www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 5 Issue 10 Oct. 2016, Page No. 18557-18563

A Study on Performance Modelling and Analysis of Network on Chip under M-Port N-Tree Bursty Traffic

A.Malathi[1*], *M.Theinmozhi*[2*]
1*Assistant professor, Department of Computer Science, Government Arts College, Coimbatore.
2* Research scholar,Master of Philosophy,Department of Computer Science, Government Arts College, Coimbatore.
E-mail:malathi.arunachalam@yahoo.com,theinmozhi333@gmail.com

Abstract— **Physical constrains of integrated circuits (commonly called chip) in regards to size and finite number of wires, has made the design of System-on-Chip (SoC) more interesting to study in terms of finding better solutions for the complexity of the chip-interconnections. The SoC has hundreds of Processing Elements (PEs), and a single shared bus can no longer be acceptable due to poor scalability with the system size. Networks on Chip (NoC) have been proposed as a solution to mitigate complex on-chip communication problems for complex SoCs. They consist of computational resources in the form of PE cores and switching nodes which allow PEs to communicate with each other. Performance modelling and analysis has great theoretical and practical importance. This research is devoted to developing efficient and cost effective analytical tools for the performance analysis and enhancement of NoCs with m-port n-tree topology under bursty traffic. Therefore a new analytical model is developed to investigate the performance of NoCs with the m-port n-tree topology under bursty traffic. Even though it is broadly proved in practice that fat-tree topology and its varieties result in lower latency, higher throughput and bandwidth, still most studies on NoCs adopt Mesh, Torus and Spidergon topologies. The analytical results and those obtained from extensive simulation experiments have shown a good degree of accuracy for predicting the network performance under different design alternatives and various traffic conditions.**

Recent measurement studies have strongly verified that the traffic generated by many real-world applications in communication networks exhibits bursty and self-similar properties in nature and the message destinations are uniformly distributed. NoC's performance is generally affected by different traffic patterns generated by the processing elements. As the first step in the research, a new analytical model is developed to capture the burstiness and self-similarity characteristics of the traffic within NoCs through the use of Markov Modulated Poisson Process. The performance results of the developed model highlight the importance of accurate traffic modelling in the study and performance evaluation of NoCs.

Having developed an efficient analytical tool to capture the traffic behaviour with a higher accuracy, in the next step, the research focuses on the effect of topology on the performance of NoCs. Many important challenges still remain as vulnerabilities within the design of NoCs with topology being the most important. Therefore a new analytical model is developed to investigate the performance of NoCs with the m-port n-tree topology under bursty traffic. Even though it is broadly proved in practice that fat-tree topology and its varieties result in lower latency, higher throughput and bandwidth, still most studies on NoCs adopt Mesh, Torus and Spidergon topologies. The results
iii **gained from the developed model and advanced simulation experiments significantly show the effect of fat-tree topology in reducing latency and increasing the throughput of NoCs.**

In order to obtain deeper understanding of NoCs performance attributes and for further improvement, in the final stage of the research, the developed analytical model was extended to consider the use of virtual channels within the architecture of NoCs. Extensive simulation experiments were carried out which show satisfactory improvements in the throughput of NoCs with fat-tree topology and VCs under bursty traffic. The analytical results and those obtained from extensive simulation experiments have shown a good degree of accuracy for predicting the network performance under different design alternatives and various traffic conditions.

Keywords- ***Network On Chip, Bursty traffic, Virtual Channels, Latency***

I. INTRODUCTION

Networks on Chip have been proposed as a solution to mitigate complex on-chip communication problems. NoCs are composed of PE cores (or processing elements), which are interconnected by on-chip switching fabrics. A step in the design process of NoCs is hardware virtualization, which is mapping the PE cores on to the tiles of chips [69-71].

The communication among the PE cores greatly affects the performance and power consumption of NoCs, which itself is closely related to the placement of PEs on to the tiles of the network.

To overcome the problems of scalability, complexity and performance, Networks-On-Chips (NoCs) have been proposed as a promising solution to reduce the overheads of buses and MPSoCs connected by means of general-purpose communication architectures [2, 72-74]. This technology provides a solution to overcome the limitations that exist in the traditional bus-based interconnection technologies. The scalability of NoC has made it ideal for larger designs. Since the NoC technology builds on top of the latest evolutions of bus architectures, and uses packet-based communication paradigms for the means of communication, it can overcome many of the issues related to the interconnect fabric design better than shared buses architectures.

Most of the work has been done in analytical models on the performance study in the area of NoCs has been focus on specific topology and Poisson traffic only [51, 55, 75]. The well-known Markov Modulated Poisson Process has been used for the purpose of its ability to model the time varying arrival rate and correlation between inter arrival time. In this

DOI: 10.18535/ijecs/v5i10.38

chapter the analytical model has been implemented investigate the performance of NoC by employed m-port n-tree topology under bursty traffic.

II. The m-port n-tree NOC

Network on Chip topology identifies the physical organization of the interconnection network. It describes how nodes, switches and links are connected to each other. NoC topologies are classified into two classes based on the type of connectivity. On one hand, there is the direct connect network, where each node is connected to at least one core (PE), and on the other hand is the indirect connect network, where the node is not connected to any core (PE). Most of NoC employs regular topology, since it is easy to be laid out on a chip surface. Fat-tree has been the most popular networks topology for the past fifty years, and has been adopted for NOC domain. Commercial machines and many research prototypes have adopted the variations of different kinds of fat-tree. Bisection bandwidth is one of the important properties for fat-tree, which scales linearly with the network size[76]. In this work we focus on fat tree formed by m-port n-tree topology.

