An Improved Congestion Free Modified Fat Tree Network – On – Chip Topology

Abhijit Biswas
Department of CSE
TSSOT, Assam University
Silchar, Assam
Email:abhi.021983@gmail.com

Md. Anwar Hussain
Department of ECE
N.E.R.I.S.T, Nirjuli
Arunachal Pradesh
Email: bubuli_99@yahoo.com

Alakesh Kalita
Department of CSE
TSSOT, Assam University
Silchar, Assam
Email: sunucse255@gmail.com

Abstract— **as the current decade is witnessing a shift from traditional System – On – Chip (SoC) to Network – On – Chip, but the inherent problems such as congestion, latency and delay still remains the major issues of concern. The effect of these issues in a chip may somehow be minimized by carefully designing a topology and a suitable routing algorithm which offers both path diversity and scalability. In this paper, an attempt is being made to minimize, the congestion in the internal routers of the MIN Fat Tree, by adding an extra link in each router for the purpose of interconnection amongst the router. The simulation of the resultant topology shows promising results in terms of average delay and average hop count when compared to MIN Fat Tree,**

Keywords— ***Network-on-chip, Fat Tree , Toplology, perfomance***

I. INTRODUCTION

The gradual shift from System – On – Chip (SoC) to Network – On – Chip (NoC) was aimed to minimize the problems of early SoCs such as, non – scalability, non-parallelism (only one communication in progress), arbitrary long delays, wirability so on and so forth [2].

A number of On – Chip topologies have been proposed over the years, such as, Hypercube, Torus, Mesh, butterfly etc. these topologies, on the other hand, are also not free from bottlenecks, for example, Hypercube suffers from wirability and VLSI packaging implementations due to non-uniform node degree[2][4]. The Mesh topology suffers from tight edge bandwidth and large network diameter. The Torus topology was proposed to reduce network diameter, which was inevitable in Mesh topology, by connecting the edge nodes in the same row and column, but this again resulted in long wraparound edges. This limitation can again be alleviated by folding the torus but which results in doubling the wire length. The butterfly topology on the other hand, has the advantage of having constant hop count for the given size of network but at the cost of long wires and no path diversity[2][5].

A generalized FAT tree proposed by Leiserson in [6], proves to be very hardware efficient for a given amount of hardware used. The FAT – Tree described by Leiserson takes the form of a binary tree, where the leaves represent processing elements and the non – leaves works as routing nodes. The fact that, the links become fatter as we move up in the network, makes the FAT – Tree resemble more like a real life tree. The routing in the said FAT – Tree is same as binary tree traversal.

The general FAT – Tree, as proposed by Leiserson in [6], imposes a very unrealistic bandwidth requirement for the root node to make the communication efficient. To mitigate this impracticality, a different class of FAT – Tree structures called *k – ary – N – trees*[3] were proposed, which goes on to be known popularly as MIN Fat – Trees.

In [1], the authors have implemented a partial group based routing in MIN FAT – Tree structure. The MIN FAT – Tree topology used in [1] is same as is proposed by A. Bouhraoua and M. E. Elrabaa in [7]. The figure below shows the structure of the MIN FAT – Tree.

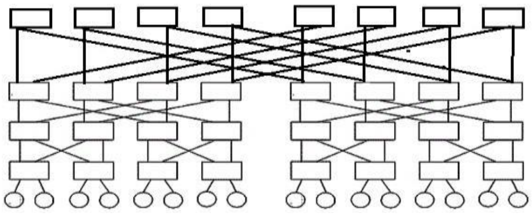

Figure 1: MIN Fat – Tree [1][7].

As shown in the Figure 1, the MIN FAT – Tree is not having a single root, hence the impractical bandwidth imposition on the single root FAT – Tree as proposed by Leiserson in [6] is mitigated.

This paper attempts to reduce the hop count and delay incurred in the MIN FAT – Tree, by adding an extra link in each router to connect to the higher level which is one level above the immediate upper level of the router, thereby increasing the path diversity and reducing the hop count by one hop. Also a suitable adaptive routing strategy is proposed for the network.

The rest of the paper is organized in V sections. Section II describes the proposed Improved Congestion Free Modified Fat Tree topology in detail, section III presents the proposed

978-1-5090-4620-1/16/$31.00 ©2016 IEEE

routing algorithm, section IV presents the simulation results and finally at section V a conclusion is drawn for the work presented here.

II. PROPOSRD TOPOLOGY

The proposed improved modified congestion free FAT – Tree topology is an attempt to reduce the delay, congestion and hop count of the FAT – Tree network of the given size. This objective is achieved by adding a bidirectional link to each router. These added links are connected to the higher levels which are one level higher to the immediate upper level as shown in the Figure 2.

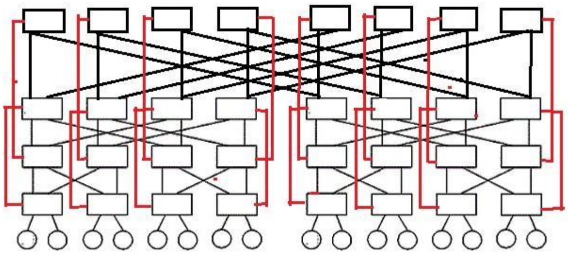

Figure 2: Improved congestion free Modified frees FAT – Tree Topology.

The proposed topology has similarity with the architecture as proposed by A. Bouhraoua and M. E. Elrabaa in [9]. The authors in [9] have proposed to double the downward link in each level of router. That is, in each level, the number of downward links will double the number than its previous level, this is reproduced as it is below from [9]

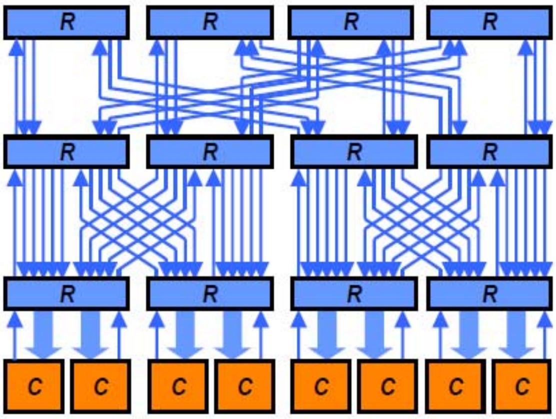

Figure 3: Modified FAT – Tree topology [9].

The Modified Fat – Tree removes contention but is low in resource utilization. The link utilization in the topmost level is 100%, as there are only small number of downward links are present, but as we move down, the number of downwards links increases and the link utilization decreases. At the lowest level, where there are large numbers of downward links present, the link utilization falls below 20% for a large network.

Our proposed network resembles the topology in [9], in such that, both the topologies add extra links, but unlike the topology in [9], the proposed topology here adds only one extra link to minimize the internal congestion, also the link is added with an aim to reduce the routing distance.

For the purpose of routing, again logical groups have been made, which are shown below in Figure 4.

The links serve the dual purpose of reducing the hop count and message delays offered by the router.

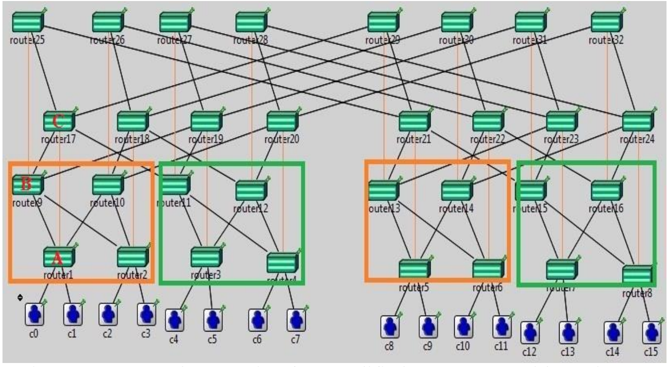

Figure 4: Improved congestion free Modified FAT – Tree with routing groups.

At any given moment, if the clients in the logical groups (Marked by colored square boxes) want to have a inter group communication; they can easily bypass the immediate upper level of routers by sending the packets to the level which is one above the immediate upper level. The routing is adaptive in the sense that whenever, the extra link added is congested or is busy, the routing can be done in a traditional MIN FAT – Tree routing strategy.

A. Router Design

The Internal structure of the router is designed such that, it has 6(six) internal ports which are well connected to each other by internal gates. The typical port structure is depicted below in the Figure 5.

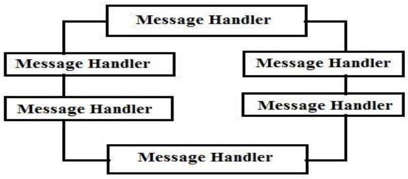

Figure 5: Internal View of the router with six ports.

The message handler is to deal with a message received by the router in its ports. When a new message is received by the message handler, first it will check whether the message received is from one of the internal gates within the router or from an external gate. If it is from an internal gate then it signifies that the routing function has been done and it will be sent directly out of the port to another router based on the routing decision. On the other hand if the message is received from an external gate, meaning from another router, then it will calculate the next path for the message and send to a specific port within the router for sending it to the next router in the path. Note that in every router the message handlers shall take the routing decision for each message received from external gates.

To identify a router we have assigned a level id and a group id to each router. The router connected with the client is considered as a level 1 router and they have no group id. The immediate upper level has a level id 2 and they form a group including two routers. The next level is considered as level 3 router and they form a group including four routers. And in this way, in level N, a group will be form including 2^{N-1} routers. The Figure 6 depicts the above explained scenario.

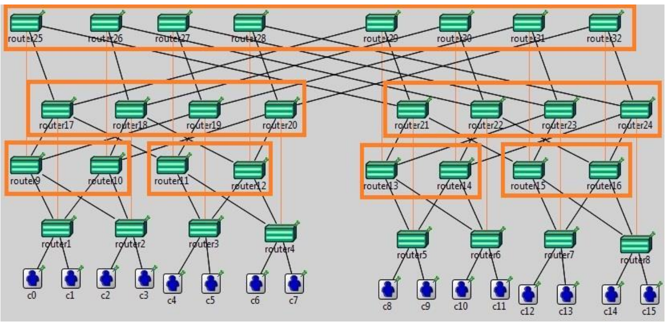
Figure 6: The routers with level depiction.

B. *Packet Flow communication*

The packet flow communication describes how a packet flows within a router. A new packet is generated in client's source module. After generating the packet, the packet is forwarded to the connected client router and the Message Handler module of connected client router accepts the packet. Then calculates the next route, the packet is forwarded to the next router and every router do the same process as the client connected router, until the packet reaches its destination router. After that the destination router sends the packet to the client and in client, the sink module accepts the packet.

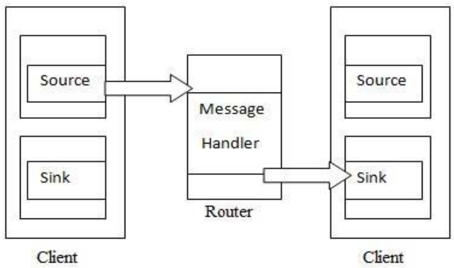

Figure 7: Packet flow communication.

C. *The client design*

Clients are IP blocks that are attached to the routers via bidirectional links. For the identity of a client we assigned a number to each and every client from 0 to n. The value of n depends upon the size of the network. Each client has two modules, source and sink [1][8], for managing incoming and outgoing packets. The source module is responsible for generating and sending a new packet from the client. The sink module is responsible for receiving an incoming packet at the client.

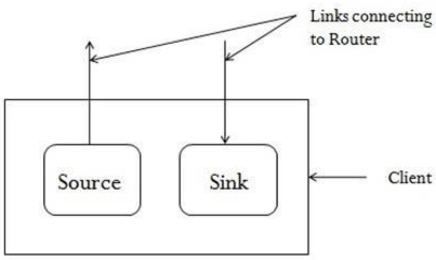

Figure 8: Client Structure [1][8].

D. *Packet structure*

A message is divided into several packets for routing. A packet carries all the routing information with it, i.e. the source address, destination address and also data or payload. For our Topology we consider four fields in a packet i.e. source id, destination id, opline and data.

Opline is used to hold the link details/id, through where the packet needs to be forwarded [8].

Source Id	Destination Id	Opline	Data

Figure 9: Packet structure [8].

III. PROPOSRD ALGORITHM

For better understanding, the routing algorithm is presented below with an example.

Input: New message from external router or client.

Output: Message forwards to the next router or client depending on the destination address.

Procedure steps:

Step 1. Receive the packet and extract the destination id of the packet.

Step 2. Calculate the destination router to which the destination client is connected by the following:

Destination Router = Destination Id/2+1

Step 3. Extract the current router level id and group id.

Step 4. ***If***(level = 1)

If (current router id = destination router)

Deliver the packet to client

Else If (current router id +1=destination router) ***Or*** (current router id -1 = destination router)

Forward the packet to neighbor level 1 router

Else

Forward the message by the added link.

Step .5. Extract the current router group id and calculate the following.

M = pow(2,level)

Destination router group id = floor (destination id/M) + 1

K= (M/2)/2

rId_k = ceil (current router id / K)

destR_k = ceil (destination router Id/K

If (current router group id = destination router group id)

If (level=2)

If (current router = destination router) means same column.

Send the packet directly downward to next level

Else

Send the packet diagonally downward to the next level

If (level>2)

If (current router = destination router) means same column.

Send the packet directly downward by the new added link.

Else

If (rId_k<=2 ***AND*** destR_k<=2)

If (rId_k - destR_k = +-1)

Send diagonally downward.

Else

Send vertically downward.

Else

Send vertically downward.

If (rId_k>2 ***AND*** destR_k>2)

If (rId_k - destR_k=+-1)

Send diagonally downward

Else

Send vertically downward.

If (current router group id +1 = destination router group id) ***OR*** (current router group id -1 = destination router group id)

Send the packet diagonally upward

Else

Send the packet upward by using new added link.

Step .6. End procedure.

Algorithm explained with an example Suppose, Client 0 wants to send a message to Client 8, then following step will be taken.

Step 1: Client 0 send the message to its connected router i.e. router1 in level 1.

Step2: In router 1 extract the packet and get the destination id which is 8. The client is not in the same group so router 1 forwards the packet directly to level 3 router by the new added link.

Step 3: In level 3 current group id of the router is one less than the destination group id of the packet, therefore, the packet will be forwarded diagonally upward.

Step 4: The current router and the destination router are in same column therefore the message will be forwarded by the newly added link directly bypassing an intermediate router.

Step 5: Here the router is in level 2 and the destination router and the current router in same column so the packet will be forwarded downward directly.

Step 6: It is a level 1 router and the destination router and the Current router id is same so the packet is forwarded to the client.

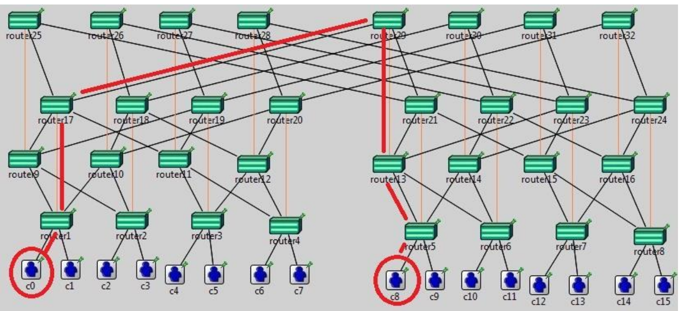

Figure 10: Pictorial depiction of the routing algorithm.

IV. SIMULATION RESULTS

For the purpose of simulating the proposed topology and routing algorithm, OMNeT++ v4.3 network simulator is used, which is a C/C++ based simulator. The hop count and the delay incurred for the simulated network has been recorded and presented for better understanding of the results. The proposed topology is simulated for 8, 16, 32 and 64 clients.

A. Average Hop count

A hop count is defined as the number of intermediate nodes visited by the message while being routed from source to destination. The hop count for MIN FAT – Tree and the proposed topology is recorded and presented below:

Table 1: Average Hop count

Network Size	Average Hop Count	
	Proposed Topology	MIN FAT – Tree topology
8	3.125	4.25
16	3.821	5.5
32	4.75	6.875
64	6.65	8.741

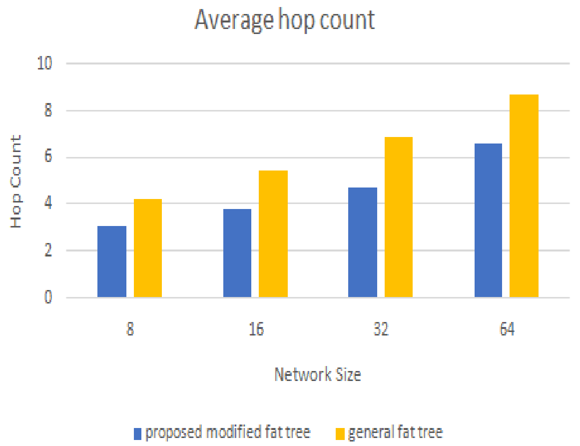

Figure 11: Average Hop count for MIN FAT – Tree (general FAT – Tree) and proposed topology.

B. Delay

Delay is the time taken by the message to reach from source to destination, given the state of the network. Here, delay is being calculated by the difference between creation time of the message in the client source and the arrival time of the message in the client sink. For the purpose of simulation, only random traffic model is being considered

Table 2: Average Delay

Network Size	Average Delay in 10^{-6} Sec	
	Proposed Topology	MIN FAT – Tree Topology
8	0.07625	0.14
16	0.09187	0.1718
32	0.11718	0.6256
64	0.15593	2.4188

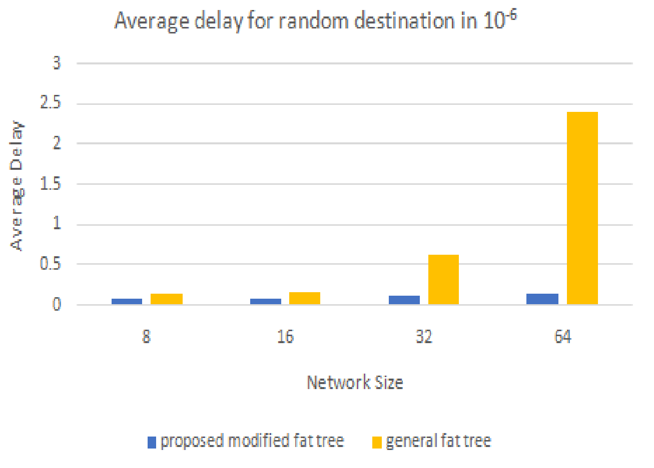

Figure 12: Average delay for MIN FAT – Tree (general FAT – Tree) and proposed topology

V. CONCLUSION AND FUTURE WORK

In this paper, a new improved congestion free modified free FAT – Tree topology is proposed by adding an extra link to connect the alternate rows to reduce the congestion and delay incurred in the original MIN Fat – Tree topology. From the results it is evident that, the proposed topology shows better results from its counterpart. The topology may be further refined in future to minimize the number of routers used by eliminating some of the routers in the intermediate levels so as to minimize the total hardware count. Care must be taken while removing some routers, so that, the removal of routers doesn't compromise the present performance of the FAT – Tree network.

References

[1] A. Biswas, H. J. Mahanta and Md. A. Hussain, "Implementing a partial group based Routing for Homogeneous Fat Tree Network-on-Chip Architecture", IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), May 2014, Ramanathapuram, Tamil Nadu. Pages: 364 - 370, DOI: 10.1109/ICACCCT.2014.7019465

[2] Jie Chen and Cheng Li, Paul Gillard, "Network-on-Chip (NoC) Topologies and Performance: A Review". IEEE Newfoundland and Labrador section,http://necec.engr.mun.ca/ocs2011, 18 oct, 2011.

[3] E Zahavi, G Johnson, D J. Kerbyson and M Lang. "Optimized InfiniBandTM fat-tree routing for shift all-to-all communication patterns". CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE *Concurrency Computat.: Pract. Exper.* (2009). Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cpe.1527.

[4] Sabine R. Ohring, Maximilian Ibel ,Sajal K. Das , Mohan J. Kumar. "On Generalized Fat Trees",In the proceedings of 9th Internation Parellel Processing Symposium, 25 – 28 April, 1995. DOI: 10.1109/IPPS.1995.395911

[5] William J. Dally, Brain Towles. "Principles and Practices of Interconnection Networks". Morgan Kaufmann Publishers. ISBN:0122007514.Book, 2003.

[6] C.E. Leiserson. Fat-trees : Universal networks for hardware efficient supercomputing. IEEE transactions on Computers, C-34(10):892-901, October 1985.

[7] A .Bouhraoua and M. E Elrabaa, "An efficient networkon - chip architecture based on the fat -tree (FT) topology." Microelectronics, 2006. ICM'06. International Conference on. IEEE, 2006.

[8] K. Ray,A.Kalita, A. Biswas, Md. Anowar Hussain "A Multipath Network-on-Chip Topology" in proceding of 5th IEEE internationa conference on information communication and embedded systems "ICICES2016",held in Chennai,India on 25th and 26th february,2016.

[9] Abdelhafid Bauhraoua and M.E Elrabaa. " Improved Modified Fat – Tree Topology Network – On – Chip". Journal of circuits, systems, and computers 20.04(2011):757 – 780.