International Journal of Automation and Computing 9(4), August 2012, 403-413
DOI: 10.1007/s11633-012-0661-1

Performance Evaluation of Mesh-based NoCs: Implementation of a New Architecture and Routing Algorithm

Sudhanshu Choudhary Shafi Qureshi
Indian Institute of Technology, Kanpur 208016, India

Abstract: This paper presents the result of experiments conducted in mesh networks on different routing algorithms, traffic generation schemes and switching schemes. A new network on chip (NoC) topology based on partial interconnection of mesh network is proposed and a routing algorithm supporting the proposed architecture is developed. The proposed architecture is similar to standard mesh networks, where four extra bidirectional channels are added which remove the congestion and hotspots compared to standard mesh networks with fewer channels. Significant improvement in delay (60 % reduction) and throughput (60 % increase) was observed using the proposed network and routing when compared with the ideal mesh networks. An increase in number of channels makes the switches expensive and could increase the area and power consumption. However, the proposed network can be useful in high speed applications with some compromise on area and power.

Keywords: Interconnects, network on chip (NoC), routing, traffic, switching.

1 Introduction

On chip interconnection architecture is commonly based on dedicated wires or shared busses. As reported in [1–5], it will not be possible for the dedicated wires and shared busses to fulfill the communication requirements of future integrated circuits (ICs). As the system complexity increases the number of dedicated wires grows which means they are effective only for small core systems. Also there are issues of poor reusability and flexibility with dedicated wires. A shared bus can be more scalable and reusable compared to dedicated wires but it cannot support the communication parallelism and bandwidth requirements of multi core systems[6,7].

Network on chips (NoCs) have emerged as a solution to the existing interconnection architecture constraints[8−10] for building scalable multi-core/multiprocessors systems on chip (SoC) due to the following characteristics: 1) energy efficiency and reliability[3], 2) scalability of bandwidth when compared to traditional bus architectures, 3) reusability, and 4) distributed routing decisions[4,5]. NoC is an on-chip interconnection network[4] composed by cores (IP blocks) connected to switches (routers) which are in turn connected among themselves by communication channels. Two main resources that compose NoCs are buffers and channels. To increase the resources allocation for each packet, a physical channel is multiplexed into a number of virtual channels[4, 5]. Each virtual channel has one or more buffers which provide multiple buffers for each physical channel. The performance of NoC with virtual channels can be evaluated and the important parameters can be fixed by the designers for specific applications[10].

The main steps of an NoC design are architecture specification, traffic modeling, and performance evaluation. In this paper, a new NoC architecture (partially connected mesh topology) and a suitable routing algorithm for the new architecture are proposed. We start with the description of architecture requirements of NoCs in Section 2. In Section 3, an 8×8 mesh network is evaluated on various routing algorithms, traffic and switching schemes. We use Noxim simulator[11] in this section for performance evaluation to identify the best NoC architecture supporting low latency and high throughput. Here we used few important existing schemes in our evaluation. In Section 4, partially connected mesh network topology and a routing scheme for this topology is proposed. This proposed architecture and routing algorithm is compared with the best configured mesh network architecture obtained in Section 3. We used Nirgam simulator[12] to build the proposed architecture and routing scheme, the simulator was modified to support the new topology and routing scheme.

2 Architechture specification

Important parameters that define NoC architecture are 1) topology, 2) switching and control logic, and 3) routing algorithm. Topology defines the way routers are connected. Mesh and torus networks are shown in Fig. 1, where each box represents a tile in the network. NoC is shown as a two dimensional interconnection of tiles (or nodes). Generally each tile has an IP core connected to routers/switch through a bidirectional channel responsible for communication between the switch and its local IP core. Also, each tile is connected to neighboring tiles through four bidirectional channels: East, West, North and South. In a standard mesh network, each corner tile has two neighbors, border tile has three neighbors and others have four neighbor tiles. Each tile can be identified by a unique integer tile ID and x-y coordinates, as shown in Fig. 1.

Manuscript received October 29, 2010; revised October 9, 2011

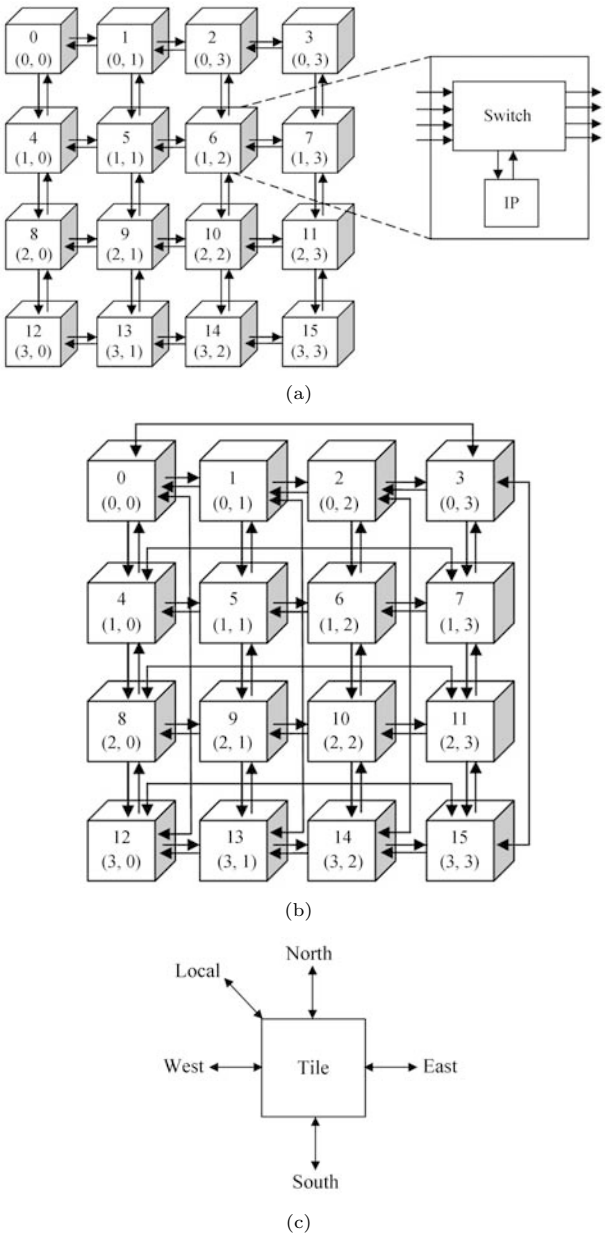


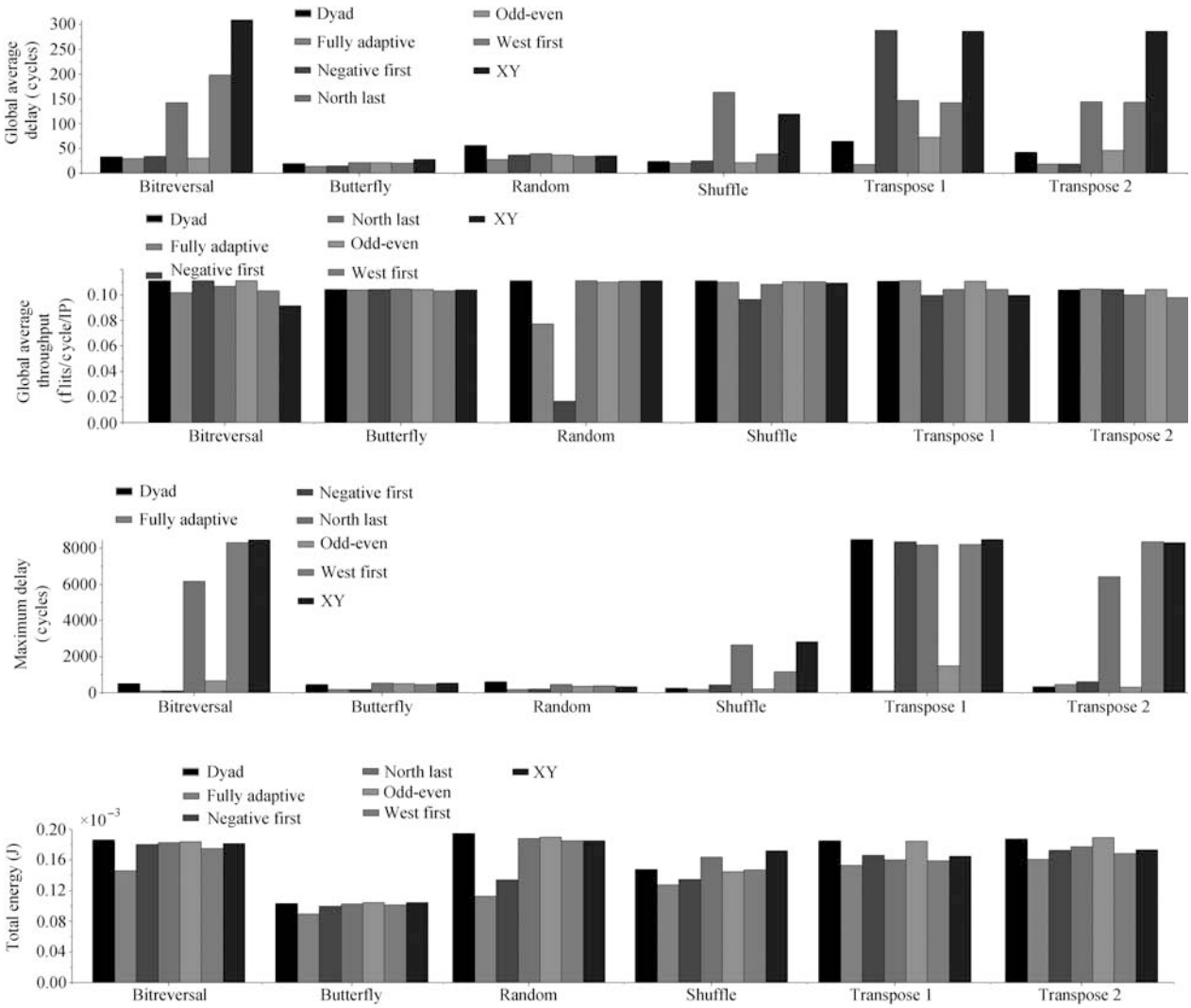
Fig. 1 Network architectures: (a) Mesh network; (b) Torus network; (c) Single tile of a mesh network with bidirectional channels

How packets move through the routers is decided by switching mode and controller. All the tiles have an input channel controller, output channel controller and a virtual channel allocator (VCA) for each neighbor. VCA services the request for virtual channel allocation from all input channel controllers. Each tile has one controller which enables router to service all requests from input channel controllers. Under heavy load conditions, dynamic routing can be applied while static routing architecture can be applied to save area[13]. Inter-router packet routing schemes like store-and-forward[14], virtual cut-through and wormhole routing can be applied[15]. In store-and-forward routing, a received packet is immediately stored and forwarded by the router resulting in lower network contentions but with increased latency and area occupied by the buffers used during storage. Lower latency is possible with virtual cut-through routing where each packet is sent only if the next router can store it but buffer requirement can be a problem. Wormhole routing[16,17] was designed to overcome these difficulties while offering similar network latency[18]. In wormhole routing, each packet is divided into a number of flits (flow control digits) for transmission. All flits use the same route which is determined by the first flit called header flit. If the header flit finds a blocked channel in its path, it waits until that channel is free. The trailing flits are also blocked in the flit buffers of the channel acquired by the packet. In other words, the channel is reserved for the whole packet. The channel is released when the last or tail flit has been transmitted on the channel. This technique requires a lower buffer memory in comparison to virtual cut-through and store-and-forward routing but is subject to head-of-line problem, since a physical channel may get blocked and hence cannot be used for other communications.

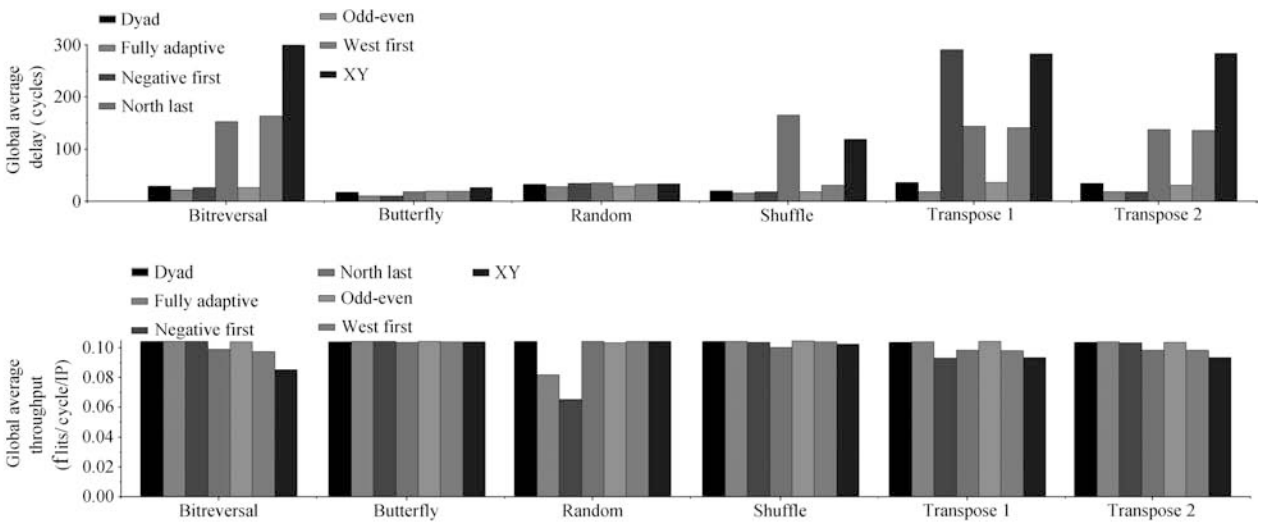
Routing algorithm defines the path taken by the packets from source to destination tile. Algorithms that are in current use are dyad, fully adaptive, negative first, north last, odd even, west first, and XY[19−31]. XY routing is a very common routing algorithm where a packet is routed in x-direction or y-direction, depending on the x-y coordinates of the source and destination tile. Fully adaptive routing is the best known algorithm which overcomes the high latency and dead-lock problem associated with XY routing, here a path is a function of network traffic. We will discuss routing algorithms in Section 4, where a new routing algorithm is proposed for the new NoC topology. Traffic generation scheme defines the structure of data transmission from source to destination tile. Traffic modeling is governed by three parameters: packet spatial distribution, packet injection rate and packet size[13]. The injection rate is always a fraction of the maximum channel bandwidth. From the injection rates defined by the user, globally or for each IP, a traffic generator can compute the interval between packets. Traffic patterns that are in current use are constant bit rate, bursty, buffer level, bit reversal, shuffle, butterfly, matrix transpose, random and complement[20−30]. In the next section, we use Noxim simulator[11] to evaluate mesh networks.

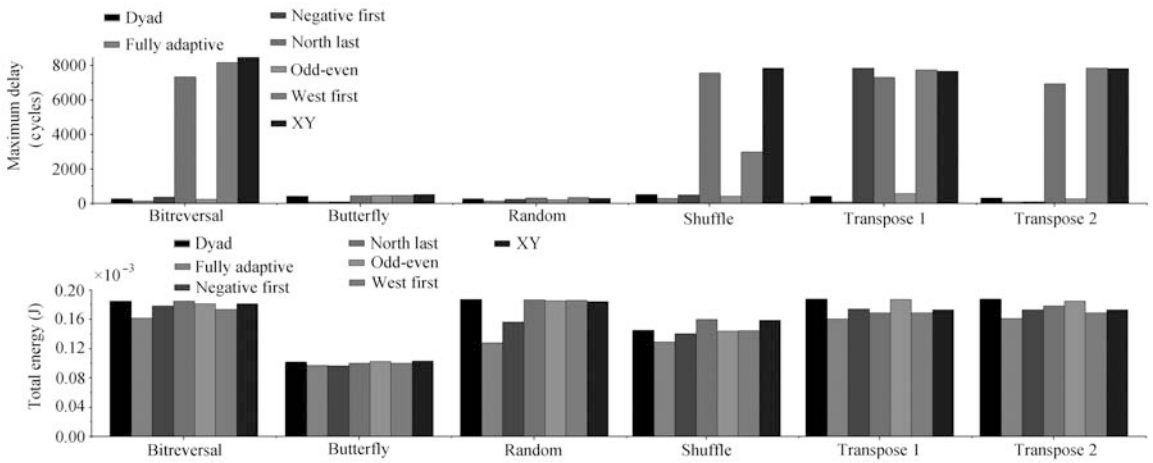
3 Architecture performance evaluation

We evaluate an 8×8 mesh network in this section on different traffic, routing algorithms and switching schemes. Traffic generation schemes that are considered here are bitreversal, butterfly, random, shuffle, transpose 1 and transpose 2[20−30]. Routing algorithms that are considered are dyad, fully adaptive, negative first, north last, odd even, west first, and XY[20−30]. The performance is evaluated on three different switching schemes: bufferlevel, neighbors on path (NOP) and random switching. A complete set of tests involving above mentioned traffic scenarios and routing strategies was performed using Noxim[11], an open source SystemC simulator of mesh-based NoC. Fig. 2 is obtained by the test performed on an 8×8 mesh network similar to Fig. 1 (a). The results in Fig. 2 are useful for comparing the performance with respect to global average delay (in cycles), global average throughput (in flits per cycle per IP), maximum delay (in cycles) and total energy consumption (in joules).

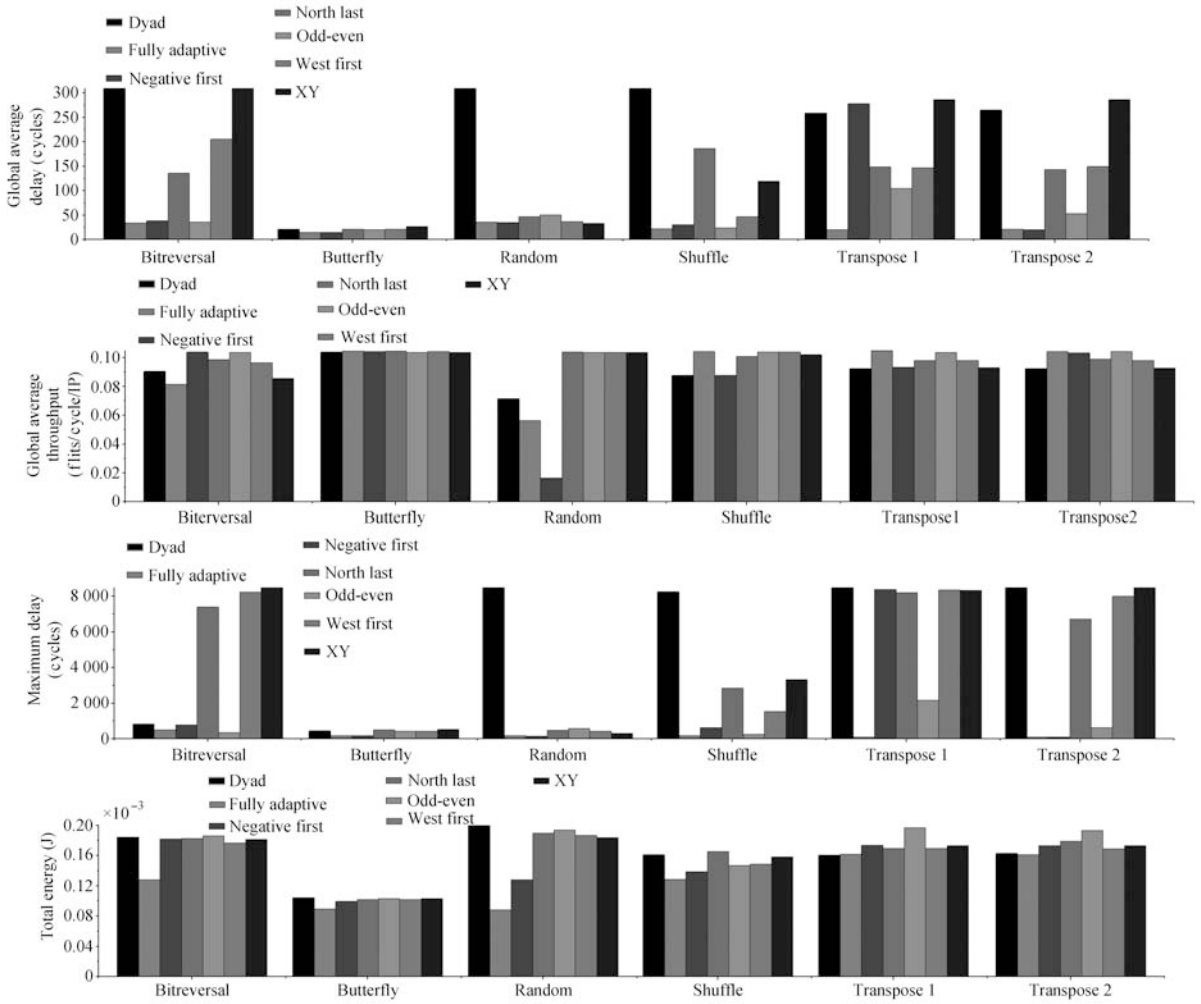


(a) Delay, throughput and energy measurements on buffer-level switching





(b) Delay, throughput and energy measurements on neighbors-on-path switching



(c) Delay, throughput and energy measurements on random switching

Fig. 2 Simulation results showing global average delay, throughput, maximum delay and energy consumption for an 8×8 mesh network at: (a) Buffer-level switching; (b) Neighbors-on-path switching; (c) Random switching

Delay is defined as the time (in clock cycles) that elapses between the occurrence of a header flit injection into the network at the source node and the occurrence of a tail flit reception at the destination node. Total flits received in a clock cycle define the throughput. The rate at which packets are injected into the network is packet injection rate (*pir*), which is varied from 0.01 to 0.015 to obtain the performance at different *pir*. A *pir* of 0.1 (packets/cycle/node) means that each node sends 0.1 packets every clock cycle or that each node sends a packet every 10 clock cycles. Poisson packet injection distribution[11] is used in this paper. The first in first out (FIFO) buffers have a capacity of four flits. Each simulation was run for 10 000 cycles with a warm-up period of 2 000 cycles after which data is collected. To guarantee the accuracy of results, the simulation at various *pir* was repeated for twenty times.

3.1 Experimental results and discussion

For each traffic scenario, switching scheme and routing algorithm, we obtain the average packet delay and throughput at different *pir*. The maximum delay and total energy consumption was also obtained for various *pir*. Fig. 2 presents the simulation results at *pir* of 0.014, results for different *pir* were obtained separately and are examined in this section.

It was seen that in buffer level switching scheme with bitreversal traffic scenario, throughput of fully adaptive routing decreases rapidly when *pir* goes high and it falls below XY routing′s throughput at a *pir* of 0.014. Also energy consumption of XY routing reduces with increase in *pir*, it falls below west first routing′s energy consumption at a *pir* of 0.014. Energy consumption of fully adaptive routing also decreases with increase in *pir*. When traffic scenario was changed to shuffle, the throughput of negative first routing decreases rapidly for *pir* above 0.011. Also on this traffic, the energy consumption of fully adaptive and negative first decreases at high *pir*

In random switching scheme with shuffle traffic, fully adaptive′s and negative first′s throughput decrease rapidly at high *pir*. For random traffic negative first′s maximum delay goes below fully adaptive′s max delay at a *pir* of 0.014. For transpose 1 and transpose 2 traffic, dyad′s energy consumption falls below fully adaptive′s at high *pir*.