Comparison between a different network topologies has been proposed by Pandel [27], considering throughput, latency, and energy consumption.

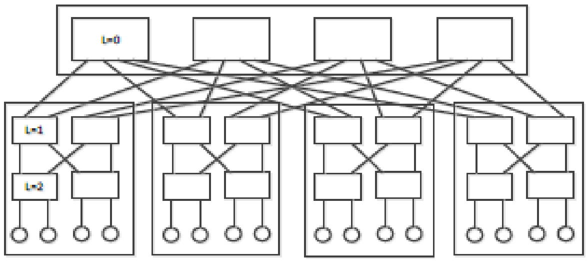
An m-port n-tree topology contains N processing nodes and Nsw switches.The processing node is known as n-tuple PN (P_0, P_1, P_{n-1}), where P ∈ {0, 1,… $m-1$} {0, 1, …., $(m/2)-1\}^{n-1}$ and it can be calculated by

$$N = 2\,(m/2)^{n} \qquad (2.1)$$

The communication switch known as n-tuple SW (w_0, w_1, w_2, … . . , w_{n-2}, l) L is the level of the switch, where l ∈ {0,1,2, … . , $n-1$}, and w ∈{0,1, … . , $(m/2)-1\}^{n-1}$ and can be calculated by

$$N_{sw} = (2n-1)\,(m/2)^{n-1} \qquad (2.2)$$

Figure 1: 4-port 3-tree technology



III. Traffic Analysis

Traffic information is an important key to evaluate the network performance at each channel. This section presents detailed information regarding the traffic on each channel in the m-port n-tree NoC. The MMPP is a doubly stochastic process with an arrival rate governed by m-state irreducible continuous time Markov chain [44]. The 2-state MMPP has been broadly used for its simplicity, ability to capture the time-varying arrival rate and correlation between intra-arrival times. Additionally the operations of superposition and splitting are highly recommended to be used by the MMPP traffic. These two operations of MMPPs give rise to a new MMPP [44]. The success of the aforementioned features makes the MMPP more attractive for traffic generated by multimedia application [45, 46].

IV. Assumptions of the model

The model uses some assumptions that are widely used in the literature [44, 77-80]:
1. Nodes generate correlated traffic, which follows an independent MMPP [81] whose infinitesimal generator ⋀ , and Ɏs is the rate matrix as given in the previous section.
2. Message length (Lm flits).
3. Messages generated by the source nodes are sent with probability ϕ.
4. The network switches are input buffered and each channel is associated with a single buffer.
5. Deterministic routing is employed, with the shortest path routing algorithm.

Table 4-1: Key notations used in the derivation of the model.

WI,j	Blocking time that message acquires a channel at stage s
$P1s,j$	Probability that the message is blocked at this stage s
WpI,j	Waiting time by the message to acquire a channel
m	Number of port
n	Number of tree
W	The waiting time at the source node
N	Number of node
Nsw	Number of switch
αnet	Network latency
βnet	Transmission time
j	Number of link (ascending or descending link)
S	Network stage to calculate service time

V. The Analysis Method

The mean message distance mainly is affected by the traffic pattern, the average number of links that message needs to reach its destination. The probability of a message traversing $2k$ links (one k link in ascending phase and another k link in descending phase) to reach its destination is Pk. Different values of Pk could produce different distributions of message destinations and, thus, different mean message distances. In [5] the number of nodes at distance $2n$ is $(m/2)$−1$(m-1)$ in m-port n-tree topology, thus Ᵽk is used:

$$P_k = \begin{pmatrix} (m/2-1)\,(m/2)^{k-1})/N-1 & for\ k=1,2\ \ldots, n-1 \\ (m-1)\,(m/2)^{k-1/}N-1 & for\ k=\mathrm{n} \end{pmatrix}$$

(5.1)

Message latency can be defined as the amount of time that message takes to traverse through the network to reach its destination including any time spent buffered at the source node, to compute mean message latency L in this network, it comprises of three factors: the mean network latency, T, the mean waiting time seen by messages at the source node , W, and the average time for tail flit to reach its destination, R.

Therefore, the mean message latency, L, can be written as:

$$L = T + W + R \qquad (5.2)$$

Moreover, each message generated through any of the source node to be sent with mean message distance, can be given by

$$d = \sum_{k=1}^{n} 2\mathrm{jp}^{k} n^{k}$$

The traffic arriving at the network channel is t , the number of nodes is N and the messages generated by these nodes are sent to $4nN$ channels in the network, thus tsa is:

$$tsa = d/4n \qquad (5.3)$$

5.1 **Mean Network Latency**

Each message may travel to a different number of nodes to reach its destination by taking into account the transmission delay of $2j$ link. The location of the switches between the source and destination node labeling as the network stages. The first stage starts numbering from the stage next to the source node (stage 0), and goes up till it is closer to the destination node. In our topology (m-port n-tree) the number of stages to travel $2j$ link is $\mathrm{S} = 2j - 1$. Determine the service time experienced by the message at the final stage first and then carried out backward to the first stage. Thus, the service time experienced by $2j$ link at $S - 1$ stage is:

$$Ts = Lm\ ts\ (4.4) \qquad (5.1.1)$$

Where t_s is a type of connection in this topology node to switch or switch to node, and switch to switch. The ts can be calculated as $t_s = 0.5\alpha_{\mathrm{net}} + F_m\ \beta_{\mathrm{net}}$, where α_{net} is network latency, and β_{net} is the transmission time of one byte, and L_m is the message length in flits. At internal stages $0 \le s \le S - 2$ the service time might be more based on channel and would be idle when the channels of subsequent stages are busy. The service time at internal stages can be found as:

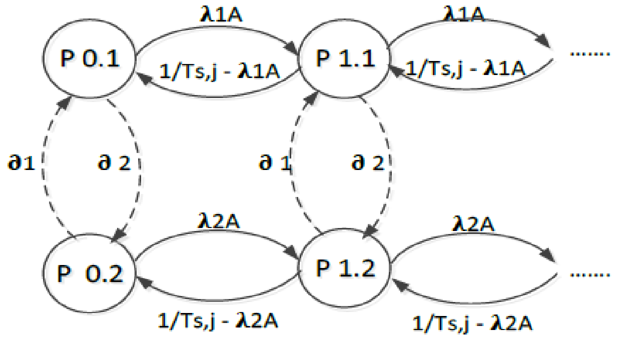
$$T_{s,j} = \sum_{l=k+1}^{k=1} \left(W_{I,j} + Mt_{s1}\right) \quad 0 \le k \le K - 2 \qquad (5.1.2)$$

Where $ts1$ is a switch to switch connection, and can be calculated as $ts1 = \alpha_{\mathrm{sw}} + L_m\beta_{\mathrm{net}}$, where α_{sw} is switch latency. Also $W_{I,j}$ blocking time that message acquires a channel at stage s, can be calculated by the waiting time of the message to acquire a channel when blocking occurs $Wp_{I,j}$, and the probability that the message is blocked at this stage $P1_{s,j}$. We can get:

$$Wp_{I,j} = W_{I,j}\ P1s{,}j \qquad (5.1.3)$$

To calculate the $Pb_{s,j}$ first it is necessary to compute the joint probability $P_{a,b}$ using a bivariate Markov chain shown in Figure 2: Bivariate Markov Chain for Determining Blocking Probability , state $\mathrm{P}_{a,b}$where a donates that channel busy or idle and MMPP(2)$_A$ is in state b..The transition rate out of state $\mathrm{P}_{a,b}$ to P_{a+1},b is $\lambda\ _A$, where $\lambda\ _A$ is the traffic arrival rate on the network channel when the MMPP(2)$_A$ is in state b, the rate from P_{a+1},b to $\mathrm{P}_{a,b}$ is $1/T_{s,j} - \lambda\ _A$. The reduction of $\lambda\ _A$ is used to account for the arrival of message while a channel in this state [12].

Figure 2: Bivariate Markov Chain for Determining Blocking Probability



The arrival rate $\lambda\ _A$ can be found from rate matrix $\Lambda_{\boldsymbol{A}}$ while the $\partial\ _A$ is a transition rate given by infinitesimal generator $\boldsymbol{Q}\ _{\boldsymbol{A}}$**.** In the steady state the model yields the following:

$$\left[\begin{array}{l}
(\lambda\ _{1A} + \partial\ _{2A}) = \partial\ _{1A}\ \mathrm{P}_{1,2} + \left(\dfrac{1}{T_{s,j}} - \lambda\ _{1A}\right)\mathrm{P}_{1,1} \\
(\lambda\ _{1A} + \partial\ _{1A}) = \partial\ _{2A}\ \mathrm{P}_{0,1} + \left(\dfrac{1}{T_{s,j}} - \lambda\ _{1A}\right)\mathrm{P}_{1,2} \\
\left(\dfrac{1}{T_{s,j}} - \lambda\ _A + \partial\ _{2A}\right)\ \mathrm{P}_{1,2} = \lambda\ _{1A}\mathrm{P}_{0,1} + \partial\ _{1A}\mathrm{P}_{1,1} \\
\left(\dfrac{1}{T_{s,j}} - \lambda\ _{1A} + \partial\ _{1A}\right)\ \mathrm{P}_{1,2} = \lambda\ _{1A}\mathrm{P}_{0,2} + \partial\ _{2A}\mathrm{P}_{1,1} \\
\displaystyle\sum_{a=0}^{1}\sum_{b=1}^{2} \mathrm{P}_{1,b} = 1
\end{array}\right.$$

(5.1.4)

Solving the above system of equation yields the probability $P1_{s,\,j}$.

To calculate the waiting time WI,, the channel at source node is modeled as an MMPP/G/1 queuing system, according to [10].

The waiting time can be calculated as:

$$W_a = \frac{1}{2(1-\rho)}\left[2\rho + \lambda_{tot}h_2 - 2h_1((1-\rho)g + h_1\pi\Lambda_A)(Q\ _A + e\pi)^{-1}\lambda\right.$$

(5.1.5)

$$W_{I,} = 1/\rho(W_a - 1/2\lambda_{tot}h_2) \qquad (5.1.6)$$

Where, the h_1 *and* h_2 are the first two moments of the service time on the network channel, which can be determined from Laplace-Stieltjes transform [82].

$\boldsymbol{\rho}$ is the traffic intensity, and can be calculated by $\rho = h_1\ \lambda_{tot}$ where λ_{tot} is the mean arrival rate at network channels, and it is equal to $\lambda_{tot} = \pi\ \lambda$, where π is the steady-state vector of MMPP , and $\lambda =$ Ɏ$_s e$the column unit vector of length 2 $e = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$. The algorithm for calculating the matrix, g has been described in [10].

