TRACKER: A Low Overhead Adaptive NoC Router with Load Balancing Selection Strategy

John Jose, K.V. Mahathi, J. Shiva Shankar and Madhu Mutyam

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

ABSTRACT

The effectiveness of an adaptive router in a Network on Chip (NoC) is evaluated by the selection metric it uses and its impact on overall performance. In this paper, we propose a flit flow history based load balancing selection strategy that can be used in any adaptive routers for output port selection. Using this selection strategy, we propose an adaptive router *TRACKER*, that keeps track of flow of flits through all its ports and updates this tracked information to its neighbors in a cost effective manner. Routers make use of these flit flow estimates to compute a novel selection metric for output port selection of incoming flits. *TRACKER* outperforms the baseline adaptive router architectures using odd-even routing model with conventional selection metrics like count of free virtual channels, count of fluid buffers and buffer occupancy time at reachable downstream neighbors.

Categories and Subject Descriptors

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

Keywords

NoC, adaptive routers, selection metric, load balancing

1. INTRODUCTION

Over the last decade, technological innovations in SoCs have been focusing on empowering the computational efficiency of computing cores. Apart from high speed computing cores, efficient and reliable communication is also essential for achieving high performance and throughput in multicore processors. Diminishing feature sizes and shrinking wire widths amplified the imbalance between gate delays and on-chip wire delays [1]. NoC architectures are proposed in multicore systems to replace design specific global on-chip wiring with general purpose on-chip interconnection network. Instead of dedicated point-to-point bus based communication, NoCs employ a grid of routers organized across the chip, connected by communication links [1, 2].

IEEE/ACM International Conference on Computer-Aided Design (ICCAD)
November 5-8, 2012, San Jose, California, USA

In a 2D mesh topology, the processing cores are organized as rectangular tiles. Each core is attached to a local router which connects the core to the neighboring cores through a well structured set of routers and links. When a core wants to communicate with another, it creates a packet containing the required data. In NoCs with wormhole switching, each packet is serialized into a sequence of flow control units called flits. The head flit contains the necessary control information needed to process and route the packet. Flits move across the routers in a hop by hop manner.

A baseline router in a 2D mesh topology consists of five input ports and five output ports; one for each direction and one for the local tile. A 5×5 crossbar manages the inter-router connections [2]. Every input port of a router is associated with a set of flit buffers called virtual channels (VCs). The use of VCs reduces the average network latency of a flit at the expense of area and power consumption [4]. Once a packet reaches a router, the VCs provide storage space for it. The control logic in the router performs a set of necessary services on the packet such as determination of next hop route, arbitration and allocation of VC in the next router, allocation and traversal of the crossbar switch. The credit signals to and from a router carry the availability of buffer space information. Flits residing in various VCs on the same input port arbitrate among themselves and winning flits from various input ports will undergo switch arbitration and allocation and then the flits are forwarded through the crossbar switch to respective output ports.

Buffering within the routers and handshaking between routers enable the smooth flow of the packets. Thus the communication among various cores is achieved by generating, processing, and forwarding data packets and control signals through the network infrastructure. Adaptive routers choose the best route for incoming packets from a set of available paths by employing a proper selection metric that captures the dynamically varying network congestion status [2].

We propose a new selection metric based on flit flow analysis that could be used with any adaptive routers. Our key focus is to enhance the link utilization of the network by balancing the traffic flow across all links. Instead of using the conventional metrics like availability of free VCs [4, 5], buffer fluidity values [9] and buffer occupany values [10] across downstream nodes, we introduce a new selection metric, *cumulative flit count* in the past, on the possible future links. Routing decisions are taken in such a way that less frequently used links are preferred. Our proposed adaptive router *TRACKER* has no impact on its critical path, with negligible impact on router area and power.

2. SCOPE OF SELECTION STRATEGY

High performance NoC designs demand low latency, load balancing and deadlock-free adaptive routers. Physical limitations in the inter-router link capacity and count of VC-buffers lead to resource contention at higher packet injection rates. When an adaptive router identifies more than one possible output port for an incoming flit, the output port selection function chooses one of these output ports for the flit by using an appropriate metric that captures congestion [5]. The effectiveness of a selection strategy depends on the choice of a metric used to represent the congestion and the accuracy level of the metric in representing the real magnitude of congestion.

