

PERFORMANCE ASSESSMENT OF DIFFERENT NETWORK-ON-CHIP TOPOLOGIES

Tetala Neel Kamal Reddy

Dept. of Electronics &
Communication Engineering,
NIT Rourkela,
Odisha, India

tneelkamalreddy@gmail.com

Ayas Kanta Swain

Dept. of Electronics &
Communication Engineering,
NIT Rourkela,
Odisha, India

swain.ayas@gmail.com

Jayant Kumar Singh

Dept. of Electronics &
Communication Engineering,
NIT Rourkela,
Odisha, India

jayantsngh101@gmail.com

Kamala Kanta Mahapatra

Dept. of Electronics &
Communication Engineering,
NIT Rourkela,
Odisha, India

kkm@nitrrkl.ac.in

Abstract— Multiprocessor System-on-Chip platforms are gaining prominence in the field of SoC design, which accommodates several large heterogeneous semiconductor intellectual property (IP) blocks, integrated onto a single chip. However, there's a crisis of global interconnection with existing bus architectures in such SoC Designs. In response to this crisis, Network-on-Chip (NoC) is an upcoming paradigm, and is becoming the leading contender to replace the conventional bus architectures. Many Network-on-Chip topologies have been proposed in an attempt to tackle various chip architecture needs and routing techniques. In this paper, some of the topologies such as Mesh, Torus, Binary Tree and Butterfly Fat Tree (BFT) have been simulated using a Network Simulator (NS2) and their performances have been assessed and compared taking throughput, maximum end-to-end latency and dropping probability as assessment parameters.

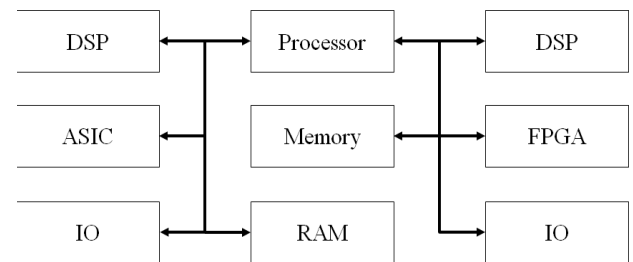
Keywords—SoC design, IP, Network-on-Chip, Topologies, NS2

I. INTRODUCTION

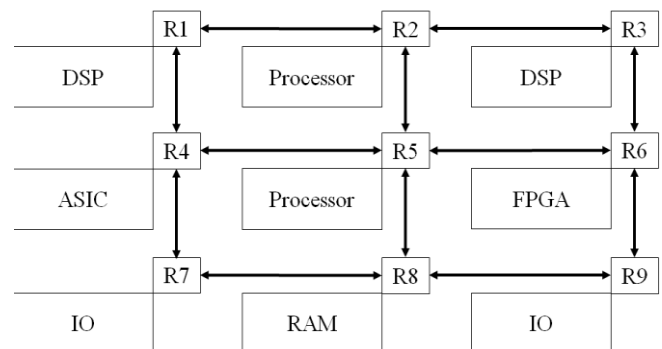
Within the next decade, it will be conceivable to assimilate hundreds of billions of transistors on a single chip, which will allow for the incorporation of hundreds or even thousands of processor cores (a multi-core architecture) on a solitary die along with the interconnect framework and memory [5]. This type of system will probably be more communication-centric. The fact that interconnects would require special consideration in upcoming multi-core systems has already been recognized several years ago when research began to emphasize on the network-on-chip (NoC) paradigm. In a multi-core architecture, interconnect structure inhabits a large amount of the on-chip area, i.e., a huge quantity of transistors that otherwise might have been used for increasing the number and the intricacy of the computational resources now are needed to be used for designing the communication infrastructure [8].

A. Shared Bus v/s Network-on-Chip

An example showing the difference between the interconnection mechanism in a SoC using shared-bus architecture and a 3x3 mesh based Network-on-Chip is shown in Fig. 1 [7]. An approach existing in traditional SoCs for inter-processor communication is having dedicated buses among communicating resources.



(a)



(b)

Fig. 1 Comparison between a shared-bus and a mesh architecture
(a) Traditional SoC using shared-bus (b) A 3x3 mesh NoC

But, this restricts flexibility. Another approach for this purpose using bus architecture is use of common buses, which although is cheaper and easier to implement than the NoC, they have deficiency in scalability and predictability and are not adept to keep up with the growing requirements of forthcoming SoCs in terms of performance, power dissipation, scalability, timing etc. [4].