The average of all possible destinations of a message in the network is the network latency

$$\mathrm{T} = \sum_{j=1}^{n} P_{\mathrm{kn}}\ \mathrm{T_j} \qquad (5.1.6)$$

Where D_i is equal to the service time of the message at stage s = 0 (D_j =$D_{0,\,j}$).

52.The mean waiting time

At the source node the message is injected into the network with equal probability, therefore the traffic arriving at an injection channel at source node represented by MMPP_{s1} is the fraction of that generated by a source node Fs1. Based on cookbook [44], it is assumed that the resulting

process from the splitting of $MMPP$ has the same underlying Markov chain, such as the original $MMPP$. The infinitesimal generator Λs and the rate matrix Ÿ_s are given by:

$$\Lambda_{s1} = \Lambda_s = \begin{bmatrix} -\partial_{s1} & \partial_{s1} \\ \partial_{s2} & -\partial_{s2} \end{bmatrix} \text{ and } \quad \text{Ÿ}_{s1} = \mathrm{F}_{s1}\text{Ÿ}_s \begin{bmatrix} \lambda_{s1} & 0 \\ 0 & \lambda_{s2} \end{bmatrix}$$

The message experiences before entering the network modeled as a $MMPP/G/1$ queuing system, where the arrival process modeled by $MMPP_{s1}$ and the service time at transmission delay is given by T =$\sum_{j=1}^{n} P_{\mathrm{kn}}$ $\mathrm{T_j}$. Waiting time W at the source node can be calculated using the

method we used in the previous section for network latency T. Thus, the waiting time can be processed as:

$$\mathrm{W1} = \frac{1}{2(1-\rho)}[2\rho + \lambda_t h2 - 2h1((1-\rho)g + h_1\pi\text{Ÿ}_{s1})(\Lambda s1 + e\pi)^{-1}\lambda_{s1} \qquad (5.1.6)$$

$$\mathrm{W} = \frac{1}{\rho}((\mathrm{W}_1 - 0.5\lambda_t h_2) \qquad (5.1.7)$$

Where h_1 and h_2 denote the first and second moments of the service time seen by the message respectively, and can be calculated by differentiating $L_t(s)$ Laplace-Stieltjes Transform, as mentioned above. Also ρ is the trafficintensity, which can be calculated by ρ = h_1 λ_t where λ_{tv} is the mean arrival rate at network channels. It is equal to λ_{tv} = Π λc where π is the steady-state vector of MMPPc, and can be written as

$$\pi = [\pi 1 \text{ , } \pi 2] = \frac{1}{\partial 1\mathrm{s} + \partial 2\mathrm{s}}[\partial 1\mathrm{s} + \partial 2\mathrm{s}],$$

$$\lambda_s = \begin{bmatrix} \lambda_{1s} \\ \lambda_{2s} \end{bmatrix} \text{ , and}$$

e is the column unit vector of length

2 e =$\begin{matrix}1\\1\end{matrix}$, and the algorithm to compute g can be found in [44].

The average time for the tail flit to reach its destination R can be given as:

$$R = \sum_{i=1}^{n}[P_{i,n}\,(\sum_{k=1}^{K-1} t_{cs} + t_{cn})] \qquad (5.1.8)$$

VII.VALIDATION AND ANALYSIS

*I*n this paper the simulation is carried out using OMNeT ++ [63] in order to validate the accuracy of the above analytical model. Various simulation experiments have been performed to validate the model.However, further increase in the simulation time will not change the results

by any significant amount. The message latency can be defined as the mean time from the message being generated at the source node until the last data flit reaches the destination node. Moreover, in the simulation the network

cycle time is defined as the transmission time of a single flit to cross from one node to another. As mentioned the messages generated by the source node are modeled by $MMPP$ with its parameters being the infinitesimal generator Λ_s and the rate matrix Ÿ .

Extensive simulation experiments have been achieved to validate the model for several message lengths, different $MMPP$ traffic inputs and switch size. Yet, for the sake of specific illustration, latency results are presented for the following scenarios: for network topology: 4-port 3-tree the system parameters are set at the following:

1. Message length: Lm=64 and 128 flits;
2. Flits length: Lf = 256, 512 and 1024 bytes
3. Network and switch latency is 0.005 and 0.01 time unit respectively.
4. Λ_s is the infinitesimal generator of $MMPP$ and is set as follows:

$$\Lambda_s = \begin{bmatrix} -0.04 & 0.04 \\ 0.08 & -0.08 \end{bmatrix}, \Lambda_s = \begin{bmatrix} -0.04 & 0.04 \\ 0.09 & -0.09 \end{bmatrix}.$$

$$\Lambda_s = \begin{bmatrix} -0.05 & 0.05 \\ 0.09 & -0.09 \end{bmatrix}, \Lambda_s = \begin{bmatrix} -0.04 & 0.04 \\ 0.06 & -0.06 \end{bmatrix}.$$

$$\Lambda_s = \begin{bmatrix} -0.06 & 0.06 \\ 0.09 & -0.09 \end{bmatrix}. \Lambda_s = | \begin{bmatrix} -0.03 & 0.03 \\ 0.09 & -0.09 \end{bmatrix}$$

$$\Lambda_s = \begin{bmatrix} -0.1 & 0.1 \\ 0.09 & -0.09 \end{bmatrix}.$$

We have decided to use a range of message lengths (64 to 128flits) to evaluate the model; this is because state of the art work [39, 83, 84] show that these message lengths exhibit different saturation points, it is therefore necessary to evaluate the latencies at varying saturations. The flit size is affecting the performance within a network on chip router; unfortunately most of the literatures do not justify the choice of a particular size. Although a recent work by Junghee Lee [85] has shown that it is difficult to select an optimum flit size, it has also been reported that large flit size add overhead

cost these by reducing performance. Since no much work has been done to prove Junghee Lee's observation we have decided to choose a range of flit size (256,512, and 1024) with a view to having optimum performance, and better answer to the question of flit size selection. The proposed network and switch latency (i.e. 0.005 and 0.01 time unit) have been used to get better performance.

The figures below, describe the performance results for the message latency predicted by the analytical model plotted beside those by simulator as a function of traffic rate.

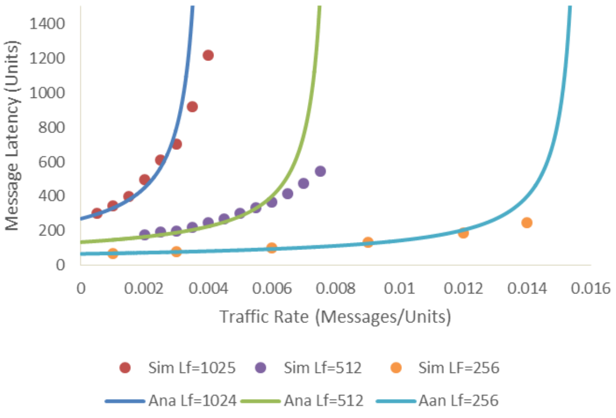


FIGURE 3: LATENCY PREDICTED BY THE MODEL AND SIMULATION: LM=128 $\partial s1$=0.04, $\partial s2$=0.08,

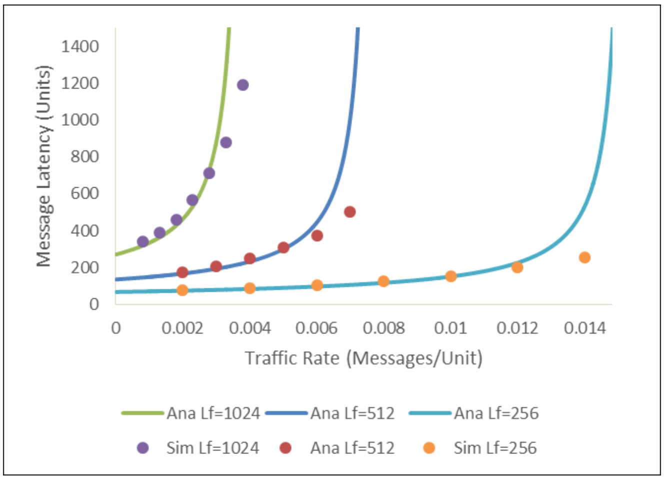


Figure 4: Latency predicted by the model and simulation: Lm=128 $\partial s1$=0.04, $\partial s2$=0.09,

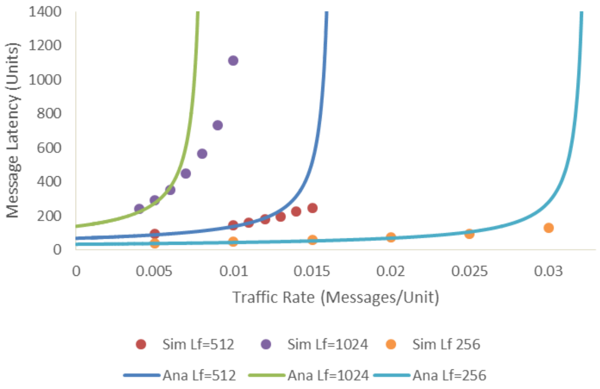


FIGURE 5: LATENCY PREDICTED BY THE MODEL AND SIMULATION: LM=64 $\partial s1$=0.05, $\partial s2$=0.09,

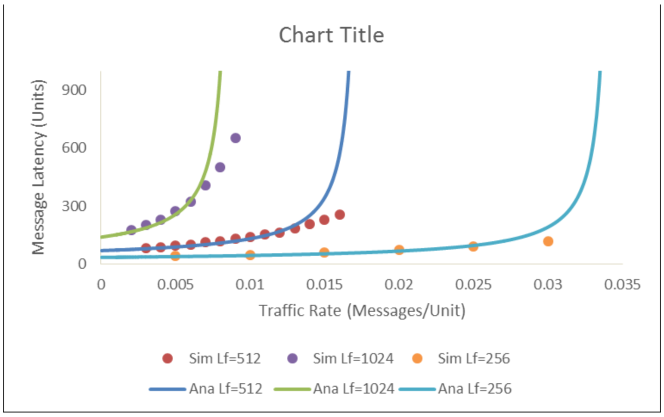


Figure 6: Latency predicted by the model and simulation: Lm=64 $\partial s1$=0.04, $\partial s2$=0.06,

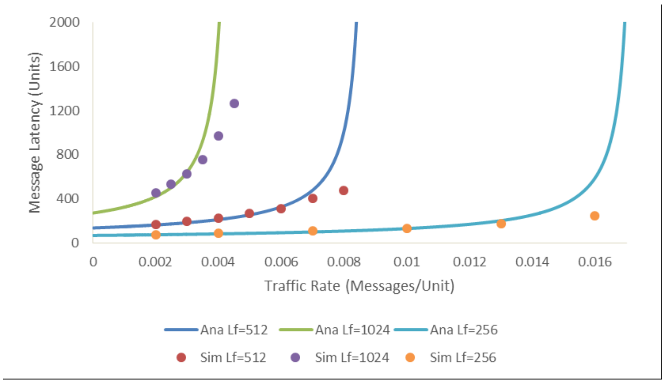


Figure 7: Latency predicted by the model and simulation: Lm=128 $\partial s1$=0.06, $\partial s2$=0.09,