Our preliminary study on various synthetic traffic patterns on 4×4 mesh employing the odd-even adaptive routing shows that, on an average, in 27% of the cases the routing function returned multiple admissible output ports. Thus, at least one for every four routing decisions, the selection strategy determines which output port is to be selected. The percentage value is even higher for larger mesh networks. A well formed selection strategy can have a significant impact in choosing the best possible path for a packet thereby increasing the throughput of the system.

3. RELATED WORK

The *minimal odd-even routing* (MOE) [3] is one of the simple and the most commonly used deadlock free adaptive routing algorithms used in mesh NoCs. The MOE routing by itself does not use any output port selection and hence it makes a random selection from available ports. To enhance the adaptivity, selection functions are employed on top of the MOE routing. *Proximity Congestion Awareness* [6] makes use of the load information of neighboring switches for channel selection decisions. *Path-Based Randomized Oblivious Minimal Routing* [8] proposes a load balancing routing scheme by random channel selection. *Congestion Aware Deterministic Routing* [7] estimates the congestion level in the network based on past flow pattern and computes optimized routing paths for all traffic flows deterministically. This is suited only for systems which generate repetitive traffic patterns.

The count of free VCs (FVC) in the adjacent downstream router is also explored as a selection metric [4, 11]. The Neighbors-on-Path (NOP) [5] strategy explores the free VC status of reachable neighbors of adjacent routers of current node. The fluidity based approach [9] explores flit movement by counting the number of fluid buffers. Forwarding a flit to a router with more fluid buffers facilitates easy flow of flits through them. During congestion, if desired output port is not available flits occupy buffers for longer time. $BOFAR$ [10] tries to capture the history of buffer occupancy over a reasonable time interval.

4. MOTIVATION

Majority of the existing adaptive routers [4, 5, 11] use availability of free VC buffers across downstream nodes as the selection metric. Even though this sounds to be a simple and meaningful approach, careful analysis exposes its inefficiency. 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 buffers on it or its downstream routers. At higher injection rates close to saturation, since almost all VCs in routers are full, the effective value of the selection metric is close to zero. When there is a tie in the value of the captured metric, a random port from the candidate ports is chosen, which makes the channel selection strategy itself meaningless.

In fluidity approach [9] a buffer is said to be fluid when a flit is moving out from it in the current clock cycle. Nodes with more fluid VCs are treated as less congestion prone. This is a more realistic representation of congestion than a FVC or NOP metric. One of the drawbacks of fluidity based output selection is its inability to distinguish the level of congestion if both neighbors of a node are either equally fluid or equally non-fluid.

Congestion cannot be defined only based on the feedback data obtained from the current and the previous clock cycle updates. Congestion is formed over a period of time due to cumulative and chain reaction effects. Unfortunately none of the baseline architectures except $BOFAR$ [10] captures this. Cumulative buffer occupancy time proves to be more realistic selection metric than the instantaneous count of free buffers and the fluidity information of downstream nodes. But capturing, computing, and propagating the cumulative buffer occupancy time as mentioned in [10] involves significant hardware and wiring overhead.

If links are not fairly and evenly used, it leads to uneven flow distribution, making certain paths heavily used leading to resource contention and delay in packet movement. Congestion can also be formed due to unregulated traffic flow. None of the baseline models explores the relation of congestion with the link utilization. Existing selection metrics are not addressing congestion from this angle. Uniform usage of links ensure the balanced flow across all possible paths and prevent premature aging of links. With a proper feedback mechanism that captures flit flow rate, we can reduce congestion by regulating flit load across the links. Through this paper we make an effort to realize this.

5. TRACKER ROUTER ARCHITECTURE

$TRACKER$ is a typical VC router [2] which keeps track of flow of flits through all its outgoing ports and exchanges this flit flow information with its neighbors. It makes use of this flit flow information to compute the proposed selection metric, i.e., *cumulative flit count* (CFC). CFC indicates the contention level of an output port of neighboring router. Higher the CFC, higher the number of flits passed through that output port in the recent past. When there is a choice in selecting an output port, we opt for an output port which has the least CFC. Processing of CFC values is done in parallel with the route computation. Control signals for VC allocation and switch allocation are generated to facilitate the flit forwarding along the chosen path.