The NoC architecture offers some advantages over shared-bus architecture. Firstly, communication is better scalable in a NoC than using bus architecture which mainly comprises of long interconnect wires. Secondly, it is conceivable to develop computational resources as individual IPs and generate the

NoC to link the IP blocks as resources in the NoC. Thirdly, Scalable and Configurable network is conceivable on a flexible platform that can be adjusted to the needs of diverse workloads. Moreover, networks are generally preferable to global wires or bus-based architecture because they have higher bandwidth and support multiple concurrent communications.

The design of a NoC-based system typically begins with a design space study phase whose objective is to find the best NoC instance by evaluating the performance of different candidates. Parameters to be assessed consist of network topology and links as well as routing and switching strategies that directly affect NoC's performance in terms of latency and throughput [6]. In this paper, we simulate different topologies of NoC keeping the routing and switching strategies constant and assessing the performance of each topology in terms of max end-to-end latency, throughput and dropping probability using a network simulator NS2 [12].

This paper has been divided into following sections. In section II, we describe the concept of network topology and different NoC topologies. In section III, the network simulator NS2 has been briefly described. In section IV, we give a brief idea regarding the constraints used in the simulation procedure. Section V describes the assessment parameters we used to analyze and compare the various topologies of NoC. We have included the obtained results of our simulations in the form of tables and graphs in section V.

II. TOPOLOGIES

Network topology refers to the organization of the shared router nodes and channels in an on-chip network. The topology of a NoC can be compared to a roadmap. The channels (similar to roads) transport packets (similar to vehicles) from one router node (crossing) to another [3]. A good topology utilizes the features of the existing packaging technology to achieve required application bandwidth and latency. Choosing a network topology is the principal step in designing a network as the routing strategy and flow-control methods are governed heavily by the topology. Deciding on a topology also helps in designing of the router to be used in the NoC, as clarified in [9]. The ways in which the different nodes in a network are connected and communicate with each other are controlled by the network topology. Some of the topologies for NoC are Mesh [10], Torus [1], Binary Tree and Butterfly Fat Tree (BFT) [11], which are discussed below.

A. Mesh

This architecture is the most common among all interconnection topologies where each router, apart from those at the edges, is linked to four adjoining routers and one computation resource (IP), by the way of communication channels. This topology allows incorporation of large number of IP cores in a regular-shape structure. Fig. 2(a) shows a 4x4 mesh NoC with 16 functional IP blocks.

B. Torus

The torus architecture as shown in Fig. 2(b) is fundamentally similar as a mesh except that routers at the

edges are linked to the routers at the opposite edge through folded channels. Every router has five ports, one linked to the computational resource and the others linked to the closest neighboring routers. The long fold-around connections may generate excessive delays.

C. Binary Tree

In the Binary Tree topology, the design is modeled in the form of a tree. Each node in the tree can be denoted by a set of coordinates (level, position) where level is the vertical level in the tree and position is the horizontal placing in left to right ordering. Here, as depicted in Fig. 2(c), each router node is linked to 2 nodes in the subsequent level with all the resource nodes present at the bottommost vertical level.

D. Butterfly Fat Tree:

In the Butterfly Fat Tree (BFT) topology, the design is modeled in the form of a tree with butterfly style links. Each node can be denoted similarly as in Binary Tree. The resource (IP) nodes are at the bottommost vertical level such that 4 resource nodes are linked to a router node, which is at a level higher than the resource nodes. Each router node is linked to either 4 router or resource nodes, as depicted in Fig. 2(d).

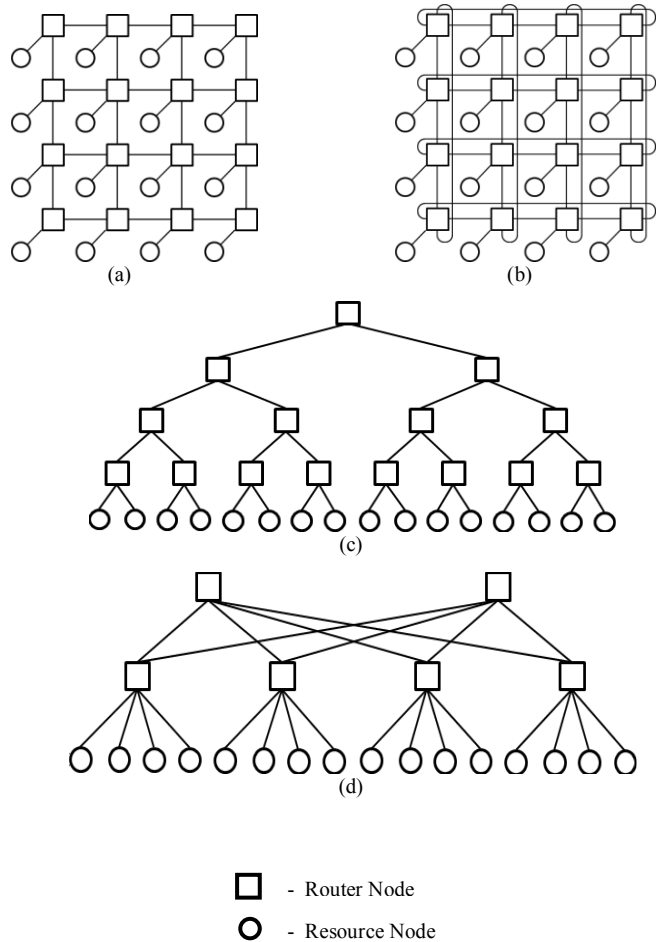


Fig. 2 Network-on-Chip Topologies
(a) 4x4 Mesh (b) 4x4 Torus (c) Binary Tree (d) Butterfly Fat Tree (BFT)

III. NETWORK SIMULATOR NS2

NS2 [12] is an open-source object-oriented discrete event network simulator developed at UC Berkeley and intended explicitly for exploration in computer communication networks. It is suitable for packet switched networking. NS2 has been developed in two languages. C++ is used for thorough executions of protocols like TCP or any tailored ones. TCL scripting, alternatively, is the front-end interpreter for NS-2 designed for creating commands and configuration interfaces [13]. NS2 contains a bundle of tools that helps in simulating the behavior of networks. It can be used to create different network topologies, which can be simulated under a traffic load to generate a log of events regarding the transfer of packets from one node to another.

NS2 provides two tools for processing data post simulation:

A. Trace File

NS2 generates a text-based packet tracing file that registers the features of packets passing through network check-points. Using AWK language [14], which is an interpreted programming language designed for text processing, these log events can be assessed to comprehend the network behavior.

B. NAM (Network Animator)

It also generates a NAM trace that registers simulation features in a text file, and then uses the text file to playback the simulation using animation.

IV. SIMULATION IN NS2

In this paper, we have simulated a 4x4 Mesh, 4x4 Torus, Binary tree, Butterfly Fat Tree (BFT) topologies using NS2. For this, each resource node, represented by a circle, has been connected to a router node, represented by a square and the router nodes are interconnected as per the topology as shown in figures 2(a)-2(d). Various constraints applied in NS2 to simulate NoCs are provided in Table I.

TABLE I
CONSTRAINTS APPLIED IN NS2 TO SIMULATE NOCs

NoC Model Parameters	Parameter Constraints applied in NS2
Number of Resource (IP) Nodes	16
Connections	Resource-Router, Router-Router
Transmission Protocols	User Datagram Protocol (UDP)
Routing Scheme	Static
Routing Protocol	Shortest Path
Queue Mechanism	Stochastic Fairness Queuing (SFQ)
Link Queue	8 packets
Bisection Bandwidth (Max.)	Router-to-router – 300Mb Resource-to-router – 200Mb
Traffic Generation	Constant Bit Rate (CBR)
Traffic Rate	180 Mb
Packet Size	16 bytes

V. PERFORMANCE ASSESSMENT PARAMETERS FOR NOC TOPOLOGIES

The assessment parameters for comparing the NoC topologies are defined as:

A. Max. End-to-End Latency

The time required to deliver a packet, i.e. the time when the first bit of the packet is sent by the source till the last bit of the packet being received by the destination, is called Latency. The Max End-to-End Latency is given by the maximum latency for a pair of source-destination nodes at the farthest distance in a network topology. The units are μ s, ns etc.

B. Dropping Probability

It is given by the ratio of the packets dropped when traversing in a topology to the total packets sent by the source nodes in that topology. A topology with Dropping Probability 0 value suggests that a packet will never be dropped, 100 would imply that all packets are dropped.

C. Throughput

It is defined as the rate at which traffic (or packets) is delivered to the destination nodes. The units of throughput are Mbps, Gbps etc.

Performance of each topology, namely 4x4 Mesh, 4x4 Torus, Binary Tree, Butterfly Fat Tree (BFT) is assessed on the basis of these parameters under different traffic load conditions, which is given by the number of source nodes active or transmitting packets out of the total available resource nodes in a topology.

VI. RESULTS AND ANALYSIS

Under the different traffic conditions, we obtained the following simulation results which have been depicted in charts to show the trade-off between the assessment parameters.

TABLE II
MAX. END-TO-END LATENCY V/S LOAD FOR DIFFERENT TOPOLOGIES

Load	Max. End-to-end Latency (μ s)			
	4X4 Mesh	4X4 Torus	Binary Tree	Butterfly Fat Tree
25%	803.844	802.133	811.562	409.855
50%	803.844	802.133	814.86	410.609
75%	803.844	802.133	831.667	412.29
100%	811.253	802.133	833.29	413.298

TABLE III
DROPPING PROBABILITY V/S LOAD FOR DIFFERENT TOPOLOGIES

Load	Dropping Probability			
	4X4 Mesh	4X4 Torus	Binary Tree	Butterfly Fat Tree
25%	0	0	0.078	0.078
50%	0.0605	0	0.161	0.141
75%	0.082	0.049	0.456	0.376
100%	0.156	0.054	0.537	0.483

TABLE IV
AVERAGE THROUGHPUT V/S LOAD FOR DIFFERENT TOPOLOGIES

Load	Average Throughput (Mbps)			
	4X4 Mesh	4X4 Torus	Binary Tree	Butterfly Fat Tree
25%	36.252	36.252	33.412	33.412
50%	66.09	70.345	59.039	60.455
75%	101.603	104.685	60.197	69.092
100%	116.876	131.726	64.168	71.565

TABLE V
NODE THROUGHPUT AT EACH NODE FOR DIFFERENT TOPOLOGIES
(100% TRAFFIC LOAD)

Nodes	Node Throughput (Mbps)			
	4X4 Mesh	4X4 Torus	Binary Tree	Butterfly Fat Tree
0	82.199	115.56	115.927	138.644
1	82.196	118.64	48	59.352
2	115.561	117.095	79.2898	59.987
3	117.095	117.095	48.972	60.362
4	138.644	138.644	48.007	58.737
5	82.197	138.644	77.797	61.253
6	117.095	138.644	50.204	62.524
7	116.398	138.644	49.339	59.741
8	115.958	115.958	60.619	63.466
9	138.644	138.644	75.993	63.153
10	138.644	138.644	76.146	64.474
11	115.561	138.644	131.161	58.739
12	116.626	138.644	59.366	59.741
13	138.644	138.644	48.111	58.737
14	138.644	138.644	115.927	138.644
15	115.912	136.826	59.823	77.489

Fig. 3 is a graph with Traffic Load on x-axis and Max End-to-End Latency on y-axis. Here, it has been found that BFT has the lowest max end-to-end latency due to the lesser number of links as compared to others. Also, the 4x4 torus has a lower max end-to-end latency than the 4x4 mesh due to the folded channels or links. The related data is available in Table II.

Fig. 4 is a graph which depicts the Traffic Load on x-axis v/s Dropping Probability on y-axis. We can observe here that 4x4 torus has the least dropping probability due to the presence of greater number of links between router nodes. The related data is available in Table III.

Fig. 5 is a graph between Traffic Load on x-axis and Average Throughput on y-axis. From this, we can infer that the 4x4 torus has the highest average throughput as there are lesser dropped packets in this topology. The related data is available in Table IV.

Fig. 6 is a histogram depicting Node Throughput for various topologies for the 16 resource nodes under 100% traffic load conditions. The related data is available in Table V.

