

# Performance Analysis of Interconnection Networks for Packet Delay using Source Routing

Lalit Kishore Arora  
Ajay Kumar Garg Engg College,  
Ghaziabad, U. P., India

Raj Kumar  
Gurukul Kangri University,  
Haridwar, U. K., India

Sapna Yadav  
SGIT, Ghaziabad, U. P., India

## ABSTRACT

The packet loss and packet delay are the measure performance parameter for evaluating the network topologies in Interconnection Network design. This paper, evaluates the performance of packet loss on two different interconnection networks, Mesh and Torus, with source routing. But here the main emphasis is on the packet transmission delay on both networks. The simulation framework designed and simulated for these networks using NS2 and evaluates the latency in both networks and results shown in the paper. The results analyzed, which are useful for designing the interconnection networks.

**Keywords:** Mesh Topology, Torus topology, interconnection networks, source routing, latency.

## 1. INTRODUCTION

Meshes and torus based interconnection networks have been utilized extensively in the design of parallel computers in recent years. Computer architects have always strived to increase the performance of their computer architectures. High performance may come from fast dense circuitry, packaging technology, and parallelism. As the density of processor package increases; the length of the link connecting a certain number of processors decreases [1].

Although numerous studies have examined NoC implementation and performance, few have examined packet loss. Flow control in interconnection networks has mainly been an issue to prevent buffer overflow and packet loss. Packet loss occurs when one or more packets of data traveling across a network fail to reach their destination. Packet loss can be caused by a number of factors including buffer overflow, congestion, corrupted packets rejected in-transit, faulty link, faulty nodes or deadlocks. In addition to this, packet loss probability is also affected by down of links and distances between the transmitter and receiver [19].

In [18] paper, we have analyzed 2D Mesh performance on the one down link for one second, and changed two parameters packet size and packet generation interval and found that the ratio of packet loss is constant in both cases where traffic generator with acknowledgement is not be considered. But on the other hand when acknowledgement is considered in both cases, no packet loss has been found. Therefore the network with the traffic agent which uses acknowledgement mechanism is more reliable, and more secure. But the delay of transmission due to the link down will be occur.

Latency or packet delay is defined as the time taken by a packet to travel from the source resource to the destination resource [19].

In this paper, a separate frameworks for mesh and torus interconnection networks has been designed, where the analysis of the packet loss during the link down for 1.0 second

and packet delay during the transmission has been evaluated. Both interconnection networks, Mesh and Torus, are evaluated on with and without acknowledgement mechanism using simulation on NS2.

The remainder of this paper is organized as follows. In section II, related work has been discussed. Section III, describes the system model of mesh and torus interconnection network, which implemented and designed in NS2 and section IV, evaluates the performance and shows the results of simulation. Finally, drawn some conclusion in section V.

## 2. MOTIVATION

The numerous studies have done related to simulation on Mesh topologies. Recently, the NoC has been introduced as a new research area that emphasis on modeling and analyzing the on-chip interconnect. Sophisticated networks that have specialized switches and routers and defined topologies are the main NoC points for analysis and optimization [17]. Recently, NoC architectures have been surveyed and compared for different performance metrics. In the paper [11], a simulation-based approach using the NS-2 simulator was used to analyze a NoC mesh interconnect topology. It is based on the Chiplevel Integration of communicating Heterogeneous Element [12].

NS-2 is used to construct the topology and generate different traffic scenarios using an exponential traffic generator [19]. Packets are sent at a fixed rate during ON periods, and no packets are sent during OFF periods. Using this traffic generator, common network performance metrics such as drop probability, packet delay, throughput and communication load are analyzed against different buffer sizes and traffic injection rates [19].

Another paper [13], about the Mesh NoC has been presented, it is similar to [11] but with different results. Metrics such as latency and packet loss rate were presented as a function of the communication load and the buffer size, using the NS-2 simulator. In [14] authors compared the Ring, Irregular Mesh and Spidergon topologies using a discrete-driven simulator (OMNET++) based on the wormhole switching technique. Their analysis has shown that the Spidergon NoC outperforms others, including average latency and throughput. The type of traffic has not been mentioned despite of its prime importance in NoC. In [15], an Application Specific NoC (ASNoC) design methodology was proposed, that is, using a customized topology to fit the requirements of specific applications. In that work, the OPNET simulator is used to compare the proposed structure with a Mesh topology, using a HDTV decoder SoC as application example. An analytical model using queuing theory is introduced in [16] to evaluate the traffic behavior of the Spidergon NoC. Simulations to verify the model for message latency under different traffic rates and variable message lengths are presented in that work.