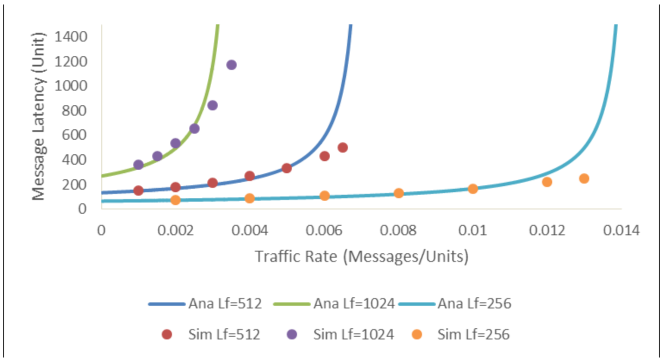


Figure 8: Latency predicted by the model and simulation: Lm=128 $\partial s1$=0.03, $\partial s2$=0.09,

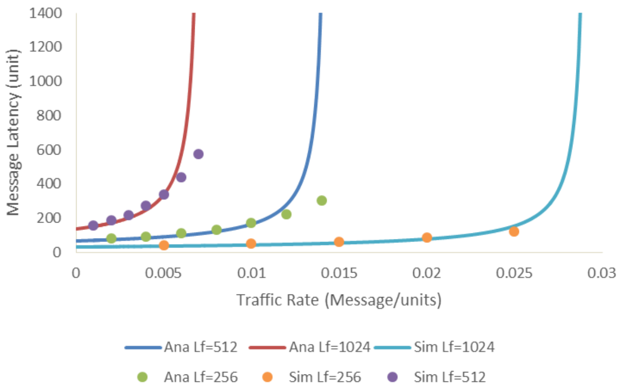
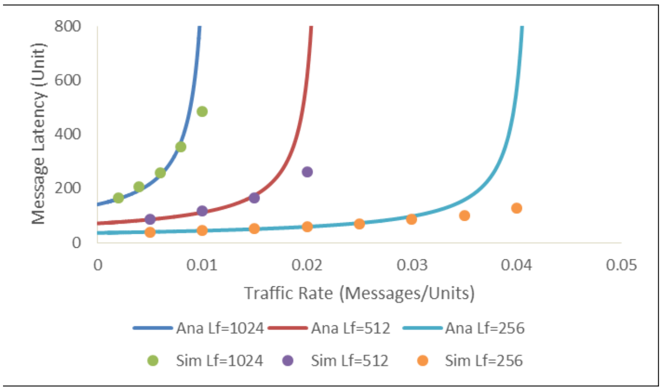


Figure 9: Latency predicted by the model and simulation: Lm=64 $\partial s1$=0.03, $\partial s2$=0.09,



DOI: 10.18535/ijecs/v5i10.38

Figure 10: Latency predicted by the model and simulation: Lm=64 ∂_{s1}=0.1, ∂_{s2}=0.09,

The horizontal and the vertical axis in those figures represent the traffic rate λ_s at which the node message is injected into the network when the $MMPP$ is in state 1, and the mean message latency obtained from the above model. To make figures clear we have purposely set the arrival rate λ_{s2} at state 2 equal to zero, because we need to use three dimensional graphs to demonstrate the results. As can be seen, results represented in those figures show that the obtained message latency for the analytical model closely match those obtained from the simulation. It can be concluded that the results produced by the proposed model are accurate in the steady state. Furthermore, the result from the analytical model are shown as a curve for all possible values of traffic generation rates until reach the saturated point. When the utilization of the system becomes one or greater the network enters the saturation region. Additionally, the latency increases as the traffic rate increases.

As the network approaches the saturation point, Some divergence appears from the simulation result with respect to the analytical result, however this is not a region of concern when performing network performance test. Rather the steady state regions are the determinants of the success of the performance [86]. After validation of the accuracy of the analytical model, we need to use this to investigate the effects of the bursty traffic with different degrees of traffic burstiness and correlations imposed by $MMPP$ input parameters on the network on chip. We can find that the bursty traffic reduces the network performance significantly, since latency increases, especially under moderate and heavy traffic loads. Moreover, the maximum throughput that the network is able to support decreases when subject to the bursty traffic.

To clarify the observations from the results developing and using the realistic model is important for the study and optimisation of network on chip, and to show that using the bursty traffic can lead to the network performance suffering significantly from degradation. Detailed discussion on error analysis is represented in section below.

VI. Error Evaluation

Divergence is observed between the analytical and simulation results due to network configurations and numerous parameters that had to be tweaked. Thus, performance of the simulation result can be evaluated by computation of absolute errors (AEs) as well as the relative errors (REs), where AE is a measure of error magnitude for each simulation from its corresponding analytical value, and the relative error measures the percentage of the total error given by:

$$RE=\frac{\text{AE}}{\text{Analytical value}} \qquad (6.1)$$

Where $AE = |Simulation\ value - Analytical\ value|$ (6.2)

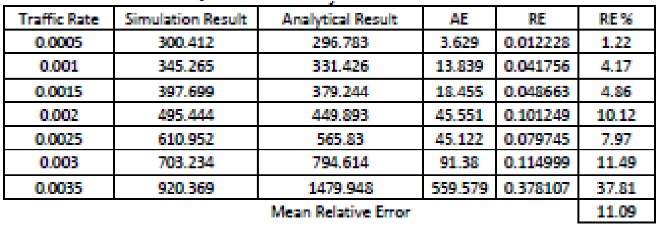
Furthermore the mean RE (MRE) can be computed for each graph using

$$MRE = \sum \frac{RE}{N} \qquad (6.3)$$

Where N is the number of samples.

