Azerbaijan Journal of High Performance Computing, Vol 3, Issue 1, 2020, pp. 94-118
https://doi.org/10.32010/26166127.2020.3.1.94.118

Fault-Tolerant Adaptive XY Routing for Multiprocessors in HPC Network

Pankaj Dadheech, Ankit Kumar

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur, Rajasthan, India, pankajdadheech777@gmail.com, iiita.ankit@gmail.com

*Correspondence: Pankaj Dadheech, Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur, Rajasthan, India, pankajdadheech777@gmail.com

Abstract

Interconnection of networks use in multiprocessors multi-computer and distributed shared memory architecture; basically, it connects many networks simultaneously in each time interval. The objective of this paper is to compare the simulation results in certain functionalities of Network on Chip. It's a wide area of research on routing and topological structure. The archi-tecture of NoC is scalable network architecture. Point to point interconnection of links, switch functionality is used, verity in routing algorithms, topologies provide enhance performance as per efficiency, throughput, Latency, channel allocation manner, and comparison with the previ-ous methodology of the chip. This paper focuses on the fault tolerance adaptive routing on HPC mesh and compares its results with already implemented 2D mesh topology with parame-ters like path diversity, Total Power Consumed, Latency, Throughput, and Fault Tolerance.

Keyword: Networks, Interconnection, Network on Chip (NoC), Efficiency, Throughput, Latency, Channel Allocation.

1. Introduction

In the last 15 years, the growing on-chip design market inspires industrial researchers and in-verters to proposed a better-optimized solution to improve on-chip communication among IP cores (Radetzki, M., Feng, C., Zhao, X., & Jantsch, A., 2013). Network on the chip is futuristic technology, which is part of multiprocessors on a single chip. These multiprocessors are a combination of thousands of processors. The idea of manufacturing of this chip is reflecting by Moore Law (Daneshtalab, M., 2011; Chen, C. L., & Chiu, G. M., 2001). Moore law specifies that if increasing cores in devices have the results of high performance at low power process budget and no need to add multiple cores on this.

Complex topologies of NoC are proposed dynamically, which helps to adjust the power of nodes and links (Bogdan, P., Dumitraş, T., & Marculescu, R., 2007; Grecu, C., Ivanov, A., Saleh, R., & Pande, P. P., 2006). This paper proposed the solution of power optimization, calculation of power consumption based on the transmission

path when a low traffic rate is there. It uses the connecting links only directly attached to the source and destination nodes instead of using a full network area (Bjerregaard, T., & Mahadevan, S., 2006). It operettas on four sub-networks using a single chip here; every node is connected with its neighboring nodes. Its structure is di-vided into physical, data, network, transport layers. It's a more exciting field for students be-cause implementing networks on a single chip is most innovative.

1.1. Interconnection of Networks

It is configured in VLSI, switches, and transmission devices. Interconnects networks internally connected with other sub-networks. These network use in microprocessors devices. It is used in telephone networks and manageable communication devices. These networks integrate LAN, WAN, and system area networks. Multiple processors are work on a parallel computer. High bandwidth large memory space to store the results and also high-speed RAM needs for fast processing. Parallel computers are a combination of multiple Subsystems, interconnections, memory devices, controllers, etc. It is a combination of programming code and electronic devic-es. The goal of this network creates strong bonding between nodes and channels.

1.2. Performance Key Metrics

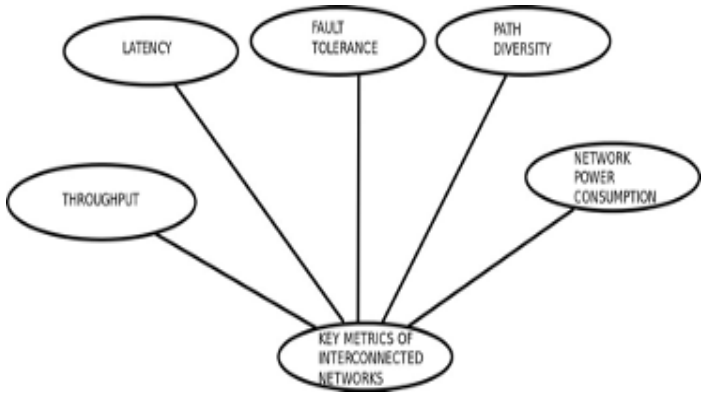

Fig. 1: Performance Key Metric

1.2.1. Latency

Latency is a time unit for Packet for traversing the network. It is the time between the header parts of the message reaches the input terminal and tail of Packet departing from the output ter-minal. Message latency is a time-sensitive issue; it is the time elapsed between the time con-sumed by a message generated at the source node, and the time message is delivered at the des-tination node. This latency id directly effects on processors idle time. If Latency is small in the network, then network performance is high.

1.2.2. Head Latency

This is time for the head of the message to traverse the network. The Head message is an iden-tical part of the information (Th).

Serialization of Latency: (Ts=L/b) this is time for the tail of the message to sends a complete packet across the channel. It contains all the information on the tail part and traverses on the network

$$T = Th + L/b.$$

The complete Latency is head latency and serialization latency.

1.2.3. Average zero load latency (T0)

This Latency is counted on zero load length, which is provided by or bound on average Laten-cy. It is used the absence of contention in the network

$$T0 = Hmin*TR + Dmin/v + L/b,$$

which is equal to router delay + time to flight + serialization latency or (Router delay) + (Time of flight) + (serialization latency) distance. Where Dmin - average distance, v propagation ve-locity, L packet length, b channel bandwidth. These contents are if combined; then, total Laten-cy is the packaging of these properties (Lehtonen, T., Liljeberg, P., & Plosila, J., 2007; El-Moursy, M. A., & Ismail, M., 2009).

1.2.4. Throughput

Throughput is the network data rate based on port. Throughput is major as bit per unit-a number of operations completed per unit of time. Throughput considers for all the network range, routing switching, and topologies. Throughput is dependent on buffer allocation of source and destination, and buffers works on virtual channels – the number of virtual channels increases, then throughput increases (Camacho, J., & Flich, J., 2011).

1.2.5. Path Diversity

It is defined that if maximum numbers of minimal paths between most of the pair of the node is higher robust, then a network has one single route between nodes. A number of maximum minimal paths are available is better than if the only single path is available between nodes. Path diversity is useful for getting surety to sends with a particular destination. Path diversity in-creases the robustness of the network, and this feature is achieved by load balancing across channels. High path diversity networks always get high throughput results (Duato, J., Yalamanchili, S., & Ni, L., 2003).

1.2.6. Network Power Consumption

Power consumption is the most important field for every topology. Different type of topologies is the basis of its injection mechanism and its production, implementation cost. The most im-portant related as per is power consumption, energy-saving

issue. In dimensional order routing, which is used in a mesh topology, .the other side of power consumption is the best network utilization. This power process also consumed some power and compensated for which period powered off. For saving power, it should be remembered that components never use powered on again until the completion of a certain number of cycles. This number of cycles does not fixed, but it is greater than 9. PML is a waked control signal between sources to destination those signals, which are either neighbor switches or attached with end nodes. Only those nodes are those who participate in that transmission time. If the message arrives at that time when switches power off then control signal catch the request and then these control signal will wake switch again. So less power consumption is related to high throughput (Harbin, J., & Indrusiak, L. S., 2013).