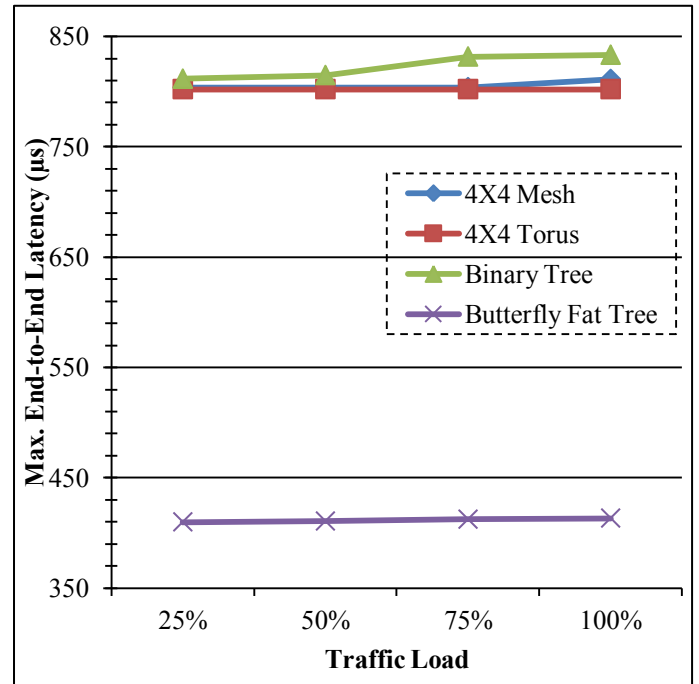


Fig. 3 Variation of Max. End-to-End Latency with Traffic Load for different topologies

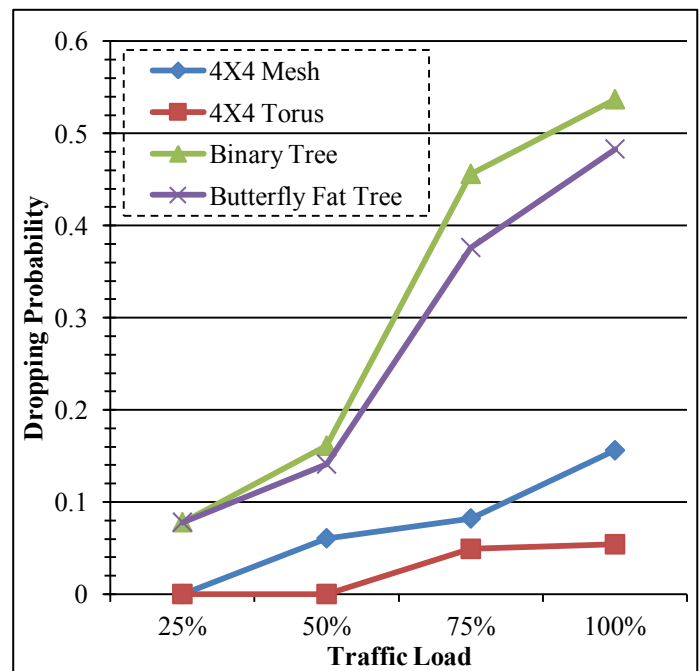


Fig. 4 Variation of Dropping Probability with Traffic Load for different topologies

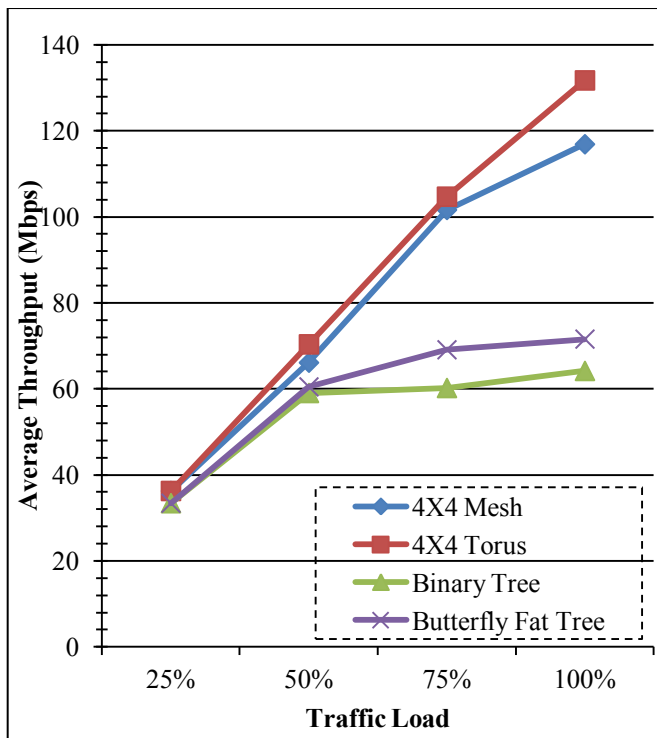


Fig. 5 Variation of Average Throughput with Traffic Load for different topologies

