BOFAR: Buffer Occupancy Factor based Adaptive Router for Mesh NoCs

John Jose J. Shiva Shankar K.V. Mahathi
Damarla Kranthi Kumar Madhu Mutyam

Computer Architecture and Systems Laboratory
Department of Computer Science and Engineering
Indian Institute of Technology Madras
Chennai-36, India
{johnjose, jss, mahathi, kranthi, madhu}@cse.iitm.ac.in

ABSTRACT

If the route computation operation in an adaptive router returns more than one output channels, the selection strategy chooses one from them based on the congestion metric used. The effectiveness of a selection strategy depends on what metric is used to identify congestion and how precisely that metric captures the actual congestion. The number of cycles a flit stays in a router is a direct indication of the contention level of the output port it desires to move out. We propose Buffer Occupancy Factor based Adaptive Router (BOFAR), wherein the history of cycles spent by flits in buffers is used as the congestion metric. BOFAR outperforms the baseline architectures built on minimal odd-even adaptive router model with conventional selection strategies like count of free downstream virtual channels at reachable neighbors, and fluidity of buffers in downstream neighbors. Our experiments on 4x4 mesh NoC with various synthetic traffic patterns show that BOFAR exceeds the performance of best baseline adaptive router with 21% average and 78% maximum latency reduction at saturation load. The reduced average packet latency, increased buffer fluidity fairness, and increased saturation point of BOFAR with minimal overhead in area, power, and wiring makes it a promising alternative to existing adaptive routers in mesh NoCs.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: [Network communications]

Keywords

Network-on-chip, adaptive routers, buffer occupancy

1. INTRODUCTION

Apart from high speed computing cores, efficient and reliable communication is also essential for achieving high performance and throughput in modern day processors. Network-on-Chip (NoC) is the most widely accepted model for implementing interconnections in multi core processor designs. Instead of bus based communication with dedicated point-to-point links, NoCs employ a grid of routers organized across the chip, connected by communication links [8]. Each router is associated with a processing core and cores use packet based communication between them. Packet header contains the control information needed to process the packet [1]. A conventional baseline router contains a 5x5 crossbar and employs wormhole routing [9]. Every input port of a router is associated with a set of virtual channels (VC). The use of VCs reduces the network latency at the expense of area and power consumption [1, 3, 7].

Once a flit reaches a router, buffers in the VCs provide interim storage space for it. The control information embedded in the head flit is used by the routers to determine the proper output port and VC for that packet. Route computation and VC allocation is done on each head flit. Body flits are assigned the same output port and VC as that of the head flit [20]. The flit then arbitrates with flits in other VCs competing for the same output port. Once the arbitration process is over, switch allocation is done. Winning flits traverse the switching fabric and reach the links. Buffering within the routers and two way handshaking between routers enable the smooth flow of the packets.

Eventhough NoC architectures offer higher bandwidth compared to traditional bus based designs, their performance can degrade significantly in the absence of effective congestion control algorithms [6]. Since resources like inter-router link capacity and buffers are limited, there may be resource contention. Delay in delivery of packets hits the performance of parallel applications running on the cores. Hence for building an efficient NoC based system, an effective mechanism that monitors the packets flowing through the network is needed. Such mechanism should foresee congestion situations and take necessary steps that facilitates congestion relief [17, 22].

Many of the existing adaptive routers use availability of free VCs across downstream nodes as the congestion metric [15]. Our experimental observations on various traffic patterns show that the real congestion status of a router cannot be fully represented by the count of free VCs on it and its downstream routers. If there is congestion in a router, the flits moving through the router stay in router buffers. It could be either due to link contention or due to

NoCArc '11, December 4, 2011, Porto Alegre, Brazil

unavailability of free VC in the downstream node. The number of cycles a flit stays in a router is a direct indication of the contention level of the output port it desires to proceed.

In this paper, we propose a novel adaptive router for 2-D mesh topology that uses history of buffer occupancy time of flits as the congestion metric. Every router keeps track of the extra delay encountered by each flit passing through that router. The tracked values are exchanged to the neighboring routers, which make use of these buffer occupancy estimates in output port selection of incoming flits. In port selection decisions neighbors with less buffer occupancy history are preferred. The proposed model incurs slight overhead on router area and power but no impact on its critical path, compared to a conventional adaptive router. The following are our main contributions:

- We propose Buffer Occupancy Factor based Adaptive Router (BOFAR) for 2-D mesh topology that uses the cumulative history of buffer occupancy time of flits as the congestion metric for output port selection.

- We compare the performance of the proposed model with other baseline router architectures employing minimal odd-oven (MOE) routing with selection techniques like free-VCs [7, 15], Neighbors on Path (NOP) [2], and Fluidity of downstream neighbors (FON) [16].

- We explore the effectiveness of BOFAR by considering buffer fluidity fairness of routers and saturation point of the network, in comparison with other baseline architectures.

2. RELATED WORK

Dimension Order Routing (DOR) such as XY routing has been a common choice in first generation mesh NoCs. Even though XY routing is very popular due to its simplicity, adaptive routing techniques provide better throughput and fault tolerance by allowing alternative paths [10, 12]. The turn models and odd-even models are the most commonly used partially adaptive deadlock free routing algorithms. These two routing models are not strictly minimal in all cases. Non-minimal routing leads to livelocks in certain conditions [12]. The odd-even adaptive routing model is preferred over various turn models because the degree of adaptiveness is evenly distributed across the network. The DyAD smart routing [12] effectively switches between the adaptive and deterministic routing based on network congestion conditions. When the network is free from congestion, the router works in a low latency deterministic mode. When it detects congestion, the router switches to the adaptive routing mode by exploiting paths through non-congested nodes.

Modern NoC designs need low latency and deadlock free adaptive routers. Once a packet reaches a router, adaptive routing algorithms identify a set of possible output channels. A selection strategy chooses one among these channels that allows the packet to reach its destination with minimum latency. To facilitate this, the selection strategy should avoid congested nodes. Proximity Congestion Awareness technique [19] makes use of load information of neighboring switches, called *stress values*, for channel selection decisions. The *stress value* is defined as the number of flits a switch handles during a cycle and this count is sent to all the neighbors of the switch.

Congestion Aware Deterministic Routing (CADR) [14] proposes a cost effective method to estimate congestion level in the network. Based upon this estimate they compute optimizing routing paths for all traffic flows. This approach is deterministic and is best suited for reconfigurable systems which run several applications with regular and repetitive computations on large set of data. Moreover the performance of CADR is not better than the DyAD model. Path-Based Randomized Oblivious Minimal Routing (PROM) [5] proposes a load balancing routing scheme which explores the path diversity in routes. PROM achieves load balancing through randomization. The performance of an adaptive routing scheme with properly measured congestion framework could not be achieved by a random selection approach eventhough the overhead in randomization is less.

Another class of selection technique is based on the count of free VCs (FVC) in downstream nodes. VC flow control [7] and low latency adaptive router [15] work with this strategy. In every cycle the count of free VCs is passed to the neighbors, and upstream nodes make channel selection based on this. The Neighbors-on-Path (NOP) [2] strategy explores the free VC status of reachable neighbors of adjacent routers of current node. This allows each router to monitor the router buffer status of two-hop neighbors. This helps in detecting potential congestion earlier. Since NOP technique act upon congestion status beyond neighboring nodes, it is more effective than FVC approach. But in NOP, channel selection is based on 2 cycle old FVC status of two-hop neighbor. Because of this, NoP shows inconsistent results in certain cases eventhough the average packet latency under NoP technique is better than all other techniques discussed above.