The architecture of the selection logic of a $TRACKER$ router (router 5 as per Figure 1) is shown in Figure 2. The figure emphasizes the interaction and control wiring associated with west neighbor of router 5. For each router, a flit tracking circuit and a flipflop (FF) are attached on each output port. At the end of every clock cycle, the flit tracking circuit updates its FF. FF of a port is set if a flit flows through the port, else it is reset. Special control wires connecting a router and its neighbors (shown as 3-bit value to/from west port of the router) carry the FF states. Once a router receives the FF values, the CFC kept in the corresponding 4-bit *Status Register* (SR) segment is updated by adding the FF values to them.

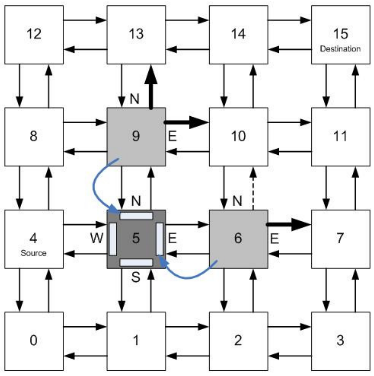

Figure 1: Illustration of Usage of Selection Metric in TRACKER Router.

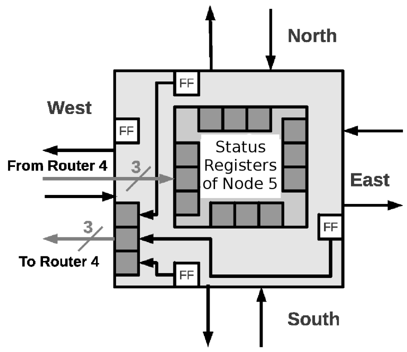

Figure 2: Internal Architecture of Selection Logic in TRACKER.

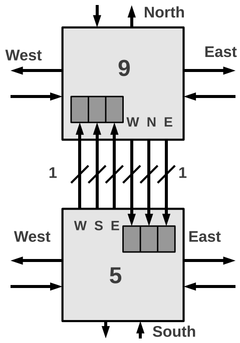

Figure 3: Status Registers and Control Network in TRACKER.

After every T clock cycles, called the *Refresh Interval* (RI), SR value is multiplied with a weight factor of α=0.25 to prevent the overflow of SR and to preserve the history of accumulated CFC. Even though the best value of α is not 0.25, a floating-point multiplier can be replaced with a simple shifter if the value of α is 0.25. Hence to realize the multiplication of α and incrementing the SR, a shifter and an incrementer circuit are attached to each segment of SR. After exploring with various values of RI, we consider the best performance-power combination with RI= 16 cycles and conduct the experimental analysis.

Now let us examine the working of flit forwarding in a $TRACKER$ router. As per Figure 1, assume a flit F, sourced at node 4 and destined to node 15, reaches node 5. The MOE route function [3] chooses east port (link to node 6) and north port (link to node 9) as possible output ports for the flit F. In the meantime FF state values from node 9 and node 6 reach node 5. From the FF states the corresponding CFC values are updated in the respective segments of SR. A flit from node 5 destined to node 15, upon reaching node 9 has two possible output ports, north and east ports (shown by thick arrows) of node 9. Hence the *mean-CFC* for F along the north port of node 5 is the average of CFC through east and north ports of node 9. Similarly the *mean-CFC* for F along the east port of node 5 is computed. In this case, the north port of node 6 is not a reachable port (shown by dotted arrow) for a flit coming from node 5 due to MOE turn restriction [3]. The output port of node 5 with smaller *mean-CFC* is given higher priority at port selection for F. For every incoming flit, $TRACKER$ chooses the link with the minimum traffic in the past, thereby ensuring the load balance across the links.

Every router forwards at most three FF states to a neighbor. For example (as per Figure 1), node 9 tracks the flit flow through all its four outgoing ports (connected to nodes 13, 8, 5, and 10) and updates the respective FF states, but only FF state of ports connected to nodes 13, 8, and 10 are forwarded to node 5 and that of nodes 8, 5, and 13 are forwarded to node 10. Thus in every clock cycle, each node gets the FF state associated with reachable ports of its neighbor. Figure 3 shows the control network that carries FF values between node 5 and node 9. Each port of a router has a SR of size $4n$ bits, where n is the number of segments in the SR. This n is same as the number of FF values a router receives from its neighbor associated with that port.