1.2.7. Fault Tolerance

Fault crated at any network, and it slows down the network efficiency and increases the net-work performance need to create a network that increases the efficiency of against faults. If you need to create a big network, then fault tolerance is a good factor. It is an ability to operate in the presence of faults. There are many types of fault tolerance that are network on the chip but first need to understand fault. Types of fault classes in NoC Transient Fault, Permanent Fault, Intermittent Fault.

2. Literature Review

Lehtonen, T., Wolpert, D., Liljeberg, P., Plosila, J., & Ampadu, P. (2009) proposed network pow-er consumption is based on power management control logic (PML). PML controllers the on-off power process. PML decides that when and how much time power should ON or OFF, the control signal will weaken the switch again.

This technique divides into the following steps:

I. When the network not used for sending receiving packets, than PML, is powered down to the network.

II. The second step is how much period switch's power off; it depends on the clock cycle, which attached to each node). The PML is in charge of deciding when switch and its at-tached clock are powering down. The following actions are performed locally within the switch in this case.

III. Only those nodes are on which participate in transmission from source to destination other nodes which not participated at that than PML also switch off these nodes for power saving purpose (Sharma, A., Tailor, M., Bhargava, L., & Gaur, M. S., 2018).

IV. High throughput is directly proportional to less power consumption. The main goal of the power management factor is to increase the time when the switch's input nodes and channels are powered off. The power-saving factor is work on this. The algorithm works in multiple steps near about 3 steps.

V. The first one is deciding that when power can switch off.

VI. The routing algorithm decides that which nodes are powered on or which nodes are powered off.

VII. The third one in at what time cycle nodes is again activated.

These key steps describe the overall process of the switch on and OFF of the nodes and chan-nels. This power optimization increases the efficiency of the chip. Power calculation factors are typically used in the adaptive routing algorithm. In this work, we calculate only the current node's power calculation.

This paper includes the power relational function and design of HPC mesh topology and also compares with PC and NR mesh topology. It describes the tile-based design of topology and what is the process of fault injection and tolerance is in this process.

Bouguettaya, A., Kimour, M. T., & Toumi, S. (2014) proposed the PNC method based on power-efficient network calculation. It is based on slack that further delayed in packet sending is pos-sible without violating its deadline. This power gating method reduces active buffer size, less active buffer, and lower voltage consume less power; its simulation shows that it consumes 50% of total power consumption. de Melo, D. R., Zeferino, C. A., Dilillo, L., & Bezerra, E. A. (2019) proposed in a solution to replace the classic interconnection method of SoC (System on Chip). In this paper, the author uses a clustering-based fault-tolerant routing algorithm in 2d mesh. In this method, each network divides into a certain cluster, and each cluster has a header, which contains information of other members and network state.

Proposed network related by several factors: phase behavior with local and temporal bursts shows the variation in inoculating traffic behavior, the data's locality-of reference, miss behav-ior of applications, application phase behavior (Yu, Q., & Ampadu, P., 2011; Chen, C., Lu, Y., & Cotofana, S. D., 2012; Dally, W. J., & Towles, B. P., 2004). Scheduling issue-if multiple core's application is simultaneously run on-chip if they do not schedule. Hence, issue occurs, crucial problem, multi-dimensional optimization problem -it which exploiting the flexibility of sched-ule, it also minimizes performance and power, Trojan injection-when some outer attacker at-tacks on data or attacker may distribute Trojans across multiple cores of NoC, Security delay sending and receiving, failure of critical circuitry, leak confidential information, insert malicious logic.

Benini, L., De Micheli, G., & Ye, T. T. (2006) proposed multi-core facility in Network on Chip extract from the Multi-Core System on Chip(MC SOC). Network on chip resolves the problem of interconnecting multiple cores on a single system on chip; these problems are like delay in non-scalable global wires, globally non -synchronized, etc.

Joshi, A., Batten, C., Stojanović, V., & Asanović, K. (2008) proposed network on-chip is a new technology, which becomes a higher level of circuit design. This paper presents functionality and methods for fault and fault tolerance of Network-on-Chip (NoC) and its devices. The fault model and topology considered and explained in

this research. During the transmission period, certain issues are occurred related to faults, and some faults remain lifetime period, so needs to diagnosis the faults, some fault model created to fault tolerance.

2.1. Routing

Routing is a method that defined what path chooses to send packets to the destination. The goal of routing is to distribute packets to paths of the network, which defined by topology structure. Specific routing path are choose by routing algorithm, and it shows effects on throughput and power consumption. The most important and widely used routing algorithm used in NoC is the dimension ordering routing algorithm. Dimensional ordering routing based on dimensions X, Y, Z, etc. Source has choices to sends a packet in unicast or multicast order. If Packet sends in unicast order, then a source and a destination are exchanged for the packets. But if one source and more than one destination in the network, then it is called multicast routing, and Packet is sent at multiple destinations. Multicast routing is used to get the fastest results. Multiphase or multicast routing is when there are multiple destinations are available from source to destina-tion, and distributed paths are available for sending packets from source to destination. These paths have balanced the load of channels.

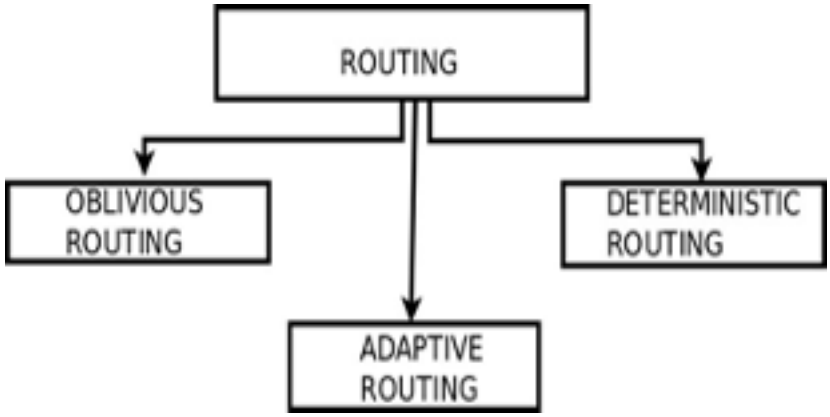

Fig. 2: Types of Routing

2.1.1. Types of Routing Algorithm

a) Deterministic Routing: Dimension ordering routing routes packets either X-Y direction or Y-X direction, but if the deterministic dimension is X-Y type, it moves the first west side or east side then it moves north or south side according to destination path but it does not turn back to the east or west side. If routing is Y-X types it over first to the north or south side, then it moves east or west side, but it does not turn back to the north or south side again. It's a disadvantage deterministic dimension ordering routing. It is a free deadlock because there is a single path is available from

source to destination. It is simple, and results give low throughput. But in the case of path diversity and load balancing, it shows bad re-sults (Dally, W. J., 1991).

b) Oblivious Routing: Oblivious routing describes that many paths are available from source to destination, but this algorithm chooses the path without any relates to network conges-tion for an example if the origin (0,0) and destination is (2,1).then oblivious routing would choose X-Y path or Y-X path (Dally, W. J., & Seitz, C. L., 1988).

c) Adaptive Routing: The most effective routing is adaptive routing. Adaptive routing routes the packets according to traffic parameters like if traffic congestion occurs then it sends the packets to the alternate path. Low channel load paths are used by adaptive routing. For ex-ample, if the source is(0,0) and destination is (2,1) if congestion has occurred on (1,0)'s east side, then it will move according to the shortest path here it will move the north side of(1,0) (Villanueva, J. C., 2012).