From Fig. 2 and other similar simulation results it was found that fully adaptive routing has the smallest delay with any combination of traffic scenarios and selection schemes. However, for random selection scheme the delay was comparable to odd-even and negative first routing schemes as shown in Fig. 3. A comparison of delays in case of fully adaptive routing with odd-even and negative first routing with random switching, bitreversal and butterfly traffics is also shown. A comparison also suggests that at high *pir* odd-even routing should be used for smaller delay penalty. It can also be found from Fig. 3 that fully adaptive routing scheme with NOP switching and butterfly traffic results in smallest delay.

Further, energy consumption of fully adaptive routing was also found smallest on any combination of selection strategies and traffics. For small *pir*, butterfly traffic results in smallest energy consumption. However, for high *pir*, random traffic with buffer level or random selection scheme results in lowest energy consumption which can be seen in Fig. 4.

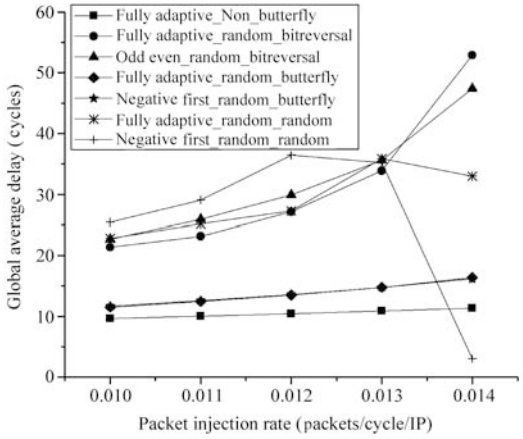


Fig. 3 Delay comparison

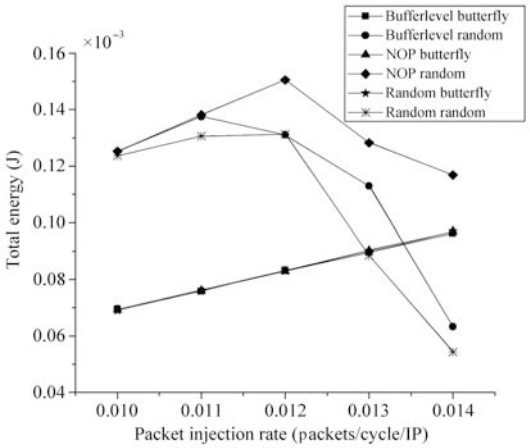


Fig. 4 Energy consumption comparison

Fully adaptive routing has the smallest maximum delay with any combination of traffic scenarios and selection schemes, except for bufferlevel and random selection with bitreversal traffic where it is comparable to odd-even routing that offer smallest delay (see Fig. 5). As can be seen from Fig. 5 that fully adaptive routing has smaller maximum delay at small *pir* but the maximum delay increases as the *pir* is increased, hence odd even routing may be used at high *pir* with this configuration of switching scheme and traffic. Also, the smallest maximum delay is offered by fully adaptive routing with NOP switching scheme and butterfly traffic, which is plotted with respect to *pir* in Fig. 5.

Fully adaptive routing with transpose 1 traffic and random switching results in highest throughput at high *pir*, while at low *pir* highest throughput was observed for transpose 1 traffic with bufferleve 1 switching, as shown in Fig. 6. So far, fully adaptive routing with NOP switching and butterfly traffic has been advantageous in terms of delay and energy consumption, we evaluate the throughput also for the same configuration in Fig. 6. In the next section, we will

compare the performance of proposed partially interconnected mesh network with fully adaptive routing on ideal mesh network. A new routing algorithm is developed and applied to the proposed network in performance evaluation.

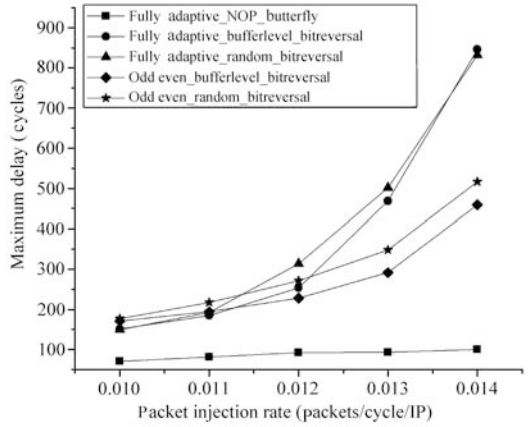


Fig. 5 Maximum delay comparison

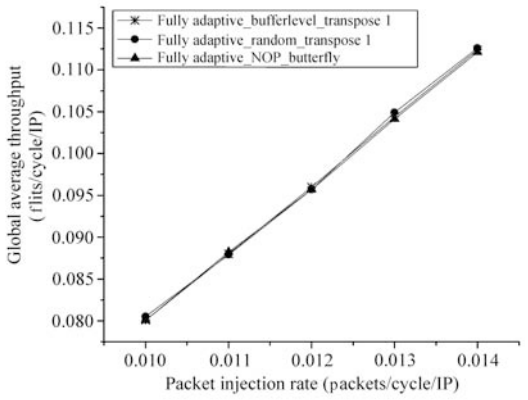


Fig. 6 Throughput comparison

4 Proposed architecture and routing

Full mesh or fully interconnected topology is a network topology in which all nodes are directly connected to each other. In a fully interconnected network with n nodes, there are $n(n-1)/2$ direct links. Full mesh networks are very costly to setup, but provide a highly reliable data transfer because of the availability of multiple paths to a node. This topology is used when there are only a small number of nodes to be interconnected. These networks are usually seen in military applications[32].

Partially interconnected mesh is a network topology in which some of the nodes of the network are connected to more than one other node in the network with a point-to-point link. This makes it possible to take advantage of some of the redundancy that is provided by a physical fully connected mesh topology without the expense and complexity required for a connection between every node in the network[32]. In this section, a new architecture based on partially connected mesh topology is proposed and a routing algorithm for this new topology is developed. We use Nirgam[12] simulator for evaluation. This simulator supports both mesh and torus networks. The simulator was modified to support the proposed new architecture and a routing algorithm was also written for the simulator to send packets on the proposed network. In this new architecture we add four extra bidirectional channels to each router of the mesh network, so that there are nine bidirectional communication channels in each router namely East, West, North, South, Local, Northeast (NE), Southeast (SE), Northwest (NW) and Southwest (SW) (see Fig. 7).

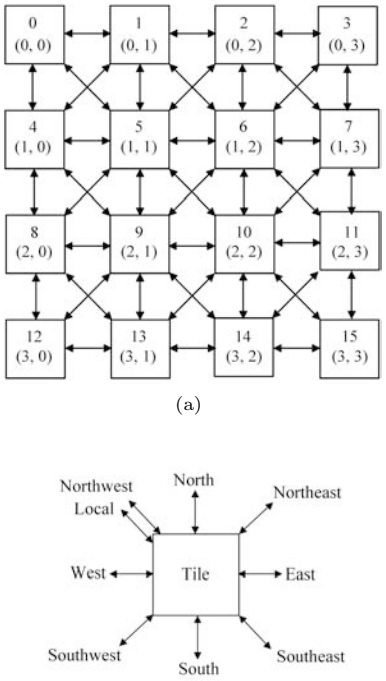


Fig. 7 NoC architecture under investigation: (a) Proposed partially connected mesh topology where tiles are connected through bidirectional channels; (b) Representation of a tile with its bidirectional channels

The performance of NoC is evaluated on per-channel basis. Fig. 8 (a) shows representation of performance metrics by Matlab generated graphs. $R0-R15$ show the placement of tiles/routers. Red bar between $R0$ and $R1$ represents metric for east channel from $R0$ to $R1$. Blue bar between $R0$ and $R1$ represents metric for west channel from $R1$ to $R0$. Green bar between $R0$ and $R4$ represents metric for south channel from $R0$ to $R4$. Orange bar between $R0$ and $R4$ represents metric for north channel from $R4$ to $R0$. Similarly in Fig. 8 (b), Red bar between $R1$ and $R4$ represents metric for Southwest channel from $R1$ to $R4$. Blue bar between $R5$ and $R0$ represents metric for Northwest channel from $R5$ to $R0$. Green bar between $R0$ and $R5$ represents metric for Southeast channel from $R0$ to $R5$. Orange bar between $R4$ and $R1$ represents metric for Northeast channel from $R4$ to $R1$.