Port selection by CFC is not affected by the status of the available VCs on the downstream router. If a node along the port chosen by the *mean-CFC* comparison is not having any free VCs, the flit stays back in the current router in the present clock cycle and makes an attempt in the next cycle. If we choose *mean-CFC* only for neighbors with available VCs, an extra cycle delay may be avoided in the current router. But results show that such greedy local decisions increase the overall packet latency and decrease the link utilization fairness.

6. EXPERIMENTAL ANALYSIS

We now compare the performance of MOE adaptive router using the proposed selection metric and other baseline selection metrics discussed in Section 3. We also examine the sensitivity of the network to various design parameters considered in the TRACKER router architecture.

6.1 Experimental Setup

We use *Booksim* [2], a cycle accurate network simulator, that models a two-cycle router micro-architecture in sufficient detail. The simulator is modified to model an adaptive router that uses MOE routing. Performance values are collected with different selection metrics: free VCs in downstream nodes (FVC); free VCs in reachable nodes at two-hop distance (NOP); count of fluid VCs on reachable nodes at two-hop distance (FON); cumulative buffer occupancy time of flits that move out through reachable output ports of downstream nodes (BOF); and our proposed metric, cumulative flit count through reachable output ports of downstream nodes (CFC).

We evaluate our proposed technique using five standard synthetic traffic patterns: *uniform*, *tornado*, *bit-compliment*, *transpose*, and *bit-reverse* for 4×4 mesh networks. Since *bit-compliment*, *bit-reverse*, and *transpose* traffics are not defined for 5×5 mesh, we use only *uniform* and *tornado* traffics for evaluating 5×5 mesh networks. Average packet latency and link utilization values are collected for different traffic patterns under various injection rates. We use 4 and 8 VCs per router port in 4×4 and 5×5 mesh networks, respectively. We consider variable length packets with 3-flit buffer per VC. We consider 60% single-flit packets and the rest larger packets of size up to four flits. α for computation of CFC is fixed at 0.25 and RI is set to 16 cycles. BOF design is done with RI of 128 cycles.

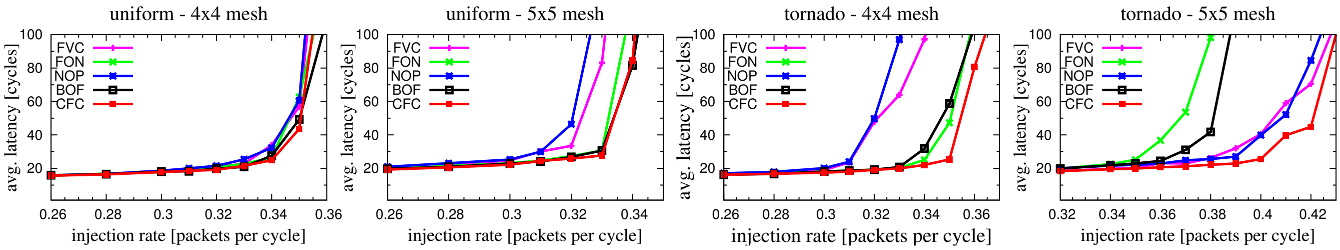

Figure 4: Average packet latency for *uniform* and *tornado* traffic patterns (packet size ranging from 1 flit to 4 flits).

6.2 Evaluation of Average Packet Latency

Figure 4 contains a set of load-latency graphs for CFC and other selection metrics with variable sized packets. In all the plots we can see that, at low injection rates, almost all the selection metrics experience the same average packet latency. This is because of the fact that the network is free from congestion. But as load in the network increases, the effect of selection metric adopted becomes more visible. Similarly the plots clearly show that CFC metric extends the injection rate at which the network saturates. This makes $TRACKER$ the best design choice for high injection rate applications.

Even though we plot the load latency results only for two synthetic traffic patterns, we analyzed the performance of various metrics for *bit-compliment*, *transpose*, and *bit-reverse* traffic patterns also. On 4×4 mesh networks with variable size packets, at saturation load CFC achieves an average latency reduction of 43%, 17%, 41%, and 21% with respect to FVC, FON, NOP, and BOF, respectively. Our technique extends the saturation injection rate by up to 1.5% in *tornado* and *bit-compliment* traffic. The effect of CFC is more predominant in these two patterns as the source and destination are diagonally oriented and the possibility of having multiple paths to spread the traffic is high.