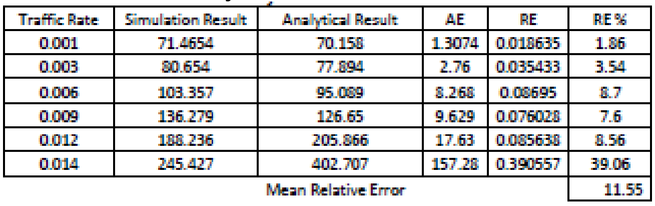
Having computed the RE, as well as MRE for each experiment. The worst case and the best case scenarios for each massage lengths 64,128 bits are represents on Tables 2,3,4,5 below:

Table 2: Summary of Errors for 64bit



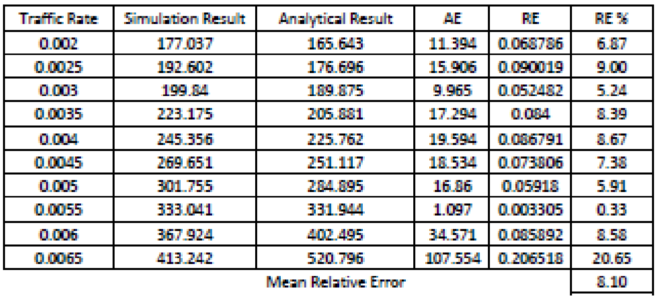
Traffic Rate	Simulation Result	Analytical Result	AE	RE	RE %
0.0005	300.412	296.783	3.629	0.012228	1.22
0.001	345.265	331.426	13.839	0.041756	4.17
0.0015	397.699	379.244	18.455	0.048663	4.86
0.002	495.444	449.893	45.551	0.101249	10.12
0.0025	610.952	565.83	45.122	0.079745	7.97
0.003	703.234	794.614	91.38	0.114999	11.49
0.0035	920.369	1479.948	559.579	0.378107	37.81
		Mean Relative Error			11.09

Table 3: Summary of Errors for worst case of 128bit



Traffic Rate	Simulation Result	Analytical Result	AE	RE	RE %
0.001	71.4654	70.158	1.3074	0.018635	1.86
0.003	80.654	77.894	2.76	0.035433	3.54
0.006	103.357	95.089	8.268	0.08695	8.7
0.009	136.279	126.65	9.629	0.076028	7.6
0.012	188.236	205.866	17.63	0.085638	8.56
0.014	245.427	402.707	157.28	0.390557	39.06
		Mean Relative Error			11.55

Table 4: Summary of Errors for best case of 128bit



Traffic Rate	Simulation Result	Analytical Result	AE	RE	RE %
0.002	177.037	165.643	11.394	0.068786	6.87
0.0025	192.602	176.696	15.906	0.090019	9.00
0.003	199.84	189.875	9.965	0.052482	5.24
0.0035	223.175	205.881	17.294	0.084	8.39
0.004	245.356	225.762	19.594	0.086791	8.67
0.0045	269.651	251.117	18.534	0.073806	7.38
0.005	301.755	284.895	16.86	0.05918	5.91
0.0055	333.041	331.944	1.097	0.003305	0.33
0.006	367.924	402.495	34.571	0.085892	8.58
0.0065	413.242	520.796	107.554	0.206518	20.65
		Mean Relative Error			8.10

Observing the results of Tables 2, 3, and 4, it is obvious that all the sample points which lie on the steady state region have RE less than 10%; which is an acceptable relative error. Additionally, considering the MREs of 11.09%, 11.55%, and 8.10% reveal that the effect of divergence due to points outside the steady state region (i.e REs 37.81%, 39.06%, and 20.65% for the three tables above respectively) do not undermine the performance of the system. Moreover, regions outside the steady state are not considered when evaluating network performance [86].

As presented above the errors are within the acceptable range, however these can be further improve by a careful selection of the network parameters with a view to achieving optimum results. It is worth noting that this will be an interesting topic for further research

VI. CONCLUSION

This analytical model was developed and implemented to evaluate the performance of network on-chip under bursty traffic. The bursty traffic is modeled by the well-known MMPP. The topology constructed in network on chip architecture is the popular fat-tree m-port n-tree. For this work extensive simulation experiments have been conducted to validate the accuracy of the model.

The accuracy and tractability of the model make it a practical and cost-effective tool to gain insight into the performance of network on chip in the presence of realistic

DOI: 10.18535/ijecs/v5i10.38

network traffic. The analytical results have shown that the network performance degrades considerably under such traffic patterns.

Future Work

The models of this work have been implemented in a way that it can be used improve the NoC performance through investigate the system under different aspects like routing mechanisms, topology scenarios, switch methods and traffic patterns. In addition considered secondary performance metrics *i.e*. energy efficiency.

Furth more, investigate the effect of different topologies with different sizes on network message latency may lead to more research with the aim of enhance the system efficiency mainly in terms of message latency and other metrics like power consumption.