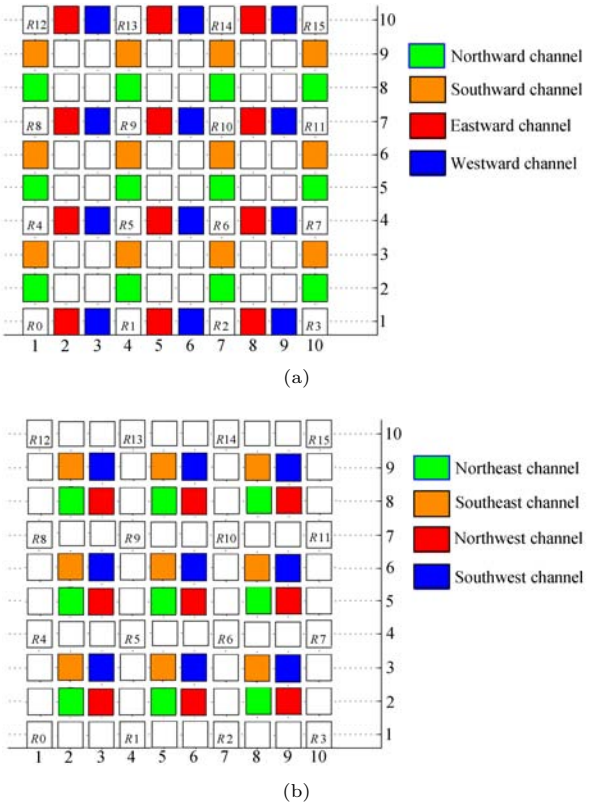


Fig. 8 Representation of performance metrics on per channel basis: (a) North, south, east and west channels; (b) Northeast, Southeast, Northwest and Southwest channels. $R0$–$R15$ represents the routers

In a standard mesh network each router/tile has four bidirectional ports/channels for communicating with other routers in the network and has one bidirectional channel for communicating with the local IP attached to tile. Routing algorithms for such network are mainly based on communication in x-direction and y-direction, using turn models that routes the packet by turning in x-direction (using east or west channel) or y-direction (using north or south channel). Dead-locks in some unique cases in turn models are avoided by prohibiting few turns while routing a packet from source node to destination node. In the new architecture, packets can also be routed in z-direction (NE, SE, NW and SW) along with the usual x- and y-directions. This saves the number of routers traversed which in turn applies the use of free router channels by other communications. For example, if a packet has to move from source tile number 12 to destination tile number 7 using mesh network of Fig. 1 (a), the path could be through tiles 12-13-14-15-11-7. However, the path is through tiles 12-9-6-7 when network of Fig. 7 (a) is used, this saves time by routing through fewer tiles and also allows the free channels/paths for other communications. Higher throughput and smaller latency is confirmed through simulations on this architecture.

Routing algorithm for this architecture is called cross-routing and is given in Fig. 9 (a). While moving a packet, the x-coordinate and y-coordinate of source and destination tiles are compared. If both x- and y-coordinates of source and destination tiles are not equal, packet is moved in z-direction (using NE, SE, NW or SW channels), else packet moves in either x- or y-direction depending on whether the x-coordinate or y-coordinate of source tile is equal to that of destination tile. We make use of turn model to avoid deadlocks. The case of a two-dimensional (2D) mesh in Fig. 9 (b) uses the turn model. There are eight possible turns and two possible abstract cycles. If the turns are not restricted, cycles among packets may result. The XY routing algorithm prevents deadlock by prohibiting four of the turns, as shown in Fig. 9 (b, ii). The remaining four turns cannot form a cycle, but neither do they allow any adaptiveness. The concept behind the turn model is to prohibit the smallest number of turns such that cycles are prevented. In fact, for a 2D mesh, only two turns need to be prohibited. Six turns are allowed in the west-first routing algorithm: a packet is routed in the west first, if necessary, and then adaptively south, east, and north. The two turns prohibited in Fig. 9 (b, iii) are the two turns to the west. Therefore, to travel west, a packet must begin in that direction[18].

The proposed cross-routing is free of dead-locks since it does not form loops, third turn is prohibited so that a packet reaches the destination by taking maximum two turns, one in z-direction and other in x- or y-direction. For a better understanding, algorithm is divided into 3 parts as shown in Fig. 9 (c) below. Explanation for part (1) and (2) of routing algorithm is presented, whereas part (3) is self explanatory. While moving a packet, the x-coordinate and y-coordinate of source and destination tiles are compared. If both x- and y-coordinates of source and destination tiles are not equal, packet is moved in z-direction (using NE, or SE (ie., part (1)) and NW or SW channels (ie., part (2)) in Fig. 2), else packet moves in either x- or y-direction depending on whether the x-coordinate or y-coordinate of source tile is equal to that of destination tile.

4.1 Experimental results and discussion on the proposed scheme

The performance evaluation is based on standard 4×4 mesh network of Fig. 1 (a) and partially connected mesh network of Fig. 7 (a). In Section 3, it was found that fully adaptive routing was superior to all other routing schemes. Hence we evaluate and compare the performance of these networks on fully adaptive and cross-routing algorithm. Also, we consider tiles of both the networks transmitting packets at constant bit-rate (CBR) with random destinations and at constant bit-rate with fixed destinations. In CBR with random destinations, each tile behaves as a source as well as sink, implying that it is capable of generating as well as receiving flits. Each tile generates CBR traffic to randomly chosen destination. For CBR with fixed destinations, we consider an example shown in Fig. 10, where, tile 0 sends CBR traffic to tile 6, tile 5 sends CBR traffic to tile 2, tile 10 sends CBR traffic to tile 0, tile 11 sends CBR traffic to tile 13 and tile 14 sends CBR traffic to tile 5. Fig. 10 (a) shows the possible path taken by the packets to move from sources to fixed destinations on standard mesh network of Fig. 1 (a) using fully adaptive routing, while Fig. 10 (b) uses partially connected mesh network of Fig. 7 (a) with cross-routing.