In [19], packet delay model is presented, where several different parameters are considered for packet delay, and so many factors are deriving the packet delay.

### 3. SYSTEM MODEL

The simulation model of interconnection architecture consists an  $(m \times n)$  mesh and an  $(m \times n)$  torus network of switches. Switches consists a slot for a resource. A resource may be a processor core, a memory block, an FPGA, a custom hardware block or any other peripheral devices, which fits into the available slot and compiles with the interface with the network. Assume that switches in network have buffers to manage data traffic. Figure 1, shows the architecture of both models with 16 nodes.

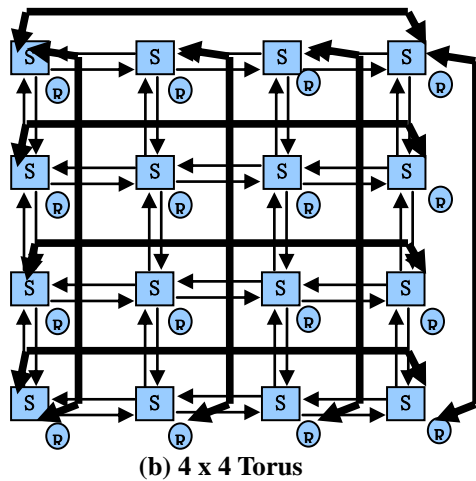
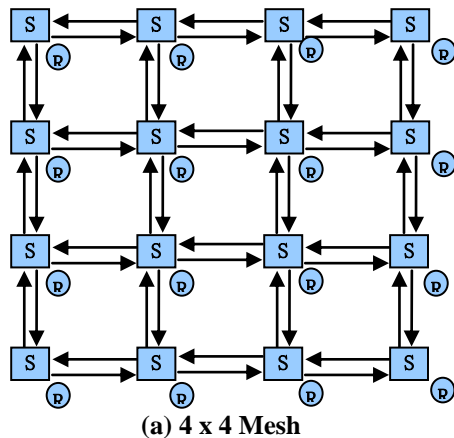


Fig. 1. Interconnection architecture

#### A. Topology

A 4 x 4 two-dimensional mesh topology and 4 x 4 two-dimensional torus topology was modelled and simulated. These topologies are easily scaled to different sizes. Different resources have their unique communication addresses, so here assumed that all switches has attached processor core as resources therefore treated similarly except that a traffic generator can be attached to resources. Switch, resource and link are three basic elements in the topology. Assume that the each resources has infinite buffer size but finite in switches. It means that the packet being dropped or lost cannot occur in resources but only take place in switches.

#### B. Communication Links

An inter-communication path between the switches is composed of links. Each node is connected with point-to-point bidirectional links. The bandwidth and latency of the link is configurable. When any link between two nodes is used during the packet travel, these nodes provides full resources to packet for transmission.

#### C. Routing

An inter-communication path between the switches is composed of links. Each node is connected with point-to-point bidirectional links. The bandwidth and latency of the link is configurable. When any link down between two nodes it implies that the packet cannot be travel between these nodes in any direction. This assumption was used in [10] and is realistic, because bidirectional links are actually implemented by using a single wire.

### 4. PERFORMANCE EVALUATION

In this section, to evaluate the performance of the mesh 7 torus interconnection networks we develop a simulation model in NS-2 with only built-in options. Tcl is used for specifying the Mesh interconnection network simulation model and running the simulation. We have used existing routing algorithm to compute the path and for packet generation.

Our implementation of both interconnection networks uses the source routing to send packets from source node to destination node. In source routing the information about the whole path from the source to the destination is pre-computed and provided in packet header [3] [19].

#### A. Simulation Environment

For the evaluation, a separated detailed event-driven simulators has been developed for both mesh and torus networks. These simulators model 16-node (4x4) network architecture where routing decision will be takes at source node using source routing methodology. Each node is connected with point-to-point bidirectional serial links. The bandwidth of link is set to 1 Mb and latency/delay is set to the 10 ms. All packets are generated using constant bit rate mechanism. All these topology parameters can be describe as a script file in Tcl, as shown below:

```
#Default Values for topology
set n 16; # Total number of nodes

set max_bw 1Mb; #maximum link band width
set linkDelay 10ms; # delay on each link

#configuration for links between the nodes in topology
$ns duplex-link $node(i) $node(i+1) $max_bw $linkDelay DropTail
```

In both environment models fixed the source, destination node and link for down for a particular period, 1 second. The time window of simulation is not fixed but time of link down and link up is 1.0 and 2.0 second respectively after the starting the simulation. We assume that when link is down, this link can not be used in any of its directions. This assumption was used in [10] and is realistic, because bidirectional links are actually implemented by using a single wire.

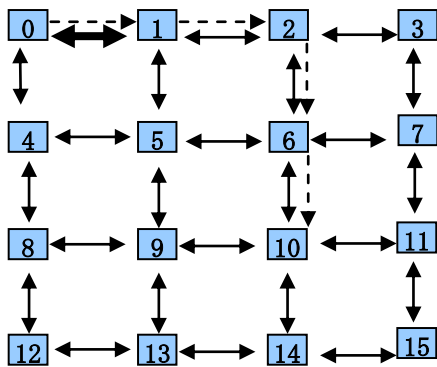
In these environment, model uses the packet generation of traffic is uniform. Simulation models fix the packet size and interval of packet generation. The following Tcl code shows

the traffic configuration setting for constant bit rate packet generation:

```
#Traffic Configuration: Constant bit rate traffic source
set cbr0 [new Application/Traffic/CBR]
$ cbr0 set packetSize_ 500
$ cbr0 set interval_ 0.0625
```

*Scenario-1 Transmission in Mesh without acknowledgement*

In this experiment Mesh interconnection network has been designed and packet transmission mechanism without acknowledgement is used.



**Fig. 2. Transmission in mesh**

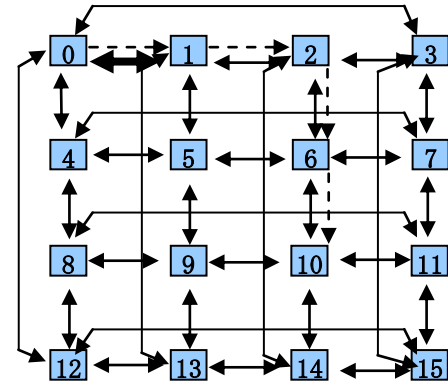
In the simulation model the source as node(0) and destination node as node(10), and link (0) to (1) for down for a particular period, 1 second are fixed. See figure-2, where dotted arrows show the path for each pair, and darkened bi-directional arrow shows the link which down for 1 second. The simulation time is fixed for 10-11 seconds, where packet generation window is from 0.5 to 4.5 second and time of link down and link up is 1.0 and 2.0 second respectively after the starting the simulation. Assume that when link is down, this link can not be used in any of its directions.

*Scenario-2 Transmission in Mesh using acknowledgement mechanism*

In this experiment model uses a packet transmission using acknowledgement mechanism on Mesh interconnection network. Source will receive acknowledgement for every packet sent to the destination. Like above scenario-1, model fixes the source as node(0) and destination node as node(10), and link (0) to (1) for down for a particular period, 1 second when first acknowledgement is going to received by source, figure-2.

*Scenario-3 Transmission in Torus without acknowledgement*

In this experiment Torus interconnection network is designed which uses a packet transmission mechanism without acknowledgement. We fixed the source as node(0) and destination node as node(10), and link (0) to (1) for down for a particular period, 1 second. See figure-3, where dotted arrows show the path for each pair, and darkened bi-directional arrow shows the link which down for 1 second. The simulation time is fixed for 10-11 seconds, where packet generation window is from 0.5 to 4.5 second and time of link down and link up is 1.0 and 2.0 second respectively after the starting the simulation. Assume that when link is down, this link can not be used in any of its directions.



