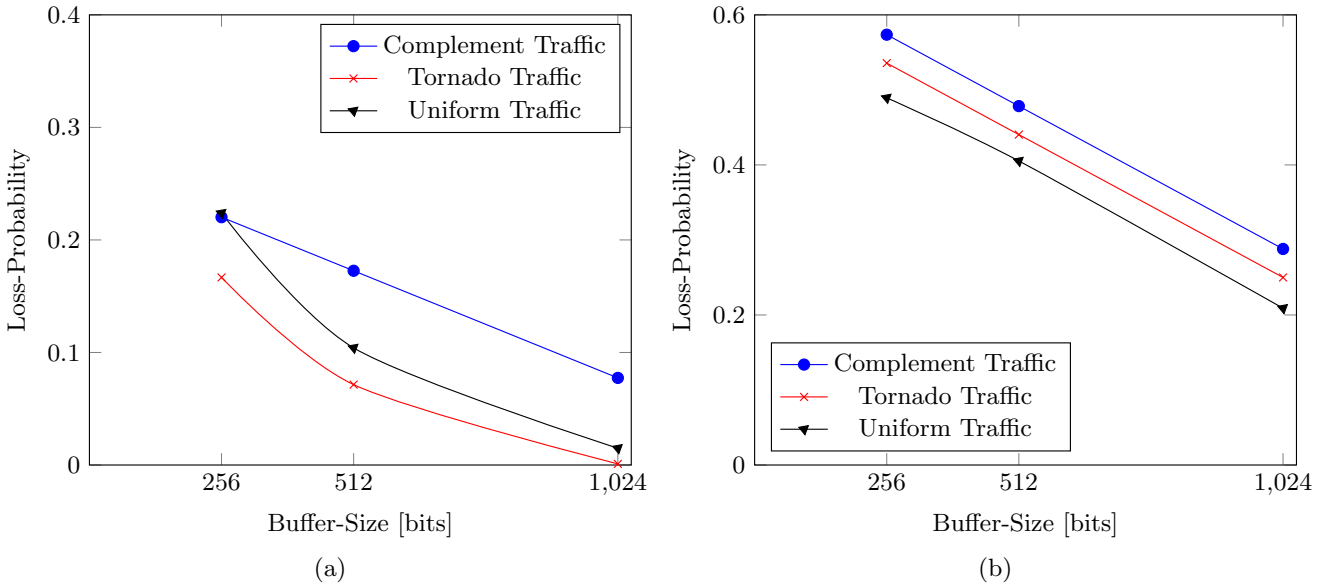


Figure 8: Average Buffer-Loss Probability for different traffic for (a) 4×4 Mesh network (b) 8×8 Mesh network

complement traffic.

The average buffer-loss probability is plotted in Fig. 8 for both architecture against the buffer sizes. The buffer-loss probability is the number of packets lost to total number of packets generated. In both the plot loss-probability decreasing as the size of the buffer is increased showing the inverse relation between the buffer size and the loss probability. As from the figure it is seen that the packet loss is higher in the 8×8 mesh topology than the 4×4 mesh topology. The one of the factor which decides the nature of loss is latency in given in Eq. 16, whereas hop distance also decides the latency, if the hop distance is greater for a core than its latency will be higher.

In Fig. 8 initially the hop distance is greater for both architectures hence for the *Complement traffic* is higher, for *Uniform traffic* shows different characteristics in both the plot i.e. in Fig. 8(a) the *Uniform traffic* has a higher loss probability as compared to *Tornado traffic* whereas in Fig. 8(b) it is reversed. The explanation for this behavior is the hop distance as in the Figs. 3 & 4, since *Uniform traffic* is a random traffic its maximum hop distance shows at core-id 7 (i.e. at the middle) whereas for 8×8 it shows at core-id 7 (i.e. at the first quarter).

5. CONCLUSION

We have presented the comparison of hop distances, end-to-end delay and packet loss probability for the three traffic patterns *Uniform traffic*, *Tornado traffic* & *Complement traffic* for two architectures of mesh topology of 4×4 & 8×8. Our simulation on the multicore architecture based on the self-similar traces for multimedia applications generated synthetically using the statistical properties of video applications. The flexible nature of synthetic generated trace helps in the fast and accurate simulation, which is not possible with real time applications.

The parameters calculated above helps in the early design of architecture and the choice of optimal selection of optimal communication resources. The loss probability shows the nature of buffer under different network pattern circumstances of different architecture and the optimal selection of resources makes the design saves energy by no dropping of packets and efficient architecture by saving extra area taken by unused buffer.

6. REFERENCES

[1] 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*, pages 461–470. ACM, 2009.

[2] P. Bogdan and R. Marculescu. Workload characterization and its impact on multicore platform design. In *Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*, pages 231–240. ACM, 2010.

[3] W. J. Dally and B. P. Towles. *Principles and practices of interconnection networks*. Elsevier, 2004.

[4] A. Doulamis, N. Doulamis, and S. Kollias. An adaptable neural-network model for recursive nonlinear traffic prediction and modeling of mpeg video sources. *Neural Networks, IEEE Transactions on*, 14(1):150–166, Jan 2003.

[5] P. Gargigni. Available online at `http://www.itrs.net/`.

[6] M. Grossglauser and J.-C. Bolot. On the relevance of long-range dependence in network traffic. *Networking, IEEE/ACM Transactions on*, 7(5):629–640, Oct 1999.

[7] M. Hazewinkel. *Encyclopaedia of mathematics, supplement III*, volume 13. Springer Science & Business Media, 2001.

[8] X. Jin and G. Min. Modelling and analysis of priority

queueing systems with multi-class self-similar network traffic: a novel and efficient queue-decomposition approach. *Communications, IEEE Transactions on*, 57(5):1444–1452, 2009.

[9] 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*, 28(1):3–21, 2009.

[10] K. Park and W. Willinger. *Self-similar network traffic and performance evaluation*. Wiley Online Library, 2000.

[11] 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*, page 2. ACM, 2013.

[12] Z. Sahinoglu and S. Tekinay. On multimedia networks: self-similar traffic and network performance. *Communications Magazine, IEEE*, 37(1):48–52, Jan 1999.

[13] V. Sehgal. Markovian models based stochastic communication in networks-in-package. *Parallel and Distributed (TPDS) Systems, IEEE Transactions on*, 2014.

[14] 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*, pages 104–116. IEEE, 2006.

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

[16] A. Varga. Network simulators, January 2010. Available online at `http://www.omnetpp.org/`.

[17] W. Willinger, M. Taqqu, R. Sherman, and D. Wilson. Self-similarity through high-variability: statistical analysis of ethernet lan traffic at the source level. *Networking, IEEE/ACM Transactions on*, 5(1):71–86, Feb 1997.

[18] 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*, pages 1–6. IEEE, 2014.