In a later stage we plan to examine the integration of adaptive routing mechanism into NoC along with deterministic routing. We expect that the mechanism of choosing the shortest path adopted by adaptive routing will

play significant roll on the performance of the message latency Hotspot is defined as a hotspot node and all other nodes send a specific portion of their messages to this node.

In wormhole based NoCs, hotspot modules or nodes dramatically reduce network efficiency and unfairly allocate the system's resources. For example, nodes near the hotspot module receive larger portions of their capacity. A single hotspot module within a NoC topology can greatly decrease the performance of the entire system; therefore, some works have been done in order to solve this problem [38].

ACKNOWLEDGMENT

REFERENCES

1. Dally, W.J. and B. Towles. *Route packets, not wires: On-chipinterconnection networks*. in *Design Automation Conference, 2001.Proceedings*. 2001. IEEE.

2. Benini, L. and G. De Micheli, *Networks on chips: a new SoCparadigm.* Computer, 2002. **35**(1): p. 70-78.

3. Bononi, L. and N. Concer. *Simulation and analysis of network on chiparchitectures: ring, spidergon and 2D mesh*. in *Proceedings of theconference on Design, automation and test in Europe: Designers'forum*. 2006. European Design and Automation Association.

5. Duato, J., S. Yalamanchili, and L.M. Ni, *Interconnection networks: Anengineering approach*. 2003: Morgan Kaufmann.

6. Dally, W.J. and B. Towles, *Principles and practices of interconnectionnetworks*. 2004: Morgan Kaufmann.

7. Pande, P.P., et al., *Performance evaluation and design trade-offs fornetwork-on-chip interconnect architectures.* Computers, IEEETransactions on, 2005. **54**(8): p. 1025-1040.

8. Jerger, N.E. and L.-S. Peh, *On-chip networks.* Synthesis Lectures onComputer Architecture, 2009. **4**(1): p. 1-141.

9. Kathleen, F., *The Markov-modulated Poisson process (MMPP)cookbook.* Performance Evaluation, 1993. **18**(2): p. 149-171.

10. Ghandali, S. and S.M. Safavi. *Modeling multimedia traffic in IMSnetwork using MMPP*. in *Electronics Computer Technology (ICECT),2011 3rd International Conference on*. 2011. IEEE.

11. Kang, S.H. and D.K. Sung, *A CAC scheme based on real-time cellloss estimation for ATM multiplexers.* Communications, IEEETransactions on, 2000. **48**(2): p. 252-258.

12. Moadeli, M., et al. *Communication modelling of the Spidergon NoCwith virtual channels*. in *Parallel Processing, 2007. ICPP 2007.International Conference on*. 2007. IEEE.

13. Kiasari, A.E., Z. Lu, and A. Jantsch, *An analytical latency model fornetworks-on-chip.* Very Large Scale Integration (VLSI) Systems, IEEETransactions on, 2013. **21**(1): p. 113-123.

14. Varga, A. *The OMNeT++ discrete event simulation system*. In *Proceedings of the European Simulation Multiconference (ESM'2001)*.2001. sn.

15. Weingartner, E., H. Vom Lehn, and K. Wehrle. *A performance comparison of recent network simulators*. in *Communications, 2009. ICC'09. IEEE International Conference on*. 2009. IEEE.

16. Sahu, P.K. and S. Chattopadhyay, *A survey on application mapping strategies for network-on-chip design.* Journal of Systems Architecture, 2013. **59**(1): p. 60-76.

17. Singh, A.K., et al. *Mapping on multi/many-core systems: survey of current and emerging trends*. in *Proceedings of the 50th Annual Design Automation Conference*. 2013. ACM.

18. Yang, B., et al. *Tree-model based mapping for energy-efficient and low-latency network-on-chip*. in *Design and Diagnostics of Electronic Circuits and Systems (DDECS), 2010 IEEE 13th International Symposium on*. 2010. IEEE.

19. Hemani, A., et al. *Network on chip: An architecture for billion transistor era*. in *Proceeding of the IEEE NorChip Conference*. 2000.

20. Leiserson, C.E., et al. *The network architecture of the Connection Machine CM-5*. in *Proceedings of the fourth annual ACM symposium on Parallel algorithms and architectures*. 1992. ACM.

21. Boura, Y.M. and C.R. Das, *Performance analysis of buffering schemes in wormhole routers.* Computers, IEEE Transactions on, 1997. **46**(6): p. 687-694.

22. Po-chi, H. and L. Kleinrock. *A queueing model for wormhole routing with timeout*. in *Computer Communications and Networks, 1995. Proceedings., Fourth International Conference on*. 1995.

23. Greenberg, R.I. and L. Guan. *An improved analytical model for wormhole routed networks with application to butterfly fat-trees*. 1997. IEEE.

24. Javadi, B., J. Abawajy, and M. Akbari, *Performance Analysis of Interconnection Networks for Multi-cluster Systems*, in *Computational Science – ICCS 2005*, V. Sunderam, et al., Editors. 2005, Springer Berlin / Heidelberg. p. 205-212.

25. Gomez, C., et al. *Deterministic versus adaptive routing in fat-trees*.2007. IEEE.

26. Bouhraoua, A. and M. Elrabaa. *An efficient network-on-chip architecture based on the fat-tree (FT) topology*. in *Microelectronics, 2006. ICM'06. International Conference on*. 2006. IEEE.

27. Lee, J., et al. *Do we need wide flits in networks-on-chip?* in *VLSI (ISVLSI), 2013 IEEE Computer Society Annual Symposium on*. 2013.IEEE.