**Fig. 3. Transmission in torus**

*Scenario-4 Transmission in Torus using acknowledgement mechanism*

In this experiment Torus network model uses a packet transmission using acknowledgement mechanism on Mesh interconnection network. Source will receive acknowledgement for every packet sent to the destination. Like above scenario-3, model fixes the source as node(0) and destination node as node(10), and link (0) to (1) for down for a particular period, 1 second when first acknowledgement is going to received by source, figure-3.

**B. Simulation Results**

1) *Scenario 1:* In first experiment packet has been sent from source node (0) to destination node(10) on Mesh interconnection. Here is no facility available of acknowledgement. Total 801 packets are generated and received by the source and destination nodes respectively.

On the next step, we down the link from node(0) to node (1) of the path from source to destination for 1 second, we found source generated 801 packets but network has been lost 03 packets which are not received by the destination node. The 4.56 seconds taken by the scenario for transmission of the packets.

2) *Scenario-2:* Like above scenario-1, packet has been sent from source node (0) to destination node(10) on Mesh interconnection. But here we have used acknowledgement mechanism for packet transmission. Total 801 packets are generated and received by the source and destination nodes respectively in 7.58032 seconds.

On the next step, we down the link from node(0) to node (1) of the path from source to destination for 1 second, we found source generated 801 packets and no loss has been found but time taken for transmission was 10.24168 seconds. Which is delayed about 2.66136 seconds. We have down the link for only 1 second but delay taken 1.66136 second extra to complete the transmission.

3) *Scenario-3:* In this experiment packet has been sent from source node(0) to destination node(10) on Torus interconnection. Here is no facility available of acknowledgement. Total 801 packets are generated and received by the source and destination nodes respectively.

On the next step, we down the link from node(0) to node (1) of the path from source to destination for 1 second, we found source generated 600 packets but network has been lost 02 packets which are not received by the destination node. The

4.56 seconds taken by the scenario for transmission of the packets.

4) *Scenario-4:* Like above scenario-3, packet has been sent from source node (0) to destination node(10) on Torus interconnection. But here we have used acknowledgement mechanism for packet transmission. Total 801 packets are generated and received by the source and destination nodes respectively in 7.58032 seconds.

On the next step, we down the link from node(0) to node (1) of the path from source to destination for 1 second, we found source generated 801 packets and no loss has been found but time taken for transmission was 10.22504 seconds. Which is delayed about 2.64472 seconds. We have down the link for only 1 second but delay taken 1.64472 second extra to complete the transmission.

**TABLE 1**

**COMPARISON WITHOUT ACKNOWLEDGEMENT**

Network	Packets Generated	Packet Lost
Mesh	801	Nil
Mesh with down link	801	3
Torus	801	Nil
Torus with down link	600	2

**TABLE 2**

**COMPARISON WITH ACKNOWLEDGEMENT**

Network	Packets Generated	Time taken (seconds)
Mesh	801	7.58032
Mesh with down link	801	10.24168
Torus	801	7.58032
Torus with down link	801	10.22504

In the case of without acknowledgement mechanism on mesh network, source generated same number of packets (i.e. 801 packets) and lost only three packets at link down for one second. While, on torus network, source generated 801 packets in normal but 600 packet generated when link has been down for one second and also 2 packets are lost.

In the case of acknowledgement mechanism on mesh network, source generated same number of packets (i.e. 801 packets) and no lost has been found at link down for one second. But when link has not been down the transmission process completed within 7.58032 seconds while the transmission completed within 10.24168 seconds during the link down.

On other hand, in torus network, source generated same number of packets (i.e.801 packets) and no loss has been occurred when link has been down for one second. One more thing, when link has not been down the transmission process

completed within 7.58032 seconds while the transmission completed within 10.22504 seconds during the link down.

Mesh is slower than torus network in the acknowledgement mechanism while both network uses same path and distance from source to destination. Mesh delayed the transmission approximate 0.01664 seconds.

## 5. CONCLUSION

The 2D mesh and 2D torus interconnection networks has been evaluated, where they implemented both with and without-acknowledgement mechanism and packets has been sent using uniform traffic pattern.

Results show that, the torus has a good performance and fast when acknowledgement mechanism is used and mesh perform better when without acknowledgement mechanism is used. Therefore when we are considering transmission delay then torus is better than mesh, and when we are considering packet loss the mesh is better than torus in both mechanism.