2.2. Issue with Routing Algorithms.

a) Deadlock: When 2 packets are waiting for resources, and at that time, they acquire to re-lease their resources.

b) Livelock: Livelock occurs if the destination does not receive Packet, and Packet is sent continuously random in direction. There is a particular time for packet living (TTL). Time to live describes that how much time the Packet is live on the network, and when livelock oc-curs, then Packet is circulated on the network, and little time later, its damage or dumped but does not reach the destination (Wilson, L., 2010, November).

c) Starvation: Starvation problem occurs when higher priority packets have acquired the re-sources all time, and lower priority packets wait in the queue, but it can't use these re-sources than starvation problem occurs .if reduce this problem than implement a balanced, fair routing (Hou, J., Han, Q., & Radetzki, M., 2019).

2.3 Why NoC is better than other Socs

Network-on-chip has some advantages which create a difference with another system on chips. NoC is better with crossbar and bus system transmission. The architecture of NoC is scalable network architecture. Point to point interconnection of links, switch functionality is used, verity in routing algorithms, topologies provide enhance performance as per efficiency, throughput, Latency, channel allocation manner, and comparison with the previous methodology of the chip (Melo, D. R., Zeferino, C. A., Dilillo, L., & Bezerra, E. A., 2019; Manna, K., & Mathew, J., 2020).

a) Scalability in terms of links(wires), channels, and switches manner because nodes are con-nected with many links, and NoC simultaneously can operate and forward different data packets on it. So it balanced the load on the channel.

b) The area is calculated according to topology, so it is verity on choosing topology on an area basis.

c) The power efficiency of SoC is complex as compare with NoC. NoC has an advantage that powered on only those nodes which are participating at per time from source to destination, and other switches are powered off. So the power saving option also includes with NoC.

d) It provides scalable bandwidth at the low area and power overhead, which is directly corre-lated with the number of nodes.

e) NoC follows particular topological architecture, so wiring designing for predictable speed, power calculation, reliability, area calculation, etc. because of their well-formed predefined architecture.

f) NoC knows how efficiently the use of multiplexer, decoder, wires, switches integration on communication flows on the same link and allow fault-tolerant on links for getting higher bandwidth.

3. Problem Statement and Objective

3.1. XY Routing

XY routing is dimensional ordering routing, and it works according to the X dimension Y di-mension. When Packet send from source to destination than (X and -X) and (Y and -Y) are available.

3.2. XY Routing in MESH

Mesh XY routing smoothly works because nodes are connected in the mesh as a pint to point the way. When Packet is sent to the destination, than firstly, it moves to the X direction of des-tination column than it turns to Y direction with the destination side. YX routing is just reverse to the XY routing.YX routing packet firstly moves Y direction of destination column than it moves X directions.

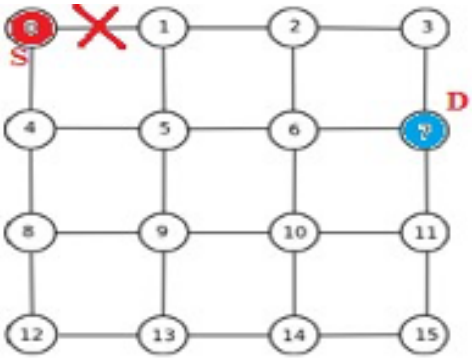

Fig. 3: Source tile is 0 and Packet not Sent Destination Tile 7

Here the example of 4*4 metric and source node is (0, 0), and the destination node is (1, 3) than if packets send to destination and link failure occurs between node 0 and node 1. The Packet cannot send to its destination because it cannot

move first in the Y direction. Packet dumped and cannot send to the destination (Frantz, A. P., Kastensmidt, F. L., Carro, L., & Cota, E., 2006).

In 4*4 Mesh, than source tile is tile 10, position (2, 2), and Packet send to the destination is tile number 13, position (3, 2), and link failure occurs between tile 10 and tile 14. According to the XY routing algorithm, the Packet is first to send to the X dimension than it will move the Y dimension.

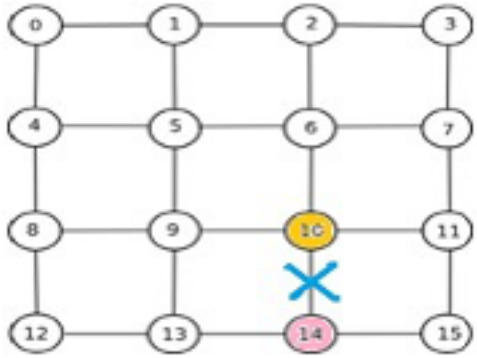

Fig. 4: Source tile is 0 and Packet Sent Destination Tile 7

So link failure between tiles 10 and 14 does not affect on packet transfer process because fail-ure is on the Y side. There are some problems with the XY routing algorithm. Traffic does not expend equally on the network because traffic increase load on generic tiles of the network, or we can say middle of the network .so it slows the speed on the middle area of the network, so the disadvantage of XY routing is the equalization of traffic.

- Un equalization of traffic
- Increase load
- Decrease throughput.

4. Implementation

4.1. Implementation of HPC Topology

When the topology is resized, the number of neighbors in the input and output ports for each node is a higher radix topology. Since in the case of NIRGAM, the number of neighbors in the input, the output ports are defined in “Constants.h”, therefore, different values are used to de-fine these NIRGAM parameters. All possible sizes of the HPC topology are shown in Table 1.

Table 1: Source Tile is 0 and Packet Sent Destination Tile 7

Size of Topology	No. of Neighbors	No. of input ports	No. of Output ports
2*2	3	4	4

2*4	4	5	5
4*4	5	6	6
8*4	6	7	7
8*8	7	8	8
8*16	8	9	9
16*16	9	10	10

HPC mesh topology divides into 4 subnetworks; switches are connected in the same sub net-work's switches, and on the other side, it also connected with end nodes and switches that be-long to the different subnetwork. In other words, internal and external links are connected to the network; the last nodes are connected to different sub-networks. As shown in Figure 5, the internal links are further divided into short links and the transverse links, and the external link is the same as the longer links? The long link is connected to different sub-networks, and the short link is connected to the same subnetwork switches (Kumar, A., Dadheech, P., Singh, V., Raja, L., & Poonia, R. C., 2019). The longer link consumes less energy and reaches the des-tination tiles faster.