In load latency plot for 5×5 mesh networks (Figure 4), we can see that CFC metric has the least pre-saturation latency. As NOP metric makes the network saturate much faster than other metrics, it is excluded for comparison. At saturation load, the average packet latency reduction of CFC is 33%, 52%, 35%, respectively, over FVC, FON, and BOF techniques. Performance gap between CFC and the other techniques increases with mesh size, thereby exposing the ineffectiveness of other techniques in capturing the real congestion in larger networks. Even though the latency plot of FON and BOF are close to that of CFC in certain cases, they fail to perform consistently across all traffic patterns. Consistent performance of $TRACKER$ across all traffics and all network sizes makes CFC an excellent metric for selection functions in adaptive routers.

6.3 Evaluation of Link Utilization Fairness

Another observation we made is the increase of Link Utilization Fairness (LUF) by using CFC as the selection metric. We compute the total number of flits flowing through each link for each simulation instance (traffic pattern, injection rate) and then compute the LUF as follows:

$$LUF = \frac{\text{Average number of flits per link}}{\text{Standard deviation of flits per link}}$$

If the standard deviation of flits per links reduces, the LUF improves. Higher the LUF, better is the load distribution across links. We compute the ratio of average packet latency to the corresponding LUF for various injection rates under varying traffic patterns and network size. Lower the ratio, more effective the metric is. Figure 5 shows that CFC metric is consistently having the least latency to LUF ratio for all network sizes.

At higher loads, when more flits are injected into the network, $TRACKER$ exhibits better LUF. Results in larger networks show that $TRACKER$ scales well. The combination of reduced average packet latency, increased saturation injection rate along with increased LUF makes CFC metric as a promising alternative for conventional selection metrics. Increase in LUF leads to uniform wear and tear and hence can increase the lifetime of links.

6.4 Sensitivity to Various Design Parameters

We also examine the sensitivity of the network to various design parameters considered in the $TRACKER$ design. A key design factor in computing CFC is the weight factor α which determines how significant is the flit flow history through a link in the previous RI period. For real representation of congestion, flit flow updates in the current RI period should be given dominance over cumulative past flit flow estimates. If α is 0, then it results in loss of flit flow history at the beginning of a new RI period. In such cases CFC cannot represent the real congestion in the initial few cycles of an RI period, leading to poor port selection decisions.

Upon experimenting with different values of α ranging from 0.05 to 0.45, we find that different traffic patterns require different value of α to give best average packet latency. Even though we compromise on average packet latency and saturation throughput, we use $\alpha = 0.25$ to keep router power and area within acceptable limits.

Another parameter whose effect is explored is the RI for SRs. High RI requires wide SRs, to keep CFC value without saturating, which incurs more power and area overhead. For low RI values, CFC value is insufficient to distinguish between candidate paths for a flit in a given router. These factors prompt us to choose an RI value large enough to capture the variations of flit flow and small enough for minimum design overhead. Considering these factors we fix the RI value as 16 cycles.

We study the effect of the *compute interval* (CI) of CFC metric. We examine the results by reducing the frequency of updation of CFC to once in 4, 8, 16, and 32 cycles under varying RI. But results show that it leads to degradation of LUF since a constant value of CFC for few cycles sets the priority of output ports of a router to be constant within this CI. So during this interval all flits destined to a quadrant move through one path only, which disturbs LUF. All the results presented in this paper are based on CI of 1 cycle.

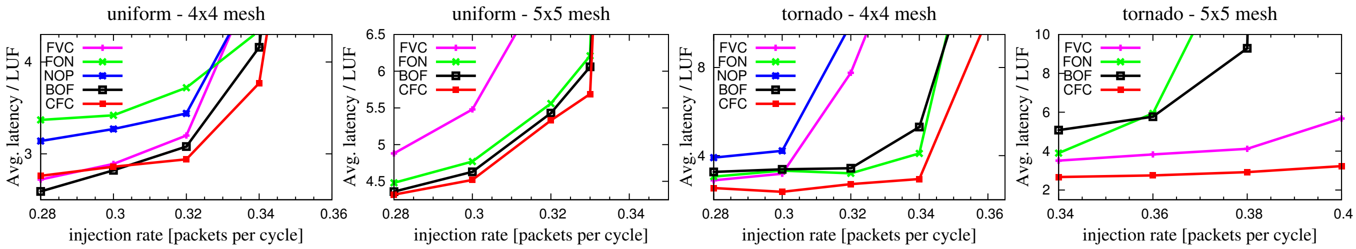

Figure 5: Ratio of Latency to LUF for various network sizes under uniform and tornado traffic patterns.