These approaches [2, 7, 15] are not looking into the real cause for congestion, but they simply act upon count of unassigned VCs. Techniques that look into real cause of congestion and taking remedial steps that alleviate the situation only will bring down average packet latency. The fluidity based Congestion Avoidance Scheme (CAS) [16] is a promising work in this direction. A buffer is said to be fluid when a flit is moving out from it in the current clock cycle. Fluid buffers indicate easy flow of flits. In this model, nodes with more fluid buffers are treated as less congestion prone. The fluidity is measured and exchanged at regular intervals and routers make adaptive routing decisions based on this downstream fluidity information. One of the main drawbacks of CAS is its inability to distinguish the level of congestion if both neighbors of a node are either equally fluid or equally non-fluid. BOFAR also works on the basis of fludity, but we define different levels of fluidity and tackle congestion more effectively.

Regional Congestion Awareness (RCA) is the first work to present a comprehensive evaluation and usage of non-local congestion information for improving the dynamic load balancing properties of fully adaptive minimally routed networks [11]. A composite metric consisting of number of virtual channels, free buffers, and crossbar demand is used in this scheme. Destination Based Adaptive Routing (DBAR) [21] also uses non-local congestion estimates in route selection. Both RCA and DBAR techniques aggregate locally computed congestion metrics with the propagated congestion estimates from neighbors. They send the aggregated congestion information about a region to upstream nodes. Since these two models gather congestion information from

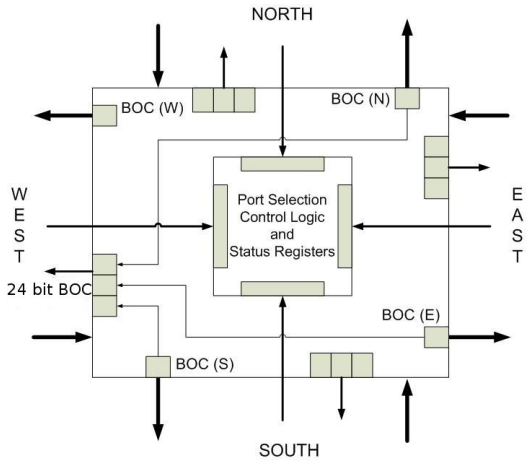

Figure 1: Internal architecture of BOFAR.

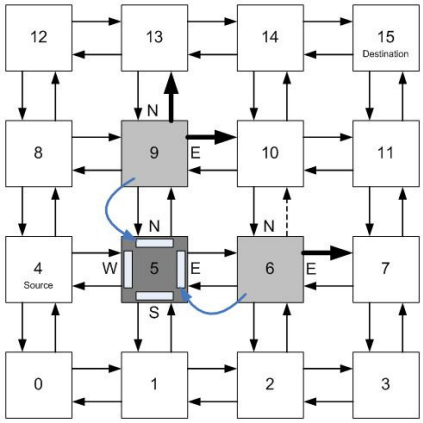

Figure 2: Illustration of exchange of BOC values in BOFAR.

non-local nodes, they belong to a totally different class of adaptive routers by virtue of capturing, computing, aggregating and propagating non local congestion information. So we are not comparing our technique with these two methods. BOFAR is an adaptive router that works with local congestion information only.

3. BOFAR ROUTER OVERVIEW

BOFAR is basically a VC-based router [9] consisting of VC-alloc unit, routing unit, and switch-alloc unit. It takes minimum of two cycles for a flit to move to next outgoing link once it reaches a router; one cycle for route computation and one cycle for VC and switch allocation together [9]. If two flits residing in two different input buffers of a router compete for the same output port, only one will win and other has to stay back in the buffer and try its chance in the next cycle. Similarly flits will stay back in a router buffer due to unavailability of buffers in the downstream router. Every extra cycle a flit stays back in the buffer reduces the fluidity of the buffer. We modify the conventional router design to capture this fluidity level variation of input buffers and effectively propagate it to neighbors.