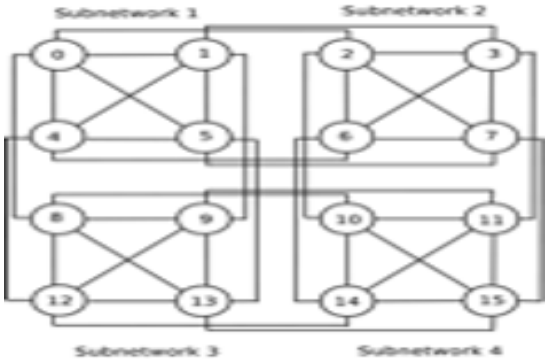

Fig. 5: HPC Mesh Topology

Because of the higher radix topology, the number of neighbors and input per output node changes as the size of the topology changes. Since there are neighboring numbers and inputs in the case of Nirgam, the output ports are defined in "Constants.h" so there is no view of chang-ing these parameters in Nirgam.

Hence instead of generalizing for any possible dimension, we have hardcoded HPC topology for 2*2, 2*4, 4*4, 8*4, 8*8, 8*16, and 16*16.

HPC mesh is the combination of 4 homogeneous sub-networks, and this topology manage channels and nodes dynamically. Channels are connected according to topological require-ments.It adjusts the overall conditions of the network and increases the best utilization of power in low traffic. HPC mesh performs operations on high and low traffic rates and gets results in maximum utilization of network

(Dadheech, P., Goyal, D., Srivastava, S., & Kumar, A., 2018).

Channels are divided into HPC mesh is 3 parts.

1. long link
2. short link
3. diagonal link

By using these links, connectivity appear in HPC mesh, or we can say HPC mesh is the con-nectivity of multiple types of links.

4.2. FTXY in MESH

Fault tolerance routing is based on XY routing, but it can manage link failure problem. It increases the probability of sending Packet from source to destination.

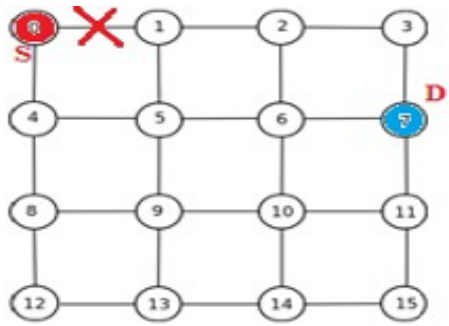

Fig. 6: Source tile is 0 and Packet Sent Destination Tile 7

4*4 metrics are used, and adapting routing is used for packet transmission. If the source is tile 0 and the destination is tile 7. Link failure occurs between tile 0 and tile 1. According to fault tol-erance rule in adaptive routing, there are multiple paths between sources to destination. The Packet is sent with the Y dimension side.

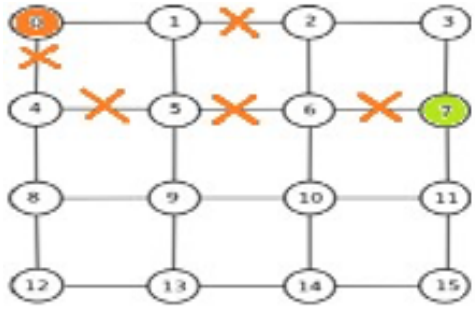

Fig. 6: Source tile is 0 and Packet Sent Destination Tile 7

In 4*4 metric source is tile 0 and sends packets to the destination is tile 11, and link failure oc-curs between tile0, tile4, tile4 tile5, tile6 tile7, tile1 tile2. Multiple link

faults in the network are decreased efficiency, but fault tolerance adaptive routing can manage in this situation.

Source routing maintains records of each node header and source, taking information to all nodes. When the problem comes up within the paths (Kumar, A., & Sinha, M., 2014; Kumar, A., & Sinha, M., 2019) and not getting diagnosed by routing protocol, then the problem will be test-ed by source routing (Kumar, A., Dadheech, P., Kumari, R., & Singh, V., 2019).

4.3. Implementation of XY Routing for HPC Topology

In this research, adaptive XY routing and normal XY routing for HPC mesh topology for par-ticular 4*4 combinations of nodes. XY routing follows its predefined path; according to that, Packet sends horizontally in X direction up to destination's column than in diagonal direction with (+45 or -45) degree than it moves vertical direction or Y direction.

*4.3.1. Algorithm for XY Routing for 4*4 HPC Topology*

Step1. Nodes are divided into four sets depending upon their x and y coordinate modulus with 2 directions on each set.

{[E, E], [O, O], [E, O], [O, E]}

Step 2. If (des_x - cur_x == 0 and des_y - cur_y == 0) return to Core.

Step 3. (cur_x % 2 and cur_y % 2). && (des_x % 2 and des_y % 2)

Step 4. Check the difference between corresponding x and y coordinates and take the link the provides the shortest path between the two nodes. In case more than one shortest path is possible, follow the priority order of link as follows.

1. Long link in x-dir.
2. Short link in x-dir.
3. Diagonal
4. Long link in y-dir.
5. Short link in y-dir.

Step 5. Following the link of highest priority reach to the next node.

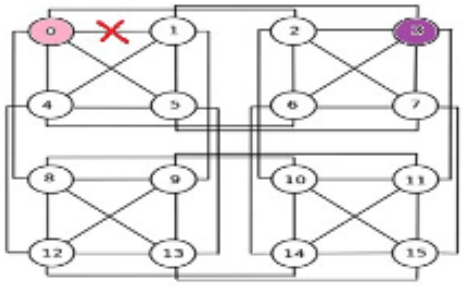

Fig. 8: Source tile is 0 and Packet not Sent Destination Tile 3

Step 6. This node now becomes the current node, and the whole process is repeated until the Packet reaches to the destination.

If the source is (0, 0) and sends a Packet to the destination (0, 3) if the link fails between tile 0 and tile1 than Packet does not send to the destination because it's a problem in XY routing that source X side value if match with destination's X side value than Packet cannot be sent to the destination.

4.4. Implementation of Fault Tolerance in HPC

We have also include Fault tolerance in the implementation part of HPC topology by imple-menting adaptive XY routing.

4.4.1. Fault-Tolerance Factor

It is a common problem when Packet is sent from source to destination, and link failure occurs than fault-tolerant factor sends Packet through alternative paths. It's a fast technique to send Packet and less probability of Packet dumped and damage (Dadheech, P., Kumar, A., Choudhary, C., et al., 2019). HPC mesh topology is worked with fault tolerance technique. Here HPC structure can supports 4 Sub Networks and able to tolerate 3 links failure. Fault tol-eration supports if single-point failure has occurred, then Packet is delivered the packets of nodes are not faulty. If we talk about a simple XY routing algorithm, the fault tolerance factor does not work with it. In NIRGAM, more than one number of faults may be injected between sources to destination in the network. The goal of fault-tolerance is to introduce maximum link injection probability. Existing routing algorithm used fault -tolerance property and made chang-es according to that NIRGAM is capable of testing various routing and topological changes and shows the result as per tile basis, so it is an easy way to calculate multiple process accord-ing to it [37, 38].