Table 1: Comparison of width of SRs and control network for various selection metrics.

Metric	Width of SR	Width of Control Network
FVC	4	4
FON	16	16
NOP	12	12
BOF	24	24
CFC	12	3

Table 2: Overhead comparison of various selection metrics with respect to FVC selection metric.

Metric	Router Power Overhead (%)	Link Area Overhead (%)
FON	5.44	9.03
NOP	3.57	6.02
BOF	9.3	12.05
CFC	3.59	1.50

6.5 Hardware Overhead

We use Orion 2.0 [12] for area and power computation of the baseline routers and *TRACKER*. We assume 65 *nm* technology at 1 *GHz* operating frequency and NoC channel delay of one cycle [13]. Since all the baseline architectures use some sort of combinational circuits for comparison of selection metric values across candidate paths, significant hardware overhead difference between these techniques is due to width of control network and size of status registers used. Our power overhead comparison is focusing on these two parameters only. Apart from a standard credit channel, all the selection metrics require some additional control wiring to transmit congestion information. Table 1 shows the difference in width of control network and size of SRs. A general comparative study on the overheads associated with various selection metrics is given in Table 2. All power and area overhead values in the table are with respect to MOE router with 128-bit flit channel and 4-bit credit channel using *FVC* selection metric.

7. CONCLUSION

An adaptive router based on the flit flow analysis of downstream router is proposed. Our model makes the best use of the link bandwidth and spread traffic across less frequently used links to balance the load. *TRACKER*, without much additional overhead, gathers non-local flit flow history from neighbors and route flits accordingly. The light weight monitoring logic and minimal extra control network ensure that the power and area overhead in the proposed design is negligible compared to the latency reduction and increased link utilization achieved. In larger networks, across all traffic patterns examined *TRACKER* has the least average packet latency in pre-saturation loads. Our experimental results showed that cumulative history of flit flow rate can be used as an effective selection metric in future NoC router designs.

8. ACKNOWLEDGMENT

This work is supported in part by grant from Defence Research and Development Organization (DRDO) under IITM-DRDO MoC. Travel support from the Indian Institute of Technology Madras and Indian Association for Research in Computer Science (IARCS) is greatly acknowledged.

9. REFERENCES

[1] W. Dally and B. Towles, Route packets, not wires: On-chip interconnection networks, in DAC, pp. 684–689, 2001.
[2] W. Dally and B. Towles, Principles and Practices of Interconnection Networks. Morgan Kaufmann Publishers Inc., USA, 2003.
[3] G. M. Chiu, The odd-even turn model for adaptive routing, IEEE TPDS, vol. 11, no. 7, pp. 729–738, 2000.
[4] W. Dally, Virtual-channel flow control, IEEE TPDS, vol. 3, no. 2, pp. 194–205, 1992.
[5] G. Ascia, et al., Implementation and analysis of a new selection strategy for adaptive routing in NoC, IEEE TOC, vol. 57, no. 6, pp. 809–820, 2008.
[6] E. Nilsson, et al., Load distribution with the proximity congestion awareness in a network-on-chip, in DATE, pp. 1126–1127, 2003.
[7] A. E. Kiasari, et al., A framework for designing congestion-aware deterministic routing, in NoCArc, pp. 45–50, 2010.
[8] M.H.Cho, et al.,Path-based, randomized, oblivious, minimal routing, in NoCArc, pp. 23–28, 2009.
[9] Y. C. Lan, et al., Fluidity concept for NoC: A congestion avoidance and relief routing scheme, in SoC, pp. 65–70, 2008.
[10] J. Jose, et al., BOFAR : Buffer occupancy factor based adaptive router for mesh NoCs, in NoCArc, pp. 23–28, 2011.
[11] J. Kim, et al., A low latency router supporting adaptivity for on-chip interconnects, in DAC, pp. 559–564, 2005.
[12] A. B. Kahng, et al., Orion 2.0: A fast and accurate NoC power and area model for early stage design space exploration, in DATE, pp. 423–429, 2009.
[13] W. Zhao and Y. Cao, Predictive technology model for nano-CMOS design exploration, ACM JETCS, vol. 3, pp. 1–17, 2007.