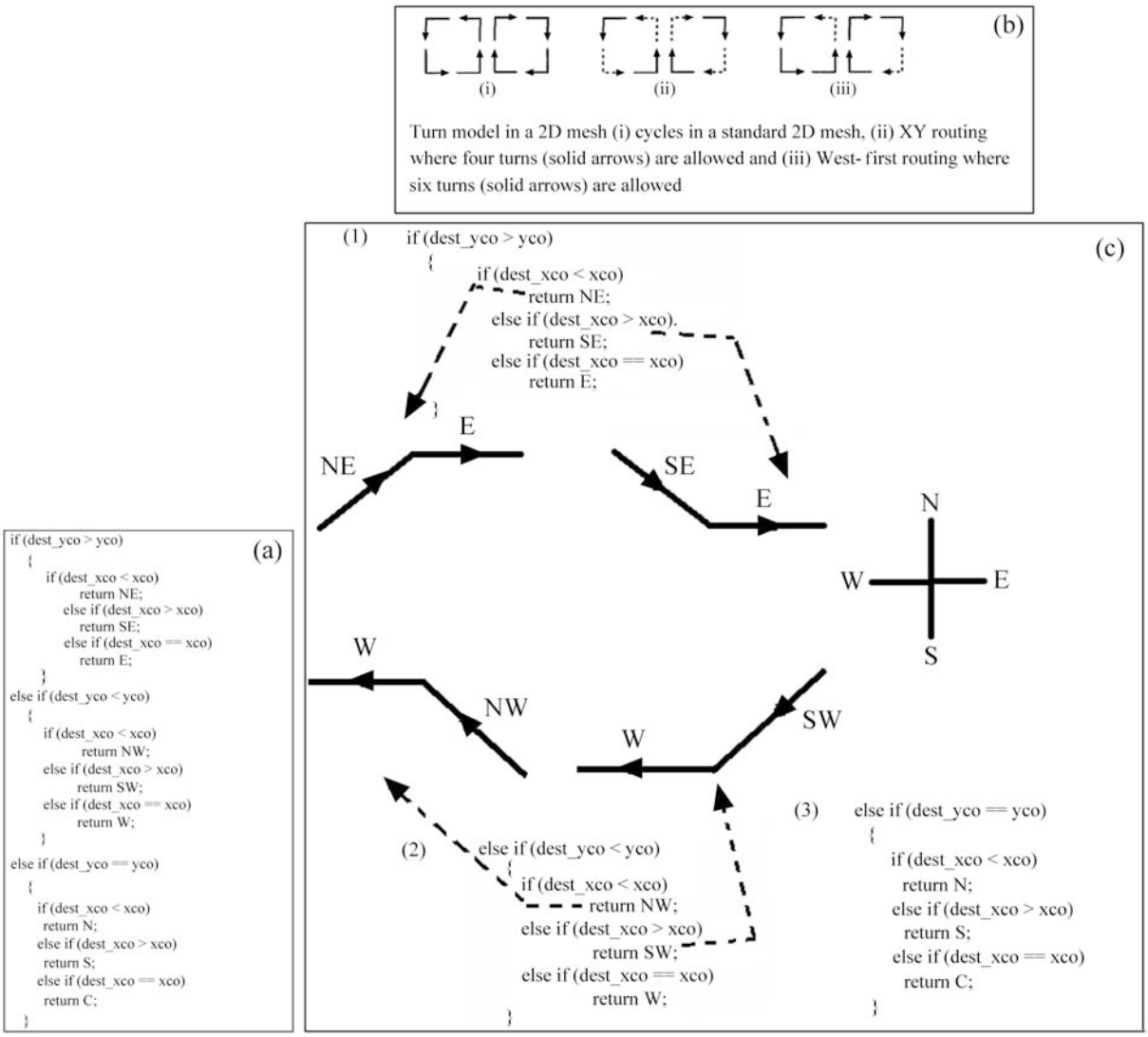


Fig. 9 (a) Cross-routing algorithm where "xco" and "yco" are x- and y-coordinates of source tile, "dest_xco" and "dest_yco" are coordinates of destination tile; (b) Illustration of turn model[18]; (c) Illustration of deadlock free cross routing

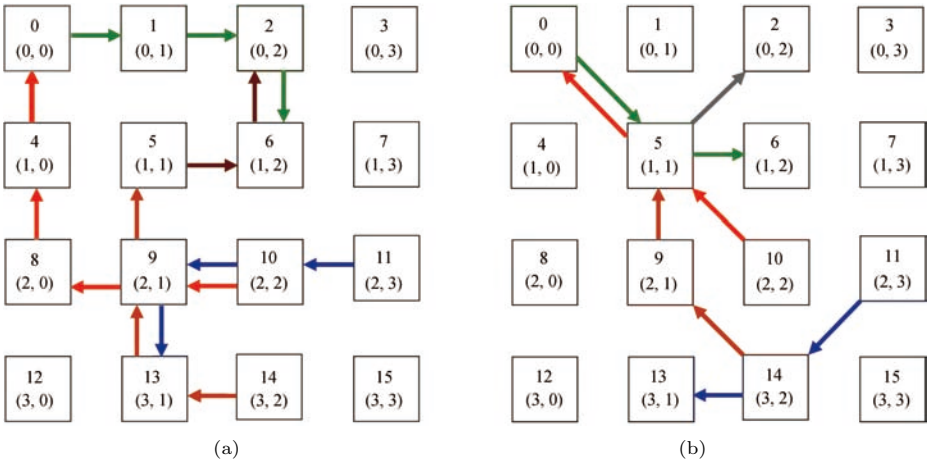
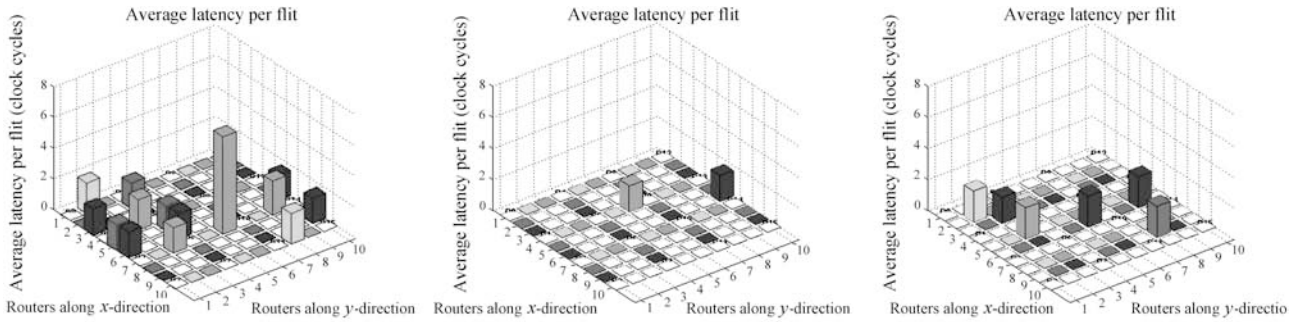
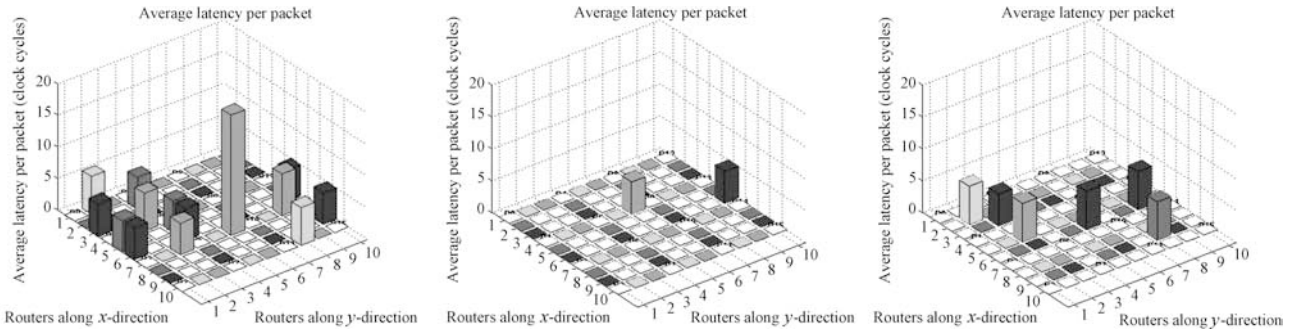


Fig. 10 Constant bit-rate (CBR) with fixed destination on: (a) Standard mesh with fully adaptive routing; (b) Partially connected mesh with cross-routing