4.4.2. Introducing Fault in Nirgam

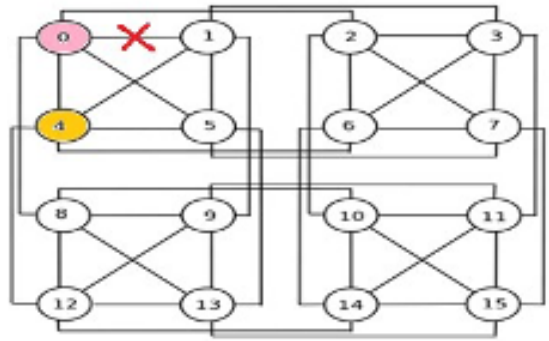

Fig. 9: Source tile is 0 and Packet Sent Destination Tile 4

For introducing faults in NIRGAM, one needs to define a link that fails in Nirgam. config In “Nirgam.config” type” LINKFAIL” without the quotes followed by the number of links to be failed and specification of every link that needs to be failed. e.g LINKFAIL 4 0 1 0 2 1 3 2 7. Here if source is (0, 0) tile 0 and destination is (1, 0) tile 4. Adaptive routing is used to send a Packet from source to destination. LINKFAIL 1 0 1.

If the source is tile 0 and Packet is sent to destination tile 5. All the links of X direction of tile 0 fails than only one way to send Packet to the destination is diagonal mesh. The Packet is sent to the destination.

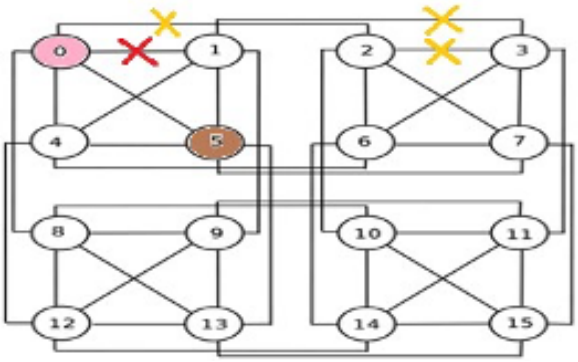

Fig. 9: Source tile is 0 and Packet Sent Destination Tile 4

*4.4.3. Implementation of Adaptive XY Routing for 4*4 HPC Topology*

HPC mesh chooses another path if a link failure occurs during sending Packet from source to destination. There are 3 ways to send Packet. In the first step, we evaluate the difference be-tween source to destination tile of X and Y dimensions; it will return core if it is equal to zero, else it evaluates link preference followed by the source to reach destinations. In every transfer, a long link is given a higher preference. If the difference between source to destination tile of X and Y dimension is greater than 1 than it will follow the long link, else it will follow a diagonal path if the difference is equal to one, for either X dimension or Y dimension, it will follow short link path if the difference of source and the destination node is equal to zero. In the sec-ond step, it calculates the status of current tile (T1) links that are connected with neighbor tiles (TN) in the dimension of E, W, N, S, Dg. In third step ,it determine faulty links among TI of TN in fE , W, N, S, Dg dimension , if TI = F[i][0] and TN = F[i][1] ,ND = faulty link. On the other side if TI = F[i][1]and TN = F[i][0] , then faulty link = ND. In both cases, it will return ND, and it takes another path to reach the destination [40]. It will last till three-link failure.

*Algorithm for Adaptive Fault Tolerance XY Routing in 4*4 HPC Topology*

Prerequisites Algorithm

L#: Faulty link

lN: North link failed
lW : West link failed
lE : East link failed
lS : South link failed
lD : Diagonal link failed
XDYD-S: Y coordinate difference between destination and source. -S:
X coordinate difference between destination and source.
F[i] [j]: i is the tile number where the fault occurs,F is the fault matrix, and j contains the status of the current node and neighboring node.
Step1:- Divide X and Y coordinates into four sets:
S1= {E, E} S2= {O, O}
S3= {E, O} S4= {O, E}
Step2:- if (Dx-C= 0 ^ Dy-Cy= 0) return Core
Step3:- Calculate Cx % 2 Cy % 2. And Dx % 2 and Dy % 2.
Step4: T = Ne (id_E V id_S V id_N V id_W V id_D) of C
Step5:- Compare (id_E ^ id_S ^ id_N ^ id_W ^ id_D) Ne of T with the F[n] [m] then return ND <=T Step 6:- FOR I=0 to Fn
If (id = F[i][0]) then if (Ne = F[i][1])
Then Dir = ND end if end if
Else if (id = F[i][1]) then if (Ne = F[i][0]) then
Dir=ND end if endif
END FOR
Step7:-return Dir <=Dir ≠ ND
Step8:- Dir of N = C

5. Results (Comparison between HPC and Mesh Topology)

5.1. Power Calculation Factor in HPC

Comparison of the output simulation results for the 4*4 mesh topology and 4*4 HPC topol-ogy is shown below:

Table 2: Power Consumption in MESH and HPC Topology for Source 0 and Destination 15

Tile Number	HPC Power estimation (in watts)	MESH Power Estimation (in watts)
0	0.0272696	0.0301501
1	0.0200985	0.030055
2	0.0272696	0.030055
3	0.0200985	0.030055
4	0.0200985	0.0215778
5	0.0200985	0.0215778
6	0.0200985	0.0215778

7	0.0200985	0.0298846
8	0.0200985	0.0215778
9	0.0200985	0.0215778
10	0.0272696	0.0215778
11	0.0200985	0.0293182
12	0.0200985	0.0215778
13	0.0200985	0.0215778
14	0.0200985	0.0215778
15	0.0272696	0.0280659
Total	0.350261	0.401784

Table. 3: Power Consumption in MESH and HPC Topology for Source 0 Destination 10

Tile Number	HPC Power estimation (in watts)	MESH Power estimation(in watts)
1.	0.0339716	0.0301501
2.	0.0200985	0.030055
3.	0.0336314	0.030055
4.	0.0200985	0.0215778
5.	0.0200985	0.0215778
6.	0.0200985	0.0215778
7.	0.0200985	0.0300555
8.	0.0200985	0.0215778
9.	0.0200985	0.0215778
10.	0.0200985	0.0215778
11.	0.0331623	0.0298591
12.	0.0200985	0.0215778
13.	0.0200985	0.0215778
14.	0.0200985	0.0215778
15.	0.0200985	0.0215778
16.	0.0200985	0.0215778
Total Network Power (In Watt)	0.362046	0.38753

Thus we can see that network power consumption is less for HPC topology than mesh topolo-gy.

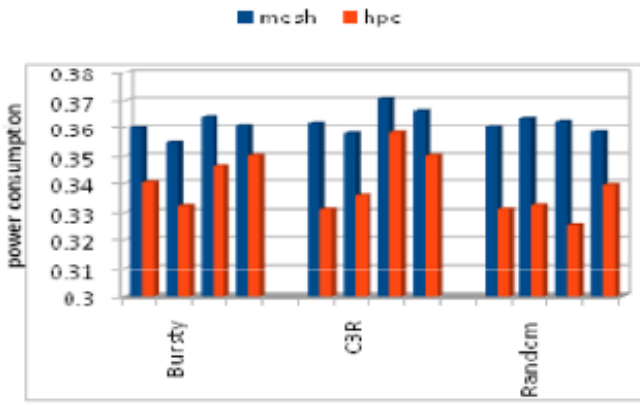

Fig. 11: Electricity Utilization in Mesh and HPC Topology for Different Traffic

In Figure 11, we are considering the simulation result mesh and HPC for four different tiles and different traffic; the result showed that the HPC mesh consumes less energy compared to the 2D mesh.

Power Calculation: Power calculation firstly calculates links power and nodes power in a dif-ferent file.

Link Power Calculation: Link power is calculated in NIRGAM 2-0. Link power is a combi-nation of coupling capacitance, buffer, etc. SIM_link file supports link power calculation. NIRGAM simulator calculates each tile's power individually than calculate overall results. Dif-ferent topologies have different link power results. The link power factor in Mesh topology is various links are connected with nodes. Mesh topology, HPC mesh topology, and torus topol-ogy link power factor are below. This power calculation is used to finding each tile's power in-dividually than this power collaborates with nodes power.

Router Power Calculation: Router power calculates in ORION -2.0's router power file. It is calculating the node's power individually. Each node or tile generates its own power for ON and OFF. Router power is used with link power and given the results of the average ratio.

5.2. Area Calculation

The area of the network in the chip is very important for picking the right topology. Each topol-ogy has its own area, which is a combination of different functions. As a power factor, the re-gion is also the calculation of the node region and the link field. If both areas are counted rather than the main field will be discovered. The router region is a combination of 4 functions in the router configuration file.

Four functions are:

$$Ra = Ba + Ca + Vc + Sw$$

Here Ra is the router area; Ba is a buffer area, Ca is crossbar area, Vc is a virtual channel, Sw is switch allocator. The topology area is equal to the router area and router link area. Here router link area specifies that link which is connected to the router. The router region first calculates the value of this function rather than the overall result. The HPC mesh works on multiple nodes and switches sub-networks. HPC is introducing a new connection pattern for the topology nodes and this pattern compared to other topologies.

5.3 Comparison in Terms of Path Diversity

5.3.1 Path Diversity

The maximum-minimum number of paths between two pairs of nodes. Increasing the diversity of the paths increases the stability of the network by balancing the load on the channels and al-lowing the system to withstand the fault channels and nodes [24].

Formula Derived for Path Diversity for HPC and Mesh Topology: Here shown path di-versity as Pd, Total Hop Count as Th, No. of Vertical Hops as Vi, No. of Horizontal Hops Hi

For HPC: Pd= (n-1)!

For MESH: PD=Th!/HiVi!

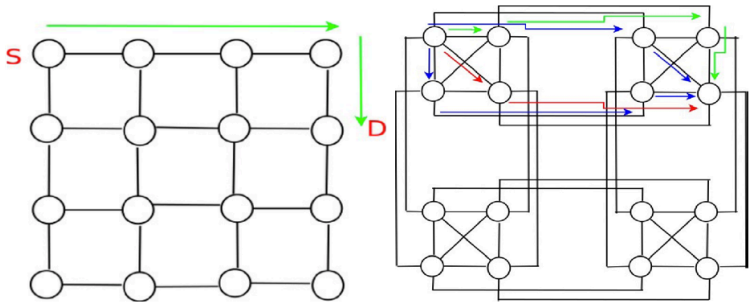

Fig. 12: (a). Power Consumption in MESH (b). HPC Topology for Different Traffic

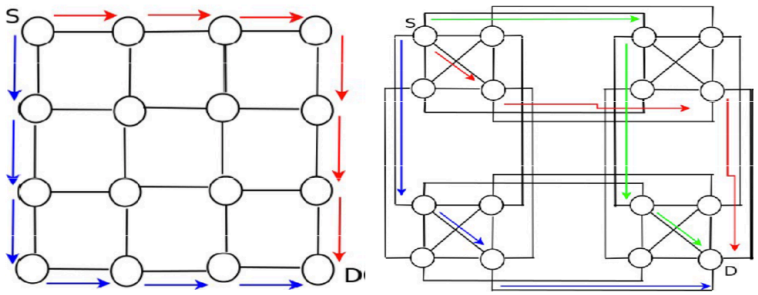

Fig. 13: (a). Power Consumption in Mesh D (b). HPC Topology for different traffic

Table 4: Path diversity for Long Distance (a) MESH (b) HPC

Source and Destination	Mesh	HPC
When source tile 0 and destination tile 7 (4 × 4)	4!/(3!*1!)	2!
When source tile 0 and destination 15 (4 × 4)	5!/(4!*2!)	3!
When source tile 0 and destination is 14 (8 × 4)	7!/(6!*1!)	3!

In fig 12 and 13 shown path diversity for long and short distance fig 12(a) shows that 4 hops on the way of destination but 12(b) shows that 2 hops are on the way of destination. As fig 13, when the destination is too far than fig13 (a) when source tile0 and destination is tile 15, 6 hops on the way of destination on the other side in HPC fig 13(b) 3 hops are between sources to destination. We can say that the path diversity of HPC less than 2D mesh.

5.4. Comparison in Terms of Latency

5.4.1. Latency

If a packet needs time to cross the network, the Packet's tail will leave the output port when the Packet's head reaches the input port. In Figure 13, we also consider the simulation results for mesh and HPC for four different tiles and different traffic; the result showed that there is a lower HPC latency than 2D mesh.

5.4.2. Simulation Result for Latency for Mesh and HPC

Table 5: Simulation Result for Latency for Mesh and HPC

Source -0 destination-10 Source -1 destination -15	HPC	MESH
Overall average Latency per channel (in clock cycles per flit)	0.809524	3.1627
Overall average Latency per channel(in clock cycles per Packet)	2.42857	9.71951
Overall average Latency (in clock cycles per flit)	9.5	48.8421
Total flits generated	48	48
Total Flits received	48	38
Source -0 destination-10	HPC	MESH
Overall average Latency per channel (in clock cycles per flit)	0.75	2.98319

Overall average Latency per channel(in clock cycles per Packet)	2.31818	9.10256
Overall average Latency (in clock cycles per flit)	8.02326	43.1304
Total flits generated	24	24
Total Flits received	24	23

Here we can see that Latency is less for HPC Topology than Mesh Topology.

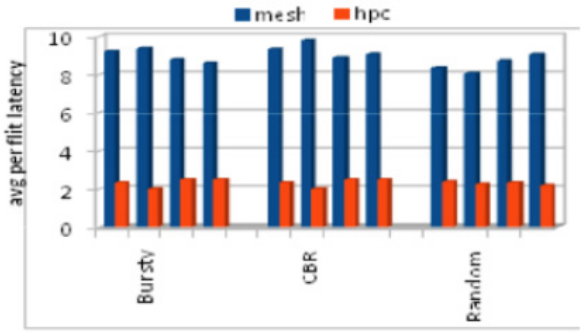

Fig. 14: Average per Flit Latency of HPC Topology and Mesh for Different Traffic

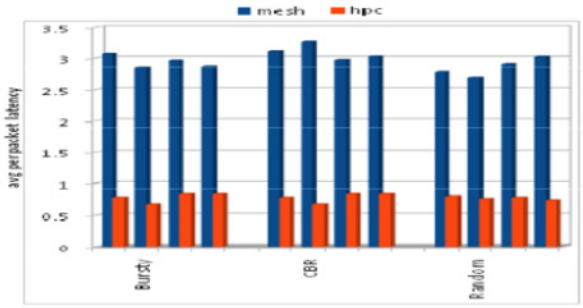

Fig. 15: Average per Packet Latency of Mesh and HPC Topology for Different Traffic

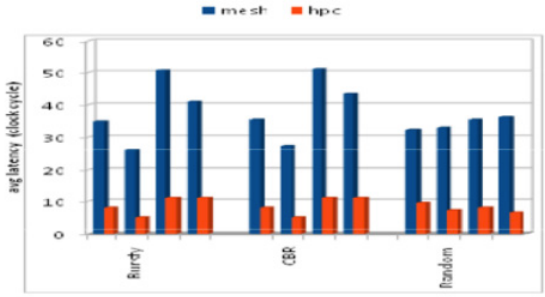

Fig. 16: Overall Average Latency of HPC and Mesh Topology.

5.5. Comparison in Terms of Throughput

Result of simulation for throughput for HPC and mesh topology are given below. HPC has more throughput than mesh topology.

Table 6: Throughput for Single Source and Destination in MESH V/s HPC

Tile-id Port	Throughput MESH (in Gbps)	Source-0 and Dest-10	Tile-id Port	Throughput in HPC (in Gbps)
0 East	15.2381		0 North	20
2 South	14.7692		2 West	20.4255
6 South	13.1429		10 Core	20.4255

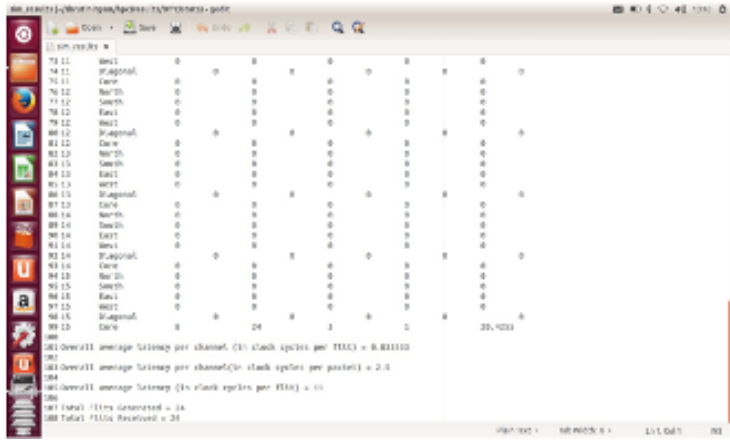

Fig. 17: Latency results in HPC where Source Node 0 and Destination Node 10

5.5.1. Multiple Source and Destination

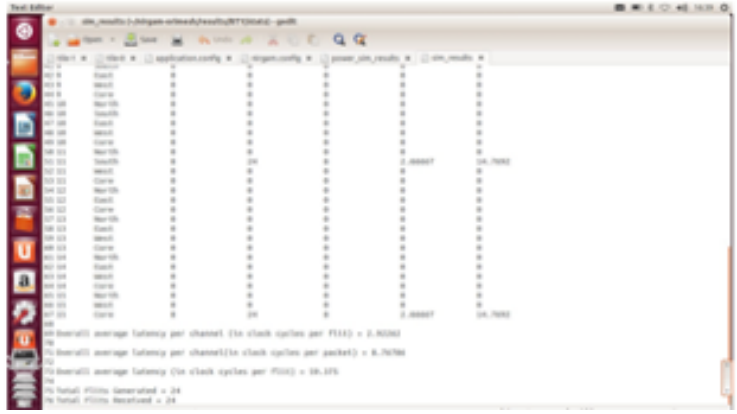

Fig. 18: Throughput Results of HPC and Mesh Topology

Table 7: Comparison in terms of Throughput

Tile-id	Port	Throughput MESH (in Gbps)	Source-0 Dest-10 and Source-1 Dest-15	Tile-id	Port	Throughput in HPC (in Gbps)
0	East	14.7692		0	North	20
1	East	23.4146		1	North	20
2	East	14.1176		2	West	20.4255
3	South	13.913		3	East	20.4255
6	South	13.1429		10	Core	20.4255
7	South	13.913		11	South	20.4255
10	Core	13.3333		15	Core	20.4255
11	South	14.6667				
15	Core	15.2				

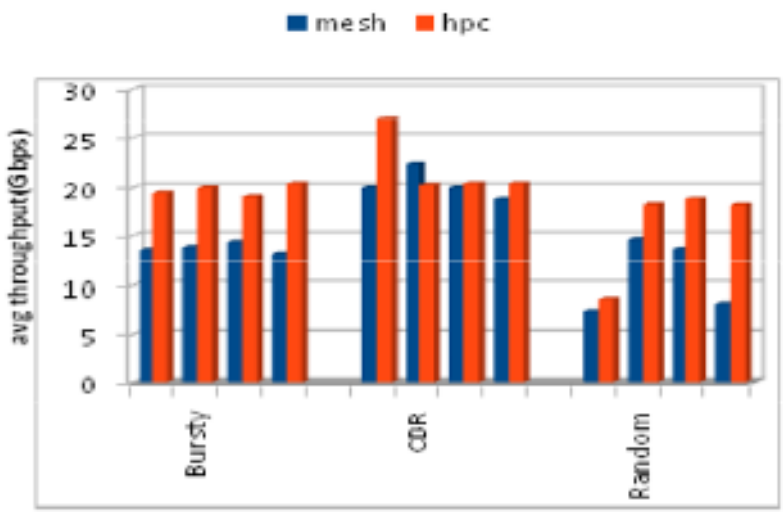

Fig. 19: Throughput Results of HPC and Mesh Topology

This figure clearly shows that the Performance of Throughput in HPC and Mesh Topology and Result shows better performance in all three environments Bursty, CDR, and Random.

6. Conclusion

The main difference is that the last node is connected to different sub-networks; therefore, there is more scope for power saving and performance optimization, which

dynamically achieves a better trade-off as we see by simulations for the performance of HPC and mesh topologies for various factors such as path variability, Latency, network power consumption, throughput and error tolerance. Fault to-lens is the ability to perform a network in the presence of one or more faults. If the packet link fails to follow a path due to failure, it selects the alternate path (follow-ing XY routing) for the source and destination pair. HPC mesh topology 2 is capable of with-standing failures as there are four parallel networks. Whereas in the case of mesh topology, it is only 1 link failure until the size of the H topology increases as the HPC's fault capacity increas-es. Where the mesh has the fault tolerance remains the same, it concludes that the HPC topology is better than the mesh topology.

7. References

Benini, L., De Micheli, G., & Ye, T. T. (2006). Networks ***on chips*** (Vol. 1). Burlington: Morgan Kaufmann.

Bjerregaard, T., & Mahadevan, S. (2006). A survey of research and practices of network-on-chip. *ACM Computing Surveys (CSUR), 38*(1), 1-es.

Bogdan, P., Dumitraş, T., & Marculescu, R. (2007). Stochastic communication: A new paradigm for fault-tolerant networks-on-chip*. VLSI design, 2007.*

Bouguettaya, A., Kimour, M. T., & Toumi, S. (2014, January). A New Clustering Based Routing Algorithm for NoC. In *The Proceedings of the International Conference on Machine Learning, Electrical and Mechanical Engineering* (pp. 9-13).

Camacho, J., & Flich, J. (2011, October). HPC-mesh: A homogeneous parallel concentrated mesh for fault-tolerance and energy savings. In *2011 ACM/IEEE Seventh Symposium on Architectures for Networking and Communications Systems* (pp. 69-80). IEEE.

Chen, C. L., & Chiu, G. M. (2001). A fault-tolerant routing scheme for meshes with nonconvex faults. *IEEE Transactions on Parallel and Distributed Systems, 12*(5), 467-475.

Chen, C., Lu, Y., & Cotofana, S. D. (2012, May). A novel flit serialization strategy to utilize partially faulty links in networks-on-chip. In *2012 IEEE/ACM Sixth International Symposium on Networks-on-Chip* (pp. 124-131). IEEE.

Dadheech, P., Goyal, D., Srivastava, S., & Kumar, A. (2018). A scalable data processing using Hadoop & MapReduce for big data. *J. Adv. Res. Dyn. Control. Syst, 10,* 2099-2109.

Dadheech, P., Kumar, A., Choudhary, C., et al. (2019). An enhanced 4-way technique using cookies for robust authentication process in wireless network. *Journal of Statistics and Management Systems, 22*(4), 773-782.

Dally, W. J. (1991). Express cubes: Improving the performance ofk-ary n-cube interconnection networks. *IEEE Transactions on Computers, 40*(9), 1016-1023.

Dally, W. J., & Seitz, C. L. (1988). Deadlock-free message routing in multiprocessor interconnection networks. *IEEE Transactions on Computers, 36*(5), 547-553.

Dally, W. J., & Towles, B. P. (2004). *Principles and practices of interconnection net-*

works. Elsevier. Morgan Kauffman.

Daneshtalab, M. (2011). Exploring Adaptive Implementation of On-Chip Networks.

de Melo, D. R., Zeferino, C. A., Dilillo, L., & Bezerra, E. A. (2019, March). Analyzing the Error Propagation in a Parameterizable Network-on-Chip Router. In *2019 IEEE Latin American Test Symposium (LATS)* (pp. 1-6). IEEE.

Duato, J., Yalamanchili, S., & Ni, L. (2003). *Interconnection networks.* Morgan Kaufmann.

El-Moursy, M. A., & Ismail, M. (2009, May). High throughput architecture for high performance NoC. In *2009 IEEE International Symposium on Circuits and Systems* (pp. 2241-2244). IEEE.

Frantz, A. P., Kastensmidt, F. L., Carro, L., & Cota, E. (2006, October). Dependable network-on-chip router able to simultaneously tolerate soft errors and crosstalk. In *2006 IEEE International Test Conference* (pp. 1-9). IEEE

Grecu, C., Ivanov, A., Saleh, R., & Pande, P. P. (2006, October). NoC interconnect yield improvement using crosspoint redundancy. In *2006 21st IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems* (pp. 457-465). IEEE.

Harbin, J., & Indrusiak, L. S. (2013, July). Fast transaction-level dynamic power consumption modelling in priority preemptive wormhole switching networks on chip. In *2013 International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS)* (pp. 172-179). IEEE.

Hou, J., Han, Q., & Radetzki, M. (2019, October). A Machine Learning Enabled Long-Term Performance Evaluation Framework for NoCs. In *2019 IEEE 13th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC)* (pp. 164-171). IEEE.

Joshi, A., Batten, C., Stojanović, V., & Asanović, K. (2008, September). Building manycore processor-to-DRAM networks using monolithic silicon photonics. In *High Performance Embedded Computing (HPEC) Workshop.*

Kumar, A., & Sinha, M. (2014, March). Overview on vehicular ad hoc network and its security issues. In *2014 International conference on computing for sustainable global development (INDIACom)* (pp. 792-797). IEEE.

Kumar, A., & Sinha, M. (2019). Design and development of new framework for detection and mitigation of wormhole and black hole attacks in VANET. *Journal of Statistics and Management Systems, 22*(4), 753-761.

Kumar, A., Dadheech, P., Kumari, R., & Singh, V. (2019). An enhanced energy efficient routing protocol for VANET using special cross over in genetic algorithm. *Journal of Statistics and Management Systems, 22*(7), 1349-1364.

Kumar, A., Dadheech, P., Singh, V., Poonia, R. C., & Raja, L. (2019). An improved quantum key distribution protocol for verification. *Journal of Discrete Mathematical Sciences and Cryptography, 22*(4), 491-498.

Kumar, A., Dadheech, P., Singh, V., Raja, L., & Poonia, R. C. (2019). An enhanced quantum key distribution protocol for security authentication. *Journal of Discrete Mathe-*

matical Sciences and Cryptography, 22(4), 499-507.

Kumar, A., Goyal, D., & Dadheech, P. (2018). A novel framework for performance optimization of routing protocol in VANET network. *J. Adv. Res. Dyn. Control. Syst, 10,* 2110-2121.

Lehtonen, T., Liljeberg, P., & Plosila, J. (2007). Online reconfigurable self-timed links for fault tolerant NoC. *VLSI design, 2007.*

Lehtonen, T., Wolpert, D., Liljeberg, P., Plosila, J., & Ampadu, P. (2009). Self-adaptive system for addressing permanent errors in on-chip interconnects. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 18*(4), 527-540.

Lit, A., Sahari, S. K., Spawi, R., et al. (2013). Power Optimization for Mesh Network-on-Chip Architecture: Multilevel Network Partitioning Approach. *Proceedings of the 6th International Engineering Conference (ENCON 2013)*

Manna, K., & Mathew, J. (2020). Alternative Approaches. In *Design and Test Strategies for 2D/3D Integration for NoC-based Multicore Architectures* (pp. 13-31). Springer, Cham.

Melo, D. R., Zeferino, C. A., Dilillo, L., & Bezerra, E. A. (2019). *Maximizing the Inner Resilience of a Network-on-Chip through Router Controllers Design. Sensors, 19*(24), 5416.

Radetzki, M., Feng, C., Zhao, X., & Jantsch, A. (2013). Methods for fault tolerance in networks-on-chip. *ACM Computing Surveys (CSUR), 46*(1), 1-38.

Sharma, A., Tailor, M., Bhargava, L., & Gaur, M. S. (2018, June). 3D LBDR: Logic-Based Distributed Routing for 3D NoC. In *International Symposium on VLSI Design and Test* (pp. 473-482). Springer, Singapore.

Villanueva, J. C. (2012). *High Performance and Power Efficient On-Chip Network Designs through Multiple Injection Ports* (Doctoral dissertation, Universitat Politècnica de València).

Wilson, L. (2010, November). Managing Vendor Relations: A Case Study of Two HPC Network Issues. In *the 24th Large Installation System Administration Conference,* San Jose California

Yu, Q., & Ampadu, P. (2011). Dual-layer adaptive error control for network-on-chip links. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 20*(7), 1304-1317.

Submitted 19.03.2020
Accepted 25.05.2020