Microprocessors and Microsystems 38 (2014) 304–315

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

# Design and evaluation of a high throughput QoS-aware and congestion-aware router architecture for Network-on-Chip

Chifeng Wang *, Nader Bagherzadeh

*Dept. of Electrical Engineering and Computer Science, University of California, Irvine, Irvine, CA 92697, USA*

ARTICLE INFO

*Article history:*
Available online 27 September 2013

*Keywords:*
Network-on-Chip (NoC)
Interconnection network
Congestion-aware
Quality-of-Service (QoS)

ABSTRACT

This paper proposes a novel QoS-aware and congestion-aware Network-on-Chip architecture that not only enables quality-oriented network transmission and maintains a feasible implementation cost but also well balance traffic load inside the network to enhance overall throughput. By differentiating application traffic into different service classes, bandwidth allocation is managed accordingly to fulfill QