JECET; March 2014-May 2014; Vol.3.No.2, 820-846.

Journal of Environmental Science, Computer Science and Science and Engineering & Technology An International Peer Review E-3 Journal of Sciences and Technology Available online at www.jecet.org Available online at www.jecet.org

Section B:Computer Science

Research Article

Performance Evaluation of Mesh Networks Under Different Routing Approach

Abhishek Singh¹, Rajiv Singh², Akash Tiwari¹, Dharmesh Kumar²

¹Department of Electronics, Jawahar Lal Nehru College of Technology, Rewa, M. P. India

²Department of Electronics, BBD Institute of Technology & Management, Lucknow, U. P. India

Received: 20 February 2014; Revised: 1 March 2014; Accepted: 10 March 2014

Abstract: The objective of this paper is to develop a Simulation Model to evaluate the performance of Mesh Network for following input parameters, Network Size Load, Number of clock cycles, Routing strategy, Buffer Through the simulation we will observe that How throughput at various loads is affected by Routing strategy

Buffer size, How the Mean and Max Delay is affected by Varying load conditions, Buffer size, How the cell dropped at each stage is affected by Routing strategies Buffer size

Keyword: Wireless Mesh Network, Simulation, Topology,

INTRODUCTION

Large-scale parallel computers are potential candidates for providing very high computational power. These systems are usually organized as an ensemble of nodes, each with its own processor, local memory, and other supporting devices. The nodes are interconnected using a variety of topologies that can be classified into two broad categories: direct and indirect.

Direct Network: In direct networks, each node has a point-to point or direct connection to some of the other nodes, called neighbouring nodes; examples of direct network topologies include hypercube, mesh, and tree. Direct networks have emerged as a popular architecture for massively parallel

JECET; March 2014-May 2014; Vol.3.No.2, 820-846.

computers because of their scalability. The total communication bandwidth, memory bandwidth, and processing capability of the system increase with the number of nodes. The nodes of a direct-network based multicomputer communicate by passing messages through an interconnection network. Neighbouring nodes send messages to one another directly, whereas nodes that are not connected directly communicate with each other by passing messages through intermediate nodes. Support hardware is essential to handle the transmission of message between nodes. In most systems, a router is associated with each node to handle communication-related tasks. Dedicated routers are also used to allow overlapping of computation and communication within each node.

In direct-network-based multicomputers, a task is allocated to a group of nodes that communicate for successful execution of the task. The speed of execution depends on the processor as well as on the communication performance. The latency incurred by a message traversing from a source node to a destination node affects the overall performance of the multicomputer system. Because of the inter processor interactions, the communication latency also affects the granularity of parallelism that can be exploited in the system. Thus it is essential to devise techniques that reduce the communication latency incurred in Direct Network. The communication latency is the most important performance metric in direct networks. It comprises start-up latency, network latency, and the blocking time.

The start-up latency is the time required for the system to handle the message at the source and destination nodes and depends primarily on the design of the interface between the local processors and routers. Network latency, defined as the time spent by a message in the network, is computed as the time between the instant when the message head is injected into the network by the source and the instant when the message tail is absorbed by the destination node. Both start-up and network latencies are fixed for a given network. The blocking time of a message is the time spent waiting for a channel currently being used by another message? Thus the blocking time depends on the resource contentions that a message encounters in its path. Blocking time cannot be determined statically, as it depends on the network traffic distribution and the path taken by a message.

Indirect Network: In indirect networks, the nodes are connected to other nodes or a shared memory through one or more switching elements. Examples of indirect networks include crossbar, bus, and multistage interconnection network.

For communication in Indirect Network first we, have to communicate through switches. Means, we can not communicate to next node directly sending message like in Direct Network, but this is possible only by using switches. These switches may be of different type. Switches may of several types like Crossbar, Direct switch.

Direct Network Topologies: The topology of a network defines how the nodes are interconnected and is generally modelled as a graph in which the vertices represent the nodes and the edges denote the channels. Multidimensional meshes and k-ray n-cubes.

The one dimensional mesh can be generalised to a k- dimensional mesh, where each node (except boundary nodes) has 2k connections. Again ,boundary nodes can be connected, but there is no general consensus on what to do on boundary nodes, however ,this type of topology is not suitable to build large scale computers, since the maximum message latency ,that is ,the maximum delay of a message from one of the N processors to another, this is bad for two reasons firstly there is wide range of latencies(the latency between neighbouring processors is much lower than between not neighbours),and maximum latency grows with the numbers of processors⁵.

In star topology there is one central node, to which all other nodes are connected; each node has one connection, except the central node,



Fig. 1: Star topology

Which has N-1 connections? Stars are also not suitable for large systems, since central node become a bottleneck with increasing numbers of processors [6].Completely connected networks guarantee fast delivery of message from any source to any destination node (only one link has to be traversed).Completely connected networks are however expensive in terms of the number of links needed for their construction. This disadvantage becomes more and more apparent at higher values of N. It should be noted that the numbers of links in a completely connected network is given, i.e.0 (N^2).

For N=16, this means 136 connections. Even large scale integrations is hard passed to accommodate the 524,288 links needed to fully connected 1024 processors. However the diameter is 1, the bisection width is high and there exist many alternatives routes between a pairs of processors. In a tree network there is only one path between any two nodes. The taller the tree, the higher is communication bottleneck at high level of the tree.

Properties of Interconnection Network: An interconnection Network should transfer a maximum number of messages in the shortest time with minimum cost and maximum reliability. The most important requirement of the interconnection networks is:



Fig 2: Tree Network

Small Diameter and small average distance: Small average distance allows small communication latency, epically for distance sensitive routing, such as store and forward .but it is also crucial for distance insensitive routing, such as wormhole routing , since short distance imply less used links and buffers and therefore communication contention.

Small and fixed vertex Degree: Small and constant vertex degree allows simple and low cost universal routers, which amortizes the design costs. On the other hand, it implies less links and lower connectivity and larger distance⁷.

Bisection Width: Bisection width is the minimum number of wires that must be cut to divide the network into two equal halves, high bisection width is better because in Algorithm requiring large amounts of data movements, the size of the data set divided by the bisection width puts a lower bound on the complexity of the parallel algorithm⁷.

High Connectivity and Fault tolerance: The network should provide alternative paths for delivering message in case of link or nodes faults or communication congestions. Large packets can be delivered faster if they can be split into smaller chunks sent along disjoints paths.

Symmetry: This is a very important requirement. Many IN topologies are vertex or edge symmetric. Intuitively, a symmetric network is easy to understand and the design of parallel and communication algorithm is very much easier.

Arbitration Policy: Arbitration policy tells that when at a node more than one packets are coming in the same clock cycle then which packet will forwarded and which one will be dropped. There are two types of Arbitration policy is used.

Min distance: Here in this policy that packet will be forwarded which has to travel least distance to reach its destination and others will be dropped.

Max distance: Here in this policy that packet will be forwarded that has travelled the maximum distance from its source and others will be dropped.

If all the packets have to travel same distance then use random one.

Buffer and Routing Strategy:

Here the Buffer strategy is that Simulation is done by using different size of buffer at the Nodes.

For Routing Strategy, since for source and destination address we have used (X, Y) co-ordinate system. So, for a packet to send its destination we have to equalize source address to its destination address. It can be done by two ways.

First is, Equalize X co- ordinate of source address of a Packet to X co-ordinate of its destination address first and then equalize Y co-ordinate.

Second is, Equalize Y co-ordinate of source address of a Packet to Y co-ordinate its destination address first and then equalize X co-ordinate.

Mesh Network: An n-dimensional mesh is an interconnection structure that has $k_0 \times k1 \times \ldots \times kn_1$ nodes, where ki denotes the number of nodes in the ith dimension. Each node in the mesh is identified by an n-coordinate vector $(x_0, x_1, \ldots, x_{n-1})$, where $0 \le xi \le ki - 1$. Two nodes $(x_0, x_1, \ldots, x_n - 1)$ and $(y_0, y_1, \ldots, y_n - 1)$ are connected if and only if there exists an i such that $x_i = y_i \pm 1$, and $x_j = y_j$ for all $j \ne i$. Thus the number of neighbours of a node ranges from n to 2n, depending on its location in the mesh

Simulation Strategy: The operation of the simulation has been organized in two units mainly, Cell Generation and cell transmission units. Cell generation unit is mainly concern with the generation of cell in each clock cycle according to the load (i.e. perchantage of nodes generating request) and Dimension of the Network.

Cell transmission unit transmits the cells from source to its final destination via different nodes. A set of The mesh network is a asymmetrical topology in which the node degree depends on its location.

Interprocessor communication performance depends on the location of Source and destination. The channels near the center of the mesh experience higher traffic density than those on the periphery. The torus and hypercube are symmetrical topologies in which the degree of a node is the same irrespective of its location in the network. Thus unlike the mesh, all the nodes in tori and hypercube are identic in Connectivity.



Fig.3: Two Dimensional Mesh (4×5)

Properties of Mesh Network:

Diameter: The diameter of a graph is the maximum eccentricity of any vertex in the graph. i.e. it is the greatest distance between any two vertices.

The eccentricity of a vertex v is the greatest distance between v and any other vertex.

For a man mesh network.

Diameter = (m+n-2)

The network diameter of a mesh is greater than that of the torus, which in turn has a greater diameter than the nodes hypercube for the same number of nodes.

Degree: In graph theory, Degree (or valence) of a node is the number of Edge incident to that node. The degree of a node v is denoted by deg (v). The maximum degree of a graph (G) is denoted by Δ (G) and minimum degree is denoted by δ (G).

For a undirected m×n mesh network number of edge will be

m (n-1) +n (m-1).

And also total degree of a network is double the number of edge.

So Degree of mesh network=2[m (n-1) + n (m-1)]

Average Path Length: The average path length of m×n mesh network (where m and n are even)

$$\sum(m+n-2) + \sum(m+n-3) + \sum_{i=0}^{i=(n-3)} (\sum(m+i-1) - \sum(i+1))/((m*n)-1)$$

Bisection Width: The bisection width of a network is defined as the number of channels that must be removed to partition the network into two equal sub networks. The bisection width has a significant effect on the interprocessor communication performance [3]. The bisection width (BW) of a $2^n \times 2^n$ 2-D mesh, $2^n \times 2^n$, 2-D torus, and a 2*n*-cube hypercube are BWmesh = 2n, $BWtorus = 2^{n+1}$, $Hypercube = 2^{2n-1}$, respectively [1]. The bisection density, which is the product of the bisection width and the channel width, can be used as a measure of the network cost. For the same cost, the 2-D mesh can support wider channels than the 2-D torus, which in turn can support wider channels than the hypercube. Thus the channel bandwidth of the three topologies can be

Expressed as: mesh > torus > hypercube.

In general, low-dimensional meshes are preferred because they have low fixed-node degrees and fixed length channel wires, which make them more scalable than high-dimensional meshes and k-ary n cubes. Low-dimensional meshes also have higher channel bandwidth per bisection density and have lower contention and blocking latencies, which results in lower communication latencies and higher hotspot throughput [3].

Experiments are conducted with respect to the dimension of the network at varying load condition.

First we will define Simulation Input parameter and simulation output quantities. The assumptions that are followed in simulation are also written the simulation model (Algorithm and Flowchart) and program definition is described next.

The Simulation Program is general in nature which can study different network topologies.e.g Mesh, Torus, Hypercube. However for present study, we are confining to mesh network only. The topology of the mesh network id defined by its Dimension only.

The simulation program is written in c.

Simulation Input Parameters:

The simulator receives the fallowing parameters.

1. TOTAL_CLOCK (CLK), Total number of simulated clock cycle.

```
CLK=1000
```

```
2. NET_SIZE, Size of the Network (N \timesN)
```

```
2<=NET SIZE<=10
```

3. LOAD, Various loading condition in Perchantage

4. BUFFER, Size of the buffer provided at each Node within the Network. 0<=BUFFER<=2

The simulator performs the experiments by setting the load value from 5% to 20% for different Dimension of Network. For each specific value of load, the simulator generates request uniformly distributed.

Output Parameter:

- 1. THROUGHPUT, Perchantage of successfully transmitted request.
- 2. PACKET DROPPED, Number of packet dropped in simulation.
- 3. THROUGHPUT according to Network size
- 4. THROUGHPUT according to Buffer size.

In the simulation the assumption are written as below.

- 1. All the packets are moving per hope per cycle.
- 2. Initially there is no buffer.
- 3. When Buffer will be there at NODE then for contention FCFS (First come first serve) service is used.

Cell Generation Unit: In each simulation clock cycle, each source is either generating a Cell or remains idle, depending upon the random number which has been generated for cell's address i.e. which address cell is going to communicate. When a cell is generated a set of attributes is attached that show the characteristics of that particular cell.

The random number is generated by Residue Method.

i.e. $r_{i+1} = (ar_i+b) \mod(m);$

Where a, b and m are constants and r [0] is our seed according to our requirements.

Data Structure and Algorithm: Data structure of packet and elements of packets is given below.

1.Source Address: For the mesh network the source address will be in the form of (x_i, y_i) and this will vary from to (NET_SIZE-1) and the value of x_i and y_i will be stored in array a [2] with the value of a[0] as x_i and a[1] as y_i .

2. **Destination Address**: For the mesh network the destination address will be also in the form of (x_i, y_i) and this will vary from to (NET_SIZE-1) and the value of x_i and y_i will be stored in array b [2] with the value of b [0] as x_i and b [1] as y_i .

3. Flag: Flag is used for the finding out the packet reached its destination or not. If the flag value is '0' the packet is at its destination and if flag is '1' the packet is either at its destination or dropped.

4. Gen_time: Present clock cycle of the simulation will be the generation time for that packet.

Algorithm for Cell Generation and Transmission: This algorithm explains that the request is generated randomly by Residue Method according to the Load, number of clock cycles and Network size. And packets are generated using function Rajeev and transmitted to its destination using functions change stage and networking.

Begin:

1. Get the parameter of the simulation

/* Network size */

/* Number of Clock cycle */

/* Load: perchantage of nodes generating their request*/

1. Calculate number of Nodes that's generating request per cycle.

/* Number will depends upon the size of the network according to the load applied */

- **2.** Calculate the number of Random number that has to be generated every cycle for source and destination address. For one source and one destination address i.e. for one packet four Random number will be generated.
- **3.** Initialize the seed =1234; for Random number generation.
- 4. do while the number of simulation cycle equal to the number of clock cycle i.e. num_clk

Change the seed for different random number

Call the function rajeev ();

Call the function change _stage ();

Call the function network ();

increase the clock cycle.

End.

Algorithm for Function rajeev : This algorithm explains that the address of requesting Nodes (randomly generated) is stored in an array and then packets are generated by the Nodes using Dynamic memory allocation.

Begin:

1. call the random number generation function and generate only that number of random number that is required for one cycle

/* Equal to the number of Packets per cycle \times 4*/

- 2. store all that number in an array r2[r5];
- 3. Using Dynamic Memory Allocation, allocate memory to a packet.

/* Assign source address and Destination address in a[2] and b[2] respectively using already stored random number in an array r2[r5]*/

/* also assign flag=0 and gen_time present clock cycle*/

- 4. Go to step '3' for allocating memory to Packets and assign respected values.
- 5. Allocation will happen, to number of Packets (communicating) per cycle.

End.

Algorithm for function change_state : This algorithm is used for copying source address into another temporary array and comparing source address to destination address, if address is not same then increase/ decrease X co-ordinate or Y co-ordinate to equalize the destination address according to our routing strategy.

Begin:

1. Copy source address a[2] of each packet in another array c[2] for Routing .

/* each packet: packets generated in one cycle*/

2. Compare the source address i.e. c [2] to destination address b[2]

If (both are same)

```
Go for the next packet. Else. add or subtract '1' to c[0] or c[1] according to the value of b[0] and b[1].
```

/* adding /subtracting should once for each packet for each cycle*/

3. Go to step 2 for next packet Routing.

Algorithm for function networking: This algorithm is used for checking the Packet's address. If starting address is equalizes to destination address then increase SUCCESS by one else check that temporary address of a packet is equal to another's packet then apply iteration policy for dropping one and forwarding another.

Begin:

1. Check the Flag of first Packet

If (flag==1)

go to step 1 for second packet and so on.

else

Again check the next stage of Packet

if(next stage of first Packet is equal to next stage of second, third or more)

Go to minimum distance function

Put the flag=0 for non minimum distance Packet With the help of minimum distance function

Increase the value of DROPPED Packet by one.

if (next stage of Packet is equal to Destination Address)

put the value of Flag=1;

Increase the SUCCESS packet by one.

End.





Fig. 4: Flow Chart of Main Program











Fig. 7: Flow Chart of Function Change_Stage



Ι

RESULT & DISCUSSION

Table-1: For Network 4×4 the Average path length by Analytical Method Is 2.667 and Diameter is 6.Average path length by Simulation when 10% Nodes are generating their request

Dimension	No of CLK	Diameter	APL
4	10	4	2.80
4	15	4	2.60
4	20	4	2.45
4	30	4	2.40
4	50	6	2.52
4	100	6	2.32
4	200	6	2.36
4	300	6	2.37
4	400	6	2.425
4	500	6	2.428

Chart 1:



Table-2: For Network 6×6 the Average path length by Analytical Method Is 4.00 and Diameter is 10. Average path length by Simulation when 10% Nodes are generating their request

Dimension	No of CLK	Diameter	APL
6	5	7	3.33
6	10	7	3.80
6	20	7	3.93
6	40	8	3.99
6	50	10	3.967
6	100	10	3.807
6	200	10	3.813
6	300	10	3.808
6	400	10	3.833
6	500	10	3.834

Chart. 2:



Table-3: For Network 8×8 the Average path length by Analytical Method_ is 5.34 and Diameter is 14 Average path length by Simulation when 10% Nodes are generating their request

Dimension	No of CLK	Diameter	APL
8	5	10	5.433
8	10	10	5.42
8	20	10	5.439
8	40	10	5.233
8	50	14	5.143
8	100	14	5.140
8	200	14	5.138
8	300	14	5.132
8	400	14	5.148
8	500	14	5.150

Chart 3:



Dimension	No of CLK	Diameter	APL
10	5	13	6.84
10	10	13	6.84
10	20	13	6.75
10	40	16	6.46
10	50	16	6.374
10	100	16	6.50
10	200	16	6.432
10	300	16	6.395
10	400	16	6.404
10	450	16	6.423

Table-4: For Network 10×10 the Average path length by Analytical Method_ is 6.667 and Diameter is 18 Average path length by Simulation when 10% Nodes are generating their request





From the table and the Chart we can see that the Throughput is Low at lower clock cycles and increase as clock cycles increase but from a certain clock cycle it decreases and at gets saturates at particular value of Throughput.Routing strategy is Equalize X first then Y.

Table-5:This table shows the Throughput of 4×4 Network when Buffer size is **Zero** and **10 %** of the Nodes are Generating the Request

No of	No of Packets	No of Packets	No of Packets	Throughput
Clocks	Gen	reached at	Dropped	(%)
		Destination		
50	50	45	03	90.0%
75	75	71	03	94.5%
100	100	95	03	95.0%
200	200	190	08	95.0%
400	400	382	17	95.5%
600	600	570	29	95.0%
800	800	762	37	95.25%
1000	1000	952	45	95.20%

JECET; March 2014-May 2014; Vol.3.No.2, 820-846.

1200	1200	1140	57	95.0%
1400	1400	1327	71	94.78%
1500	1500	1420	78	94.7%
1600	1600	1515	84	94.70%

Chart 5:



From the table and the Chart we can see that the Throughput is Low at lower clock cycles and increase as clock cycles increase but from a certain clock cycle it decreases and at gets saturates at particular value of Throughput.

Routing strategy is Equalize X first then Y.

Table-6: This table shows the Throughput of 5×5 Network when Buffer size is Zero and 5% of the Nodes are Generating the Request

No of	No of Packets	No of Packets	No of Packets	Throughput
Clocks	Gen	reached at	Dropped	(%)
		Destination		
50	50	42	06	84.0%
100	100	89	08	89.0%
200	200	181	16	90.5%
400	400	376	22	94.0%
600	600	552	46	92.0%
800	800	730	68	91.25%
1000	1000	900	97	90.0%
1200	1200	1073	124	89.41%
1400	1400	1250	148	89.28%
1500	1500	1337	161	89.13%
1600	1600	1424	125	89.00%

Chart 6:



From the table and the Chart we can see that the Throughput is Low at lower clock cycles and increase as clock cycles increase but from a certain clock cycle it decreases and at gets saturates at particular value of Throughput.Routing strategy is Equalize X first then Y.

Table- 7: this table shows the Throughput of 5×5 Network when Buffer size is Zero and 10% of the Nodes are Generating the Request.

No of	No of Packets	No of Packets	No of Packets	Throughput (%)
Clocks	Gen	reached at	Dropped	
		Destination		
50	100	60	37	60%
75	150	97	50	64.69%
100	200	135	60	67.50%
200	400	272	123	68.0%
300	600	418	177	69.67%
400	800	559	238	69.87%
500	1000	697	300	69.70%
600	1200	834	363	69.60%
700	1400	976	420	69.70%
800	1600	1122	474	69.90%

From the table and the Chart we can see that the Throughput is Low at lower clock cycles and increase as clock cycles increase but from a certain clock cycle it decreases and at gets saturates at particular value of Throughput.Routing strategy is Equalize X first then Y.

Chart 7:



Table-8: this table shows the Throughput of 6×6 Network when Buffer size is **Zero** and 5 % of the Nodes are Generating the Request.

No of	No of Packets	No of Packets	No of Packets	Throughput
Clocks	Gen	reached at	Dropped	(%)
		Destination		
100	100	92	05	92.00%
200	200	187	10	93.50%
300	300	282	15	94.00%
400	400	378	19	94.50%
500	500	472	25	94.40%
600	600	566	31	94.30%
700	700	661	36	94.43%
800	800	756	41	94.50%
900	900	851	46	94.50%
1000	1000	937	58	94.00%

Chart 8:



From the table and the Chart we can see that the Throughput is Low at lower clock cycles and increase as clock cycles increase but from a certain clock cycle it decreases and at gets saturates at particular value of Throughput.

Routing strategy is Equalize X first then Y.

Table-9: This table shows the Throughput of 5×5 Network when Buffer size is **one** and **10%** of the Nodes are Generating the Request.

No of	No of Packets	No of Packets	No of Packets	Throughput
Clocks	Gen	reached at	Dropped	(%)
		Destination		
50	100	90	05	90.00%
75	150	140	06	93.33%
100	200	187	06	93.50%
200	400	384	09	96.00%
300	600	583	10	97.17%
400	800	784	11	98.00%
500	1000	978	17	97.80%
600	1200	1172	23	97.67%
700	1400	1366	29	97.56%
800	1600	1561	34	97.56%

Chart 9:



From the table and the Chart we can see that the Throughput is Low at lower clock cycles and increase as clock cycles increase but from a certain clock cycle it decreases and at gets saturates at particular value of Throughput.Routing strategy is Equalize X first then Y.

Table-10: This table shows the Throughput of 6×6 Network when Buffer size is **one and 10%** of the Nodes are Generating the Request

No of	No of Packets	No of Packets	No of Packets	Throughput
Clocks	Gen	reached at	Dropped	(%)
		Destination		
50	150	126	13	84.00%
60	180	154	18	85.56%
75	225	200	19	88.88%
100	300	262	29	87.33%
200	600	524	66	87.33%
300	900	791	100	87.88%
400	1200	1049	139	87.41%
500	1500	1318	170	87.86%

Chart: 10



Chart: 11

This chart shows that the Throughput increase with increasing with Buffer for same Load. Routing strategy is Equalize X first then Y.



Chart: 12

This chart shows that Throughput decreases with increasing the Network Size for the same Load. Hence Mesh Network is very suitable when there are less number of Parallel Processors are there. Routing strategy is Equalize X first then Y.



Chart:13 Comparison of Throughput for different routing strategies. This chart shows the Throughput of 5×5 Network with buffer zero and at 10% Load.

We can see that Throughput of equalize Y first and then X is greater than equalize X then Y Routing strategies.



Chart:14 Comparison of Throughput for different routing strategies. This chart shows the Throughput of 6×6 Network with buffer zero and at 10% Load.

We can see that Throughput of equalize Y first and then X is greater than equalize X then Y Routing strategies.



Chart: 15 Comparison of Throughput for different routing strategies. This chart shows the Throughput of 7×7 Network with buffer **zero** and at **10%** Load.

We can see that Throughput of equalize Y first and then X is greater than equalize X then Y Routing strategies.



Chart: 16 Comparison of Throughput for different routing strategies. This chart shows the Throughput of 8×8 Network with buffer zero and at 10% Load.

We can see that Throughput of equalize Y first and then X is greater than equalize X then Y Routing strategies.



Chart: 17 Comparison of Throughput for different routing strategies. This chart shows the Throughput of 10×10 Network with buffer zero and at 5% Load.

We can see that Throughput of equalize Y first and then X is greater than equalize X then Y Routing strategies.







Chart: 19 Comparison of Throughput for different Arbitration Policies. This chart shows the Throughput of 10×10 Network with buffer zero and at 5% Load.

Arbitration 1: Min distance

Arbitration 2: Man distance



CONCLUSIONS:

In this paper we developed a Simulation model for Mesh Network with the following condition.

- A set of routing strategies that Equalize X co-ordinate first then Y and Equalize Y co-ordinate first then X.
- Traffic distribution is uniform over all outputs.
- Simulation can be done at various Loading condition.
- Average Path length from Analytical method is very near to simulation method.

- Throughput is going to increase as number of clock cycles increase but after a certain number of cycles it decreases and then gets saturates.
- Throughput increases with buffer size changes from '0' to '1'.
- Throughput decreases with increases the size of Network for the same Load.
- Throughput is better in Equalize Y co-ordinate first then X Routing strategy than Equalize X coordinate first then Y.
- Throughput is greater in the case of Arbitration Policy 1 because in first policy that packet will be forwarded that has to travel min distance to reach its destination.

REFERENCES:

- 1. P.Mohapatra, "Wormhole Routing Techniques for Directly connected Multicomputer Systems" ACM Computing Surveys (CSUR), Vol 30, no 3, 1998
- 2. R.V. Boppana, S. Chalasani, "Fault Tolerant Wormhole Routing Algorithms for Mesh Network "*IEEE Trans. Computers*, vol 44, no 7,1995
- 3. W.Dally, "Performance Analysis of k-ary n-cube Interconnection Networks" *IEEE Transaction on computer*, vol C-39, no 6, 1990
- 4. Amit Jain, "Performance Evaluation of Gamma Interconnection Network under different routing strategies" *A Dissertation Report*, Electrical Engg, 2003
- 5. T.Y.Feng, "A Survey of Interconnection Networks" *IEEE Computer, special issue on Interconnection Networks*, vol C-30, no 12, Dec 1981
- 6. L.N Bhuyan, O.Yang and D.P.Agrawal, "Performance of Multiprocessors or Interconnection Networks" *IEEE Transactions on Computers, vol.C*-38, no 2, Feb 1989
- 7. Michael J.Quin, "Parallel Computing Theory and Practice" TMH, New Delhi 2002.
- 8. A.M.Tenenbaum, "Data structures using C and C++" *Book Peasson, Prentice Hall Publication*, Vol 2, 2007

*Corresponding Author: Abhishek Singh

Department of Electronics, Jawahar Lal Nehru College of Technology, Rewa, M. P. India