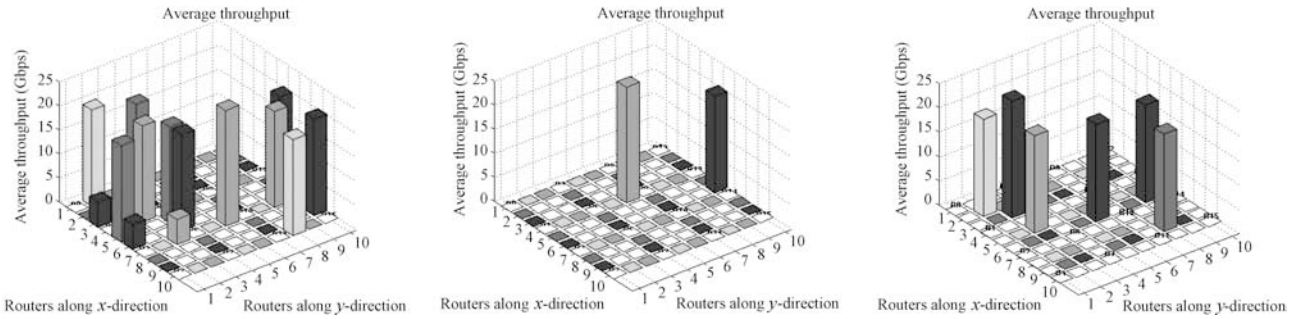
Performance of both the NoCs are measured on per channel basis. Traffic generation begins after 5 clock cycles, and continues until 300 clock cycles. Simulation stops after 1000 clock cycles. Latency and throughput per-channel for CBR with fixed destination are measured and plotted in Fig. 11. Overall average latency is shown in Fig. 11 (a), which is smaller in case of cross-routing (1.78, in clock-cycles per flit) compared to fully adaptive routing on standard mesh NoC (2.13, in clock cycles per flit). Overall average throughput (see Fig. 11 (b)) is also higher for cross-routing scheme (28 Gbps) compared to fully adaptive routing on standard mesh NoC (24 Gbps). Similar results are obtained for CBR with random destinations and are shown in Fig. 12. The results in Fig. 12 (b) and (c) show that for cross-routing the overall average latency reduces by 60 % and the overall average throughput increases by 60 % in comparison to fully adaptive routing on standard mesh NoC. This improvement in performance is possible because lesser nodes are traversed during packet transmission thus offering greater number of available free channels which could be utilized by other communications.



(a) Latency per flit on per channel basis

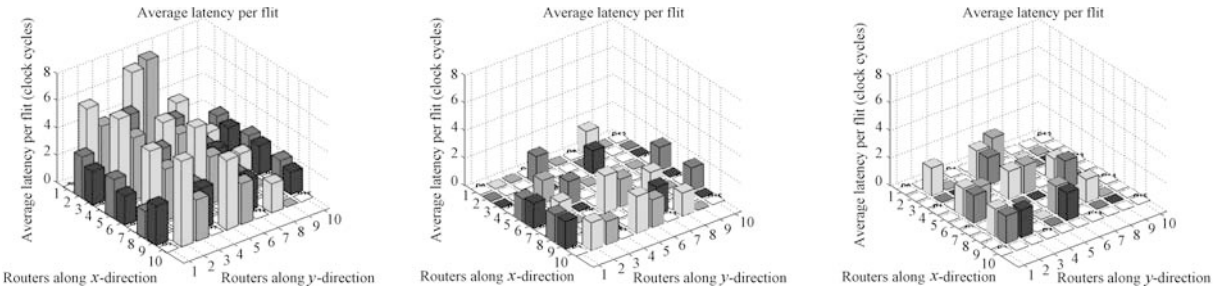


(b) Latency per packet on per channel basis



(c) Throughput per channel

Fig. 11 Simulation results for fixed destination corresponding to Fig. 10: (a) Latency per flit; (b) Latency per packet; (c) Throughput. Three results are shown in each case where first result is obtained for mesh network using fully adaptive routing, second and third results are for the proposed network topology on cross-routing. Also in each case, second result shows measurements on east, west, north and south channel and third result shows measurements on north east, south east, north west and south west channels



(a) Latency per flit for random destination, other details remain same as in Fig. 11 above

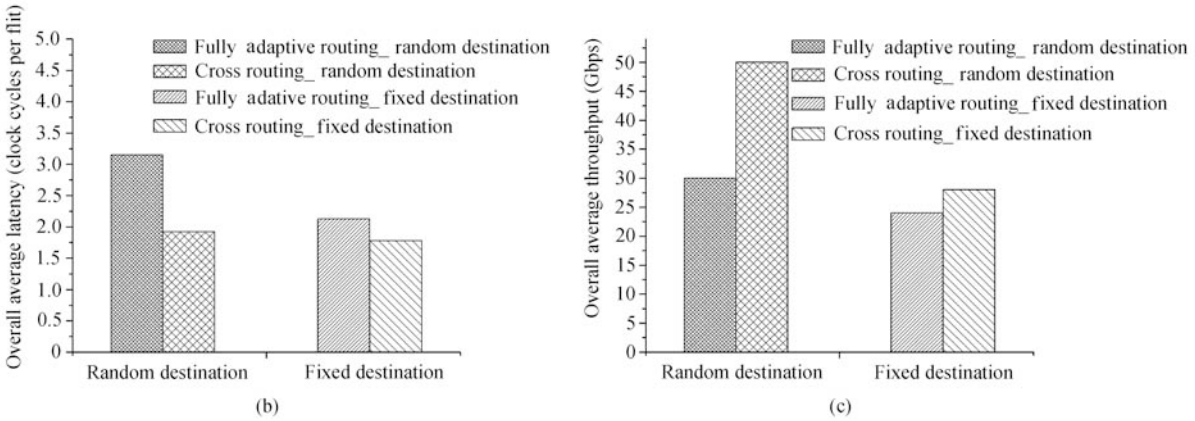


Fig. 12 Simulation results: (a) Latency per flit for random destination; (b) Overall average latency for fixed and random destination; (c) Overall average throughput for fixed and random destination

5 Conclusions

In this paper we have evaluated the performance of standard mesh-based NoC architecture and compared it with the proposed partially interconnected network architecture. We started by evaluating different routing algorithms and switching schemes on a standard mesh topology and then proposed a partially interconnected mesh topology with cross-routing algorithm. From the performance comparison, it was found that the proposed network and routing algorithm can be efficiently used for future NoCs. The experiments were also performed on larger networks and performance measurements gave significant improvements. While the proposed topology provides some improvements (e.g., increased bandwidth, decreased latency) over standard mesh, it effectively duplicates the number of links making switches expensive. Another limitation with the cross routing algorithm is that it uses diagonal links heavily as compared to X-Y links. It is also expected that this topology could increase the die area and power consumption but still it can be useful for high speed applications with some compromise in power and area.

References

[1] S. Kumar, A. Jantsch, J. P. Soininen, M. Forsell, M. Millberg, J. Oberg, K. Tiensyrja, A. Hemani. A network on chip architecture and design methodology. In *Proceedings of IEEE Computer Society Annual Symposium on VLSI*, IEEE, Pittsburgh, USA, pp. 105–112, 2002.

[2] M. Millberg, E. Nilsson, R. Thid, S. Kumar, A. Jantsch. The Nostrum backbone—a communication protocol stack for networks on chip. In *Proceedings of the 17th International Conference on VLSI Design*, IEEE, pp. 693–696, 2004.

[3] L. Benini, G. D. Micheli. Powering networks on chips: Energy-efficient and reliable interconnect design for SoCs. In *Proceedings of the 14th International Symposium on Systems Synthesis*, pp. 33–38, 2001.

[4] L. Benini, G. D. Micheli. Networks on chips: A new SoC paradigm. *IEEE Computer*, vol. 35, no. 1, pp. 70–78, 2002.

[5] P. Guerrier, A. Greiner. A generic architecture for on-chip packet-switched interconnections. In *Proceedings of Design Automation and Test in Europe Conference and Exhibition*, IEEE, Paris, France, pp. 250–256, 2000.

[6] E. Bolotin, I. Cidon, R. Ginosar, A. Kolodny. Cost considerations in network on chip. *Integration, The VLSI Journal*, no. 38, no. 1, pp. 19–42, 2004.

[7] C. A. Zeferino, M. E. Kreutz, L. Carro, A. A. Susin. A study on communication issues for systems-on-chip. In *Proceedings of the 15th Symposium on Integrated Circuits and Systems Design*, IEEE, pp. 121–126, 2002.

[8] E. Rijpkema, K. Goossens, A. Radulescu, J. Dielissen, J. van Meerbergen, P. Wielage, E. Waterlander. Tradeoffs in the design of a router with both guaranteed and best-effort services for networks on chip. In *Proceedings of Design, Automation and Test in Europe Conference and Exhibition*, IEEE, Munich, Germany, pp. 350–355, 2003.

[9] E. Rijpkema, K. Goossens, P. Wielage. A router architecture for networks on silicon. In *Proceedings of the 2nd Workshop on Embedded Systems*, pp. 181–188, 2001.

[10] B. S. Feero, P. P. Pande. Networks-on-chip in a three-dimensional environment: A performance evaluation. *IEEE Transactions on Computers*, vol. 58, no. 1, pp. 32–45, 2009.

[11] Noxim: Network-on-chip simulator, [Online], Available: http://sourceforge.net/projects/noxim, November 26, 2011.

[12] L. Jain, B. M. Al-Hashimi, M. S. Gaur, V. Laxmi, A. Narayanan. NIRGAM: A simulator for NoC interconnect routing and application modeling. In *Proceedings of the Friday Workshop on Diagnostic Services in Network-on-Chips, Design, Automation and Test in Europe Conference*, France, 2007. [Online], Available: http://www.date-conference.com/files/file/10-ubooth/ub-1.4-p04.pdf, November 26, 2011.

[13] D. A. Patterson, J. L. Hennessy. *Computer Architecture: A Quantitative Approach*, San Francisco, USA: Morgan Kaufmann Publishers, pp. 760, 1996.

[14] F. Cen, T. Xing, K. T. Wu. Real-time performance evaluation of line topology switched ethernet. *International Journal of Automation and Computing*, vol. 5, no. 4, pp. 376–380, 2008.

[15] P. Mohapatra. Wormhole routing techniques for directly connected multicomputer systems. *ACM Computing Surveys*, vol. 30, no. 3, pp. 374–410, 1998.

[16] W. J. Dally, C. L. Seitz. The torus routing chip. *Distributed Computing*, vol. 1, no. 4, pp. 187–196, 1986.

[17] Y. Wu, G. Min, M. Ould-Khaoua, H. Yin. An analytical model for torus networks in the presence of batch message arrivals with hot-spot destinations. *International Journal of Automation and Computing*, vol. 6, no. 1, pp. 38–47, 2009.

[18] L. M. Ni, P. K. McKinley. A survey of wormhole routing techniques in direct networks. *IEEE Computer*, vol. 26, no. 2, pp. 62–76, 1993.

[19] J. Duato, S. Yalamanchilli, L. M. Ni. *Interconnection Networks: An Engineering Approach*, San Francisco, USA: Morgan Kaufmann Publishers, 2002.

[20] G. Ascia, V. Catania, M. Palesi, D. Patti. Implementation and analysis of a new selection strategy for adaptive routing in networks-on-chip. *IEEE Transactions on Computers*, vol. 57, no. 6, pp. 809–820, 2008.

[21] D. Frazzetta, G. Dimartino, M. Palesi, S. Kumar, V. Catania. Efficient application specific routing algorithms for NoC systems utilizing partially faulty links. In *Proceedings of the 11th EUROMICRO Conference on Digital System Design, Architectures, Methods and Tools*, IEEE, Parma, Italy, pp. 18–25, 2008.

[22] R. Tornero, J. M. Orduna, M. Palesi, J. Duato. A communication-aware topological mapping technique for NoCs. In *Proceedings of International Conference on Parallel and Distributed Computing*, ACM, pp. 910–919, 2008.

[23] M. Palesi, G. Longo, S. Signorino, S. Kumar, R. Holsmark, V. Catania. Design of bandwidth aware and congestion avoiding efficient routing algorithms for networks-on-chip platforms. In *Proceedings of IEEE International Symposium on Networks-on-Chip*, IEEE, Newcastle upon Tyne, UK, pp. 97–106, 2008.

[24] G. Longo, S. Signorino, M. Palesi, S. Kumar, R. Holsmark, V. Catania. Bandwidth aware routing algorithms for networks-on-chip. In *Proceedings of the 2nd Workshop on Interconnection Network Architectures: On-Chip, Multi-Chip*, 2008, [Online], Available: http ://www. diit. unict. it /users /mpalesi /DOWNLOAD / inaocmc08.pdf, November 26, 2011.

[25] R. Tornero, J. M. Orduna, M. Palesi, J. Duato. A communication-aware task mapping technique for NoCs. In *Proceedings of the 2nd Workshop on Interconnection Network Architectures: On-Chip, Multi-Chip*, MENDELEY, 2008.

[26] M. Palesi, S. Kumar, R. Holsmark, V. Catania. Exploiting communication concurrency for efficient deadlock free routing in reconfigurable NoC platforms. In *Proceedings of IEEE International Parallel and Distributed Processing Symposium*, IEEE, Long Beach, USA, pp. 1–8, 2007.

[27] G. Ascia, V. Catania, M. Palesi, D. Patti. Neighbors-on-path: A new selection strategy for on-chip networks. In *Proceedings of the 4th IEEE Workshop on Embedded Systems for Real Time Multimedia*, IEEE, Seoul, Korea, pp. 79–84, 2006.

[28] M. Palesi, R. Holsmark, S. Kumar, V. Catania. A methodology for design of application specific deadlock-free routing algorithms for NoC systems. In *Proceedings of International Conference on Hardware-Software Codesign and System Synthesis*, IEEE, Seoul, Korea, pp. 142–147, 2006.

[29] M. Palesi, S. Kumar, R. Holsmark. A method for router table compression for application specific routing in mesh topology NoC architectures. In *Proceedings of SAMOS VI Workshop: Embedded Computer Systems: Architectures, Modeling, and Simulation*, pp. 373–384, 2006.

[30] G. Ascia, V. Catania, M. Palesi, D. Patti. A New Selection Policy for Adaptive Routing in Network on Chip. In *Proceedings of International Conference on Electronics, Hardware, Wireless and Optical Communications*, ACM, pp. 94–99, 2006.

[31] P. P. Pande, C. Grecu, M. Jones, A. Ivanov, R. Saleh. Performance evaluation and design trade-offs for networkon-chip interconnect architectures. *IEEE Transactions on Computers*, vol. 54, no. 8, pp. 1025–1040, 2005.

[32] Network topology, [Online], Available: http://en.wikipedia.org/wiki/Tree_and_hypertree_networks #Tree, November 26, 2011.

Sudhanshu Choudhary received bachelor degree in electronics and communication engineering from Birla Institute of Technology, MESRA Ranchi, India in 2002 and master degree from Indian Institute of Information Technology & Management Gwalior, India in 2006. He joined IIT Kanpur in 2006 as a Ph. D. candidate in the Department of Electrical Engineering at IIT Kanpur.

His research interests include microelectronics, very large scale integration (VLSI) testing and physical design.

E-mail: suds@iitk.ac.in (Corresponding author)

Shafi Qureshi received Bachelor degree in electrical engineering from University of Kashmir, India in 1974 and Ph. D. degree from University of California, Berkeley, USA in 1991. He joined IIT Kanpur in 1992 and became professor in the Department of Electrical Engineering in 1997. He is senior member of IEEE and fellow of Institution of Electronics and Telecommunication Engineers (IETE) India.

His research interests include semiconductor device physics and modeling, thin film transistors, and VLSI design.

E-mail: qureshi@iitk.ac.in