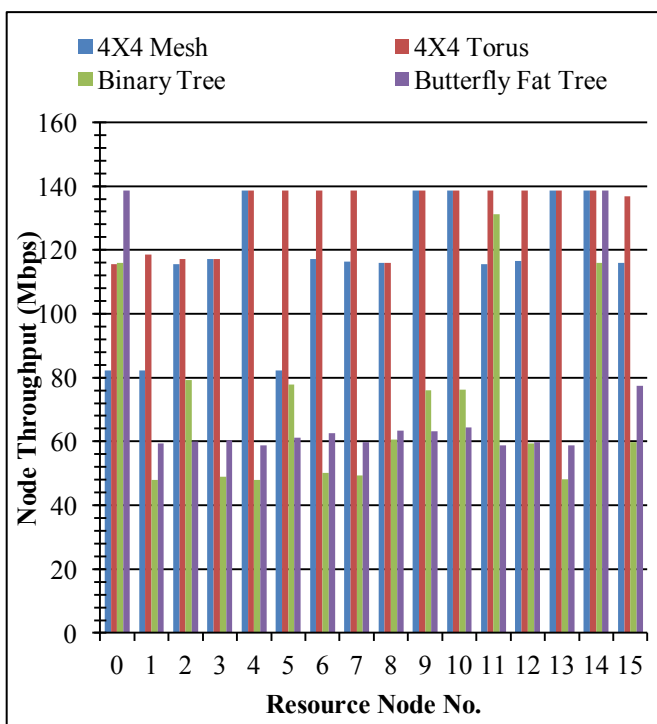


Fig. 6 Throughput for each node under 100% Traffic Load for different topologies

VII. CONCLUSION

In this paper, we assessed the performance of various Network-on-Chip topologies keeping other properties like routing strategies, traffic conditions etc. constant using the network simulator NS2. We deduce that as NoC theoretically resembles a conventional computer network in terms that both have resources, routers, flow-control mechanisms, hence NS2 can be utilized to assess the performance of NoCs. From the obtained results, we could infer that as per max end-to-end latency, BFT gives a better performance as compared to other topologies considered in the paper. Considering throughput and dropping probability, 4x4 Torus proves to be a superior topology as compared to 4x4 Mesh, Binary Tree and BFT.

REFERENCES

- [1] William J. Dally and Brian Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks", in *Proc. DAC*, 2001, pp. 683-689.
- [2] L. Benini and G. De Micheli, "Networks on chips: A new SoC paradigm", *IEEE Computer Magazine*, vol.35, no.1, pp. 70-78, January 2002.
- [3] W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks*, Elsevier Inc., 2004.
- [4] Wen-Chung Tsai, Ying-Cherng Lan, Yu-Hen Hu, and Sao-Jie Chen, "Networks on Chips: Structure and Design Methodologies", in *Journal of Electrical and Computer Engineering*, Vol.2012, Article ID 509465.
- [5] S. Borkar, "Design perspectives on 22 nm CMOS and beyond," in *Proc. DAC*, 2009, pp. 93-94.
- [6] U. Ogras, J. Hu, and R. Marculescu, "Key Research Problems in NoC Design: A Holistic Perspective," in *Proc. IEEE/ACM/IFIP Int'l Conf. Hardware/Software Codesign and System Synthesis*, pp. 69-74, 2005.
- [7] Siti Aisah Binti Mat Junos, "Network-On-Chip Mesh Topology Modeling and Performance Analysis", M. E. thesis, Universiti Teknologi Malaysia, May 2009.
- [8] Mohammad Abdullah Al Faruque, Thomas Ebi, and Jörg Henkel, "AdNoC: Runtime Adaptive Network-on-Chip Architecture," in *IEEE Transactions on VLSI Systems*, Vol. 20, No. 2, February 2012, pp. 257-269.
- [9] S. Swapna, A. K. Swain and K. K. Mahapatra, "Design and Analysis of five port router for Network on Chip", *Asia-Pacific Conference on Postgraduate Research in Microelectronics & Electronics*, December 2012.
- [10] S. Kumar, A. Jantsch, et al, "A network on chip architecture and design methodology", *Proceedings of IEEE computer society annual symposium on VLSI*, 2002.
- [11] P.P. Pande, C. Grecu, A. Ivanov, and R. Saleh, "Design of a Switch for Network on Chip Applications," *Proc. Int'l Symp. Circuits and Systems (ISCAS)*, vol. 5, pp. 217-220, May 2003.
- [12] NS2 website [Online]. Available: <http://www.isi.edu/nsnam/ns/>
- [13] Teerawat Issariyakul and Ekram Hossain, *Introduction to Network Simulator NS2*, Springer, 2009.
- [14] Stutz, Michael, *Get started with GAWK: AWK language fundamentals*, September 2006.
- [15] Yi-Ran Sun, "Simulation and Performance Evaluation for Networks on Chip", M. S. thesis, KTH Royal Institute of Technology, December 2001.