In future we are going to develop a new topology which will perform better in both performance metrics.

## 6. REFERENCES

- [1] Zhu Ding, "Adaptive Hybrid Switching Technique for Parallel Computing System", 2000
- [2] <http://pages.cs.wisc.edu/~tvrdik/5/html/Section5.html>
- [3] Dally W.J. and Towles B., "Principles and Practices of Interconnection Networks", Morgan Kaufmann Publishers an Imprint of Elsevier Inc. ISBN: 0-12-200751-4. 2004.
- [4] Axel Jantsch and Hannu Tenhunen, "Networks on Chip", Kluwer Academic Publishers. ISBN: 1-4020-7392-5. 2003.
- [5] J. Flich, P. Lopez, M. P. Malumbres and J. Duato. "Improving the Performance of Regular Networks with Source Routin", Proceeding of the IEEE International Conference on Parallel Processing. 21-24 Aug. 2000. Page(s) 353-361.
- [6] Palesi M., Holsmark R., Kumar S. and Catania V., "Application Specific Routing Algorithms for Networks on Chip", IEEE Transactions on Parallel and Distributed Systems, Vol. 20, No. 3, March 2009.
- [7] Ge-Ming Chiu. "The Odd-Even Turn Model for Adaptive Routing" IEEE Transactions on Parallel and Distributed Systems, Vol. 11, No. 7, July 2000.
- [8] A. Patooghy, H. Sarbazi-Azad, "Performance comparison of partially adaptive routing algorithms. In 20th International Conference on Advanced Information Networking and Applications", 2006. AINA 2006. Volume 2, 18-20 April 2006.
- [9] Palesi M., Holsmark R., Kumar S. and Catania V., "A Methodology for Design of Application Specific Deadlock-free Routing Algorithms for NoC Systems", In International Conference on Hardware-Software Codesign and System Synthesis, Seoul, Korea, October 22-25, 2006.
- [10] M.E. Gomez, P. Lopez, J.Duato, "A Memory-Effective routing Strategy for regular Interconnection Networks", in Proc. Int. Parallel and Distributed Processing Symp., 2005.

- [11] Y-R. Sun, S. Kumar, and A. Jantsch, "Simulation and evaluation of a network-on-chip architecture using NS-2," Proceedings of the IEEE NorChip Conference, 2002.
- [12] S. Kumar et al., "A network-on-chip architecture and design methodology," Proceedings of the International Symposium on VLSI (ISVLSI), 2002, pp. 117-124.
- [13] A. Hegedus, G. M. Maggio, and L. Kocarev, "A NS-2 simulator utilizing chaotic maps for network-on-chip traffic analysis," Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), vol. 4, 2005, pp. 3375-3378.
- [14] L. Bononi and N. Concer, "Simulation and analysis of network-on-chip architectures: Ring, Spidergon, and 2D Mesh," Proceedings of Design, Automation and Test in Europe (DATE) Conference and Exhibition (Designers' Forum), 2006, pp. 154-159.
- [15] J. Xu, W. Wolf, J. Henkel, and S. Chakradhar, "A design methodology for application-specific networks-on-chip," ACM Transactions on Embedded Computing Systems, vol. 5, May 2006, pp. 263-280.
- [16] M. Moadeli, A. Shahrabi, W. Vanderbauwhede, and M. Ould-Khaoua, "An analytical performance model for the Spidergon NoC," Proceedings of the 21st International Conference on Advanced Information Networking and Applications, 2007, pp. 1014-1021.
- [17] S. Suboh, M. Bakhouya, S. Lopez-Buedo, and T. El-Ghazawi, "Simulation-based approach for evaluating on-chip interconnect architectures", SPL Proc., pages 75–80, 2008
- [18] Lalit Kishore Arora, Rajkumar, "Simulation and Analysis of Packet loss in Mesh Interconnection Networks", IJCA Proceedings on Development of Reliable Information Systems, Techniques and Related Issues (DRISTI 2012), DRISTI(1):35 - 38, April 2012. Published by Foundation of Computer Science, New York, USA.
- [19] Saad Mubeen, "Evaluation of source routing for mesh topology network on chip platforms", Master of Science Thesis, 2009.