3.1 The Router Architecture

Our router incorporates a set of counters to track the time spent by a flit in a router. This is recorded when a flit moves from the input VC to the flit channel. Figure 1 describes the internal architecture of BOFAR. An 8-bit *Buffer Occupancy Counter (BOC)* is attached to each output port of a router. When a flit reaches a router, the arrival cycle time is recorded on the flit. When a flit moves out to the flit channel from the input VC by winning the switch arbitration, departure time of the flit is noted. The difference in departure and arrival time is the *Cycles Stayed in the Buffer (CSB)* of the flit. This *CSB* value is added to the *BOC*. In certain cycles no flit moves out through a port, wherein the *BOC* retains its previous cycle value. In every cycle the 8-bit *BOC* value associated with each port of a router is collected. The 24-bit combined *BOC* value for a port is obtained from 8-bit *BOC* values of other 3 ports. This is shown in the west port of router in Figure 1. Each node receives the *BOC* values sent by its neighbors in a 24-bit *status register (SR)* per port. In Figure 2 status registers are shown at node 9. The BOC is cleared at the end of *refresh interval (RI)*. The *RI* is chosen in such a way that the *BOC* value is sufficient to distinguish the fluidity level of two paths. Based on our experimental analysis, we consider RI=128 cycles.

Since the flit delay values are captured at output ports, we can distinguish the contention intensity at each output port and make port selection accordingly. When a flit reaches a router, route computation and processing of *SR* values are done in parallel. Based on the candidate paths selected by the routing algorithm and the possible number of reachable outgoing ports [2] of the neighbors along the candidate path, the *mean-BOC* is computed. Hence the *mean-BOC* computation can happen only after the routing operation. Indepth analysis of critical path delays of various stages of a router in [18] establishes the fact that the VC and switch allocation delay determine the pipeline delay. Their analysis show that routing operation has sufficient slack with respect to critical path delay. So the *mean-BOC* computation and associated channel selection have no impact on the critical path. The candidate path with lower *mean-BOC* represents the router with more fluid buffers. The Port Selection Control Logic assigns highest priority for the path with less *mean-BOC*. Signals for switch allocation are generated to facilitate the flit forwarding along this highest priority path.

3.2 Illustration

As per Figure 2, assume a flit F sourced at node 8 and destined at node 3 reaches node 9. The MOE route function chooses east port (link to node 10) and south port (link to node 5) as possible routes. In the mean time *BOC* values captured from node 10 and node 5 are processed. As per MOE routing, a flit from node 9 destined to node 3, upon reaching node 5 has two possible output links. They are south and east ports (shown by thick arrows) of node 5. Hence the *mean-BOC* for F through the south port of node 9 is the average of *BOC* of east port and south port of node 5. Similarly the *mean-BOC* for F through the east port of node 9 is computed. Note that in this case, the south port of node 10 is not a reachable port (shown by dotted arrow) for a flit coming from node 9 due to MOE turn restriction [4, 2]. Since we have only one possible path for F upon reaching node 10, the *mean-BOC* for F along east port of node 9 is *BOC* of east port of node 10 itself. The output port of node

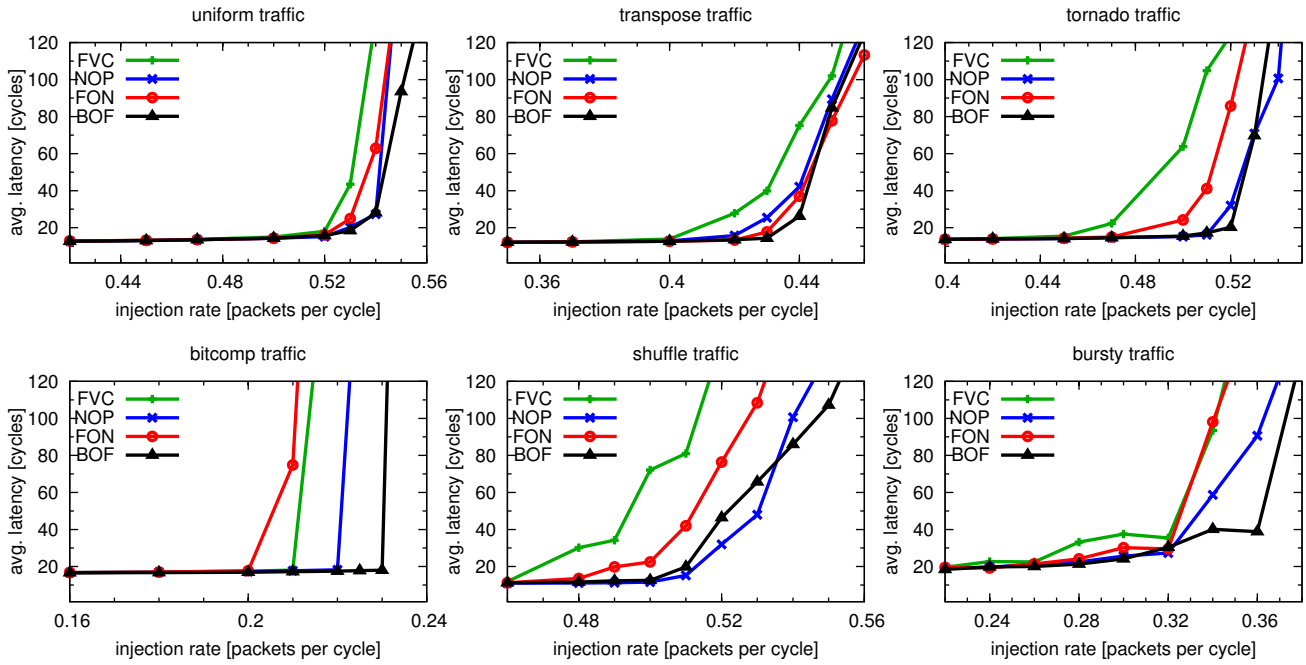

Figure 3: Average packet latency for various traffic patterns (16 core, 4 VCs per router port, packet size=1-flit).

9 with smaller *mean BOC* is given higher priority at the time of port selection for flit F. By this approach, F is forwarded through routers with more fluid buffers. BOFAR ensures that less fluid buffers are made more fluid by reducing flow of flits to it. Thus we regains the lost fluidity of a congested router in short span of time by regulating traffic flow into the router.

4. EXPERIMENTAL RESULTS

We compare the performance of BOFAR with baseline architectures employing various other selection strategies on top of MOE model. We use *Booksim* [9], a cycle accurate network simulator, that models the two cycle router micro-architecture in sufficient detail. We measure the performance of three baseline MOE router architectures: that uses free VCs in downstream neighbor as congestion metric (FVC), that uses free VCs in reachable nodes at two-hop distance as congestion metric (NOP), and that uses fluidity of input buffers in reachable nodes at two-hop distance as congestion metric (FON). Average packet latency and buffer fluidity fairness are collected for different synthetic traffic patterns under various injection rates.

All the experiments are done for 4x4 mesh network that uses 4 VCs per router port. We consider 1-flit packet and hence VC depth is taken as 1-flit buffer. The flit channel is 128-bit wide and the control channel that exchanges *BOC* values ranges from 24-bit wide in central links to 8-bit wide in edge links. The *BOC* is cleared once in every 128 cycles. Traffic model which generates bursty pattern is also incorporated into the simulator with 4 nodes injecting one long packet of 20 flits once in every 20 cycles. Results are plotted for bursty traffic also. For implementing FON model, a buffer is considered as fluid in a particular cycle, if the flit in the buffer is moving through the output port.

4.1 Evaluation of Average Packet Latency

Figure 3 contains a set of load-latency graphs for BOFAR and other baseline architectures across various synthetic traffic patterns. In all patterns BOFAR experiences lesser average latency than baseline routers at saturation throughput. Saturation throughput is measured as the point at which the average packet latency is five times the zero load latency. At saturation loads, BOFAR achieves a latency reduction of 78% maximum and 60% average over FVC, 78% maximum and 45% average over FON and 78% maximum and 21% average over NoP technique. In bursty traffic, BOFAR achieves more than 50% latency reduction than all other three baseline architectures at saturation load. This is because, in BOFAR, the flits generated by the bursty node are spread uniformly across available paths. Other baseline architectures are not able to handle this situation effectively. This makes BOFAR a perfect choice for applications that generate bursty traffic.

4.2 Evaluation of Buffer Fluidity Fairness

We analyze the effect of our selection technique on buffer usage and fluidity. If a flit leaves a buffer, the *fluidity level (FL)* of that buffer at the end of the cycle is computed as follows

$$FL = \frac{1}{1+d}$$

where d is the extra delay in cycles spent in router buffers by that flit. If a flit spends only 2 cycles in a router buffer, then d=0. Hence in those cases, FL=1. FL is fixed as 0 if no flit moves out from a buffer in a particular cycle. Generally, FL=1 at low injection rate, as every flit moves out from a router without any additional delay. As congestion in a router increases, FL value of the corresponding router decreases. More the congestion in a router, longer a flit stays in buffers, higher the value of d, and lower be the value of

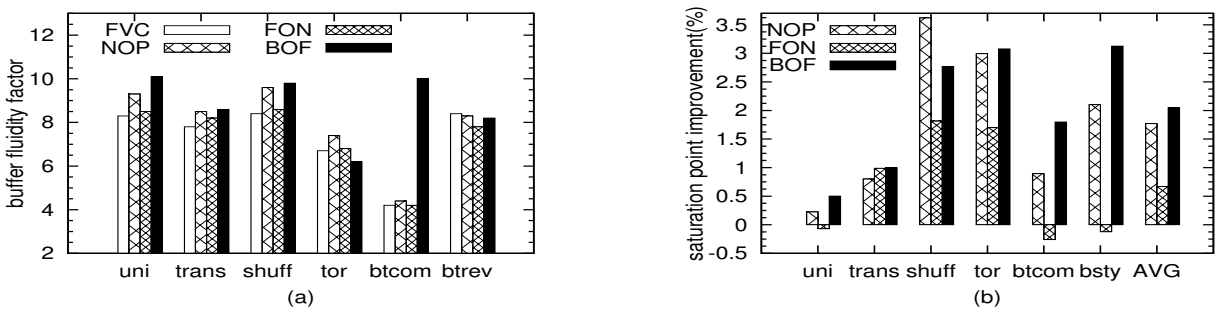

Figure 4: Performance of various selection strategies (16 core, 4 VCs per router port, packet size=1-flit) (a) buffer fluidity factor (b) Improvement in saturation point injection rate with respect to FVC technique.

FL. We compute the *Buffer Fluidity Coefficient (BFC)* for a VC buffer in a router at the end of the simulation. It is computed as a summation of *FL* value across all clock cycles as per the following equation:

$$BFC = \sum_{t} FL$$

The *BFC* of a router is computed as the average of *BFC* of each of its VC buffer. It is a representation of average buffer fluidity level within a router. We compute the *Buffer Fluidity Fairness (BFF)* at the end of the simulation by using the following equation:

$$BFF = \frac{1}{Standard\ deviation\ of\ BFC\ per\ router}$$

If the *BFC* of routers do not deviate much from the average *BFC* of the network, the standard deviation of *BFC* reduces. If the standard deviation of *BFC* across all routers reduces, the fairness factor improves. Higher the fairness factor, better is the balancing of buffer fluidity. Figure 4(a) shows the comparison of buffer fluidity fairness of BOFAR and other baseline architectures at saturation loads. The average increase in buffer fluidity fairness of BOFAR is 29.47%, 20.68%, and 28.67% with respect to FVC, NOP and FON, respectively. The effect is more visible in bitcompliment traffic because the number of possible routes between source destination pair is more and hence BOFAR equally balances the usage of all paths. Thus BOFAR tracks the fluidity levels and regulates the flit flow, and makes sure that every router is in a safe fluid level.

4.3 Evaluation of Average Saturation Point

Figure 4(b) shows the improvement on average saturation point of different selection strategies with respect to FVC technique. BOFAR achieves an increase in saturation injection rate by an average of 2.05% where as NOP and FON achieves only 1.77% and 0.67%, respectively. This shows that BOFAR is able to handle more flits and route them to destination without saturating the network thereby making BOFAR a good choice for high injection rate applications.

4.4 Power and Area Overhead

Using the Predictive Technology Model [23], we find that signal propagation through inter-router link takes one cycle at 1GHz clock frequency. Since we fix our link traversal at one cycle for latency calculations, we assume a 1 GHz operating frequency at 65 nm technology. An 8-bit *BOC* is added per router port. We need an 8-bit adder to add *CSB* value to *BOC* in every cycle. At the receiving end of a port we need a 24-bit *SR* to collect the *BOC* value passed by a neighboring router. The control channel which propagates the *BOC* value is of 24-bit for internal channels and 8-bit for edge channels.

Table 1: Overhead comparison of various selection techniques with respect to baseline MOE router.

Technique	Power Overhead (%)	Area overhead (%)
FVC	1.02	2.14
NOP	3.28	6.16
FON	4.34	8.68
BOF	5.58	10.26

We use Orion 2.0 [13] to compute area and power estimates. Since all the baseline architectures use some sort of calculation and comparision of congestion values, power overhead of the combinational circuits that do these operations are almost same in all baseline routers. Significant hardware overhead difference comes in the case of width of control network and size of counters used. Our power and area overheads comparison are focusing on these two parameters only. All our power and area comparison are with respect to MOE router with 128-bit flit channel and 4-bit control channel. Table 1 shows the area and power overheads of various baseline architectures and BOFAR. The table entries are computed by adding both router and link parameters. A major share of the overhead in BOFAR is due to the wider control network and counters used for tracking and propagating the *BOC* values.

5. CONCLUSION

We proposed an adaptive router with a port selection strategy based on the buffer occupancy factor of flits. By using this approach we effectively reduced the rate of flow of flits to non-fluid nodes. Thus fluid routers were able to get more flits and non-fluid routers less flits. If due to adverse traffic patterns the fluidity balance is broken, our approach will ensure that equilibrium is maintained by effective regulation of flit movement. The light weight monitoring logic and minimal extra control network ensured that the power and area overhead in the proposed design is negligible compared to the gain obtained by latency reduction and increased buffer fluidity fairness. For 16-core network, in all

five synthetic workloads and the bursty traffic, our design showed lower average packet latency. The performance in bursty traffic emphasizes the fact that BOFAR is best suited for traffic which exhibits irregular flit injection rates. Moreover our model was able to extend the saturation point of the network. From our study we propose that past history of buffer occupancy can be used as an effective congestion metric in future NoC router designs. Even though our results in 8x8 mesh are promising, we still need to come up with a better optimal design point that keeps power and area overhead within acceptable limits. As a future enhancement, we would like to extend this model to operate on larger packet sizes and larger networks under varying set of VCs.

6. ACKNOWLEDGMENTS

This work is supported in part by grant from Defence Research and Development Organization (DRDO) under IITM-DRDO MoC.

7. REFERENCES

[1] A. Agarwal, B. Raton, C. Iskander, and R. Shankar. Survey of network-on-chip architectures and contributions. *Journal of Engineering, Computing, and Architecture*, 3(1):1–15, 2009.

[2] G. Ascia, V. Catania, M. Palesi, and D. Patti. Implementation and analysis of a new selection strategy for adaptive routing in networks-on-chip. *IEEE Transactions on Computers*, 57(6):809–820, 2008.

[3] T. Bjerregaard and S. Mahadevan. Survey of research and practices of network-on-chip. *ACM Computing Surveys*, 38(1):11–51, March 2006.

[4] G. M. Chiu. The odd-even turn model for adaptive routing. *IEEE Transactions on Parallel and Distributed Systems*, 11(7):729–738, July 2000.

[5] M. H. Cho, M. Lis, K. S. Shim, M. Kinsy, and S. Devadas. Path-Based, Randomized, Oblivious, Minimal Routing. In *NoCArc-'09: Proceedings of the 2nd International Workshop on Networks-on-Chip Architectures*, pages 23–28, December 2009.

[6] N. Concer, L. Bononi, M. Souli, R. Locatelli, and L. P. Carloni. CTC: An end-to-end flow control protocol for multi-core systems-on-chip. In *NOCS-'09: Proceedings of 3rd International Symposium on Networks-on-Chip*, pages 193–202, May 2009.

[7] W. Dally. Virtual-channel flow control. *IEEE Transactions on Parallel and Distributed Systems*, 3(2):194–205, March 1992.

[8] W. Dally and B. Towles. Route packets, not wires: On-chip interconnection networks. In *DAC-'01: Proceedings of the Design Automation Conference*, pages 684–689, 2001.

[9] W. Dally and B. Towles. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.

[10] C. J. Glass and L. M. Ni. The turn model for adaptive routing. In *ISCA-'92: Proceedings of the 19th Annual International Symposium on Computer Architecture*, pages 278–287, May 1992.

[11] P. Gratz, B. Grot, and S. W. Keckler. Regional congestion awareness for load balance in networks-on-chip. In *HPCA-'08: Proceedings of the International Symposium on High Performance Computer Architecture*, pages 203–214, February 2008.

[12] J. Hu and R. Marculescu. DyAD: Smart routing for networks-on-chip. In *DAC-'04: Proceedings of the Design Automation Conference*, pages 260–264, 2004.

[13] A. B. Kahng, L. Bin, L.-S. Peh, and K. Samadi. Orion 2.0: A fast and accurate NoC power and area model for early stage design space exploration. In *DATE-'09: Proceedings of the Design, Automation and Test in Europe Conference*, pages 423–429, 2009.

[14] A. E. Kiasari, A. Jantsch, and Z. Lu. A framework for designing congestion-aware deterministic routing. In *NoCArc-'10: Proceedings of the 3rd International Workshop on Networks-on-Chip Architectures*, pages 45–50, December 2010.

[15] J. Kim, D. Park, T. Theocharides, and N. Vijaykrishnan. A low latency router supporting adaptivity for on-chip interconnects. In *DAC-'05: Proceedings of the Design Automation Conference*, pages 559–564, 2005.

[16] Y. C. Lan, M. C. Chen, A. P. Su, Y. H. Hu, and S. J. Chen. Fluidity concept for NoC: A congestion avoidance and relief routing scheme. In *SoC-'08: Proceedings of 21st Annual IEEE International SoC Conference*, pages 65–70, November 2008.

[17] R. Marculescu, U. Y. Ogras, L. S. Peh, N. E. Jerger, and Y. Hoskote. Outstanding research problems in NoC design: System, microarchitecture, and circuit perspectives. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 28:3–21, January 2009.

[18] C. Nicopoulos, S. Srinivasan, A. Yanamandra, D. Park, and V. Narayanan. On the effects of process variation in network-on-chip architectures. *IEEE Transactions on Dependable and Secure Computing*, 7(3):240–254, September 2010.

[19] E. Nilsson, M. Millberg, J. Oberg, and R. Robin. Load distribution with the proximity congestion awareness in a network-on-chip. In *DATE-'03: Proceedings of the Design, Automation and Test in Europe Conference*, pages 1126–1127, 2003.

[20] D. Sanchez, G. Michelogiannakis, and C. Kozyrakis. An analysis of on-chip interconnection networks for large scale chip multiprocessors. *ACM Transactions on Architecture and Code Optimization*, 7(1):4:1–4:28, April 2010.

[21] M. Sheng, N. E. Jerger, and Z. Wang. DBAR: An Efficient Routing Algorithm to Support Multiple Concurrent Applications in Networks-on-Chip. In *ISCA-'11: Proceedings of Annual International Symposium on Computer Architecture*, pages 413–424, June 2011.

[22] T. Tao, L. Benini, and D. Micheli. Packetization and routing analysis of on-chip multiprocessor networks. *Journal of Systems Architecture*, 50:81–104, February 2004.

[23] W. Zhao and Y. Cao. Predictive technology model for nano-CMOS design exploration. *ACM Journal on Emerging Technologies in Computing Systems*, 3:1–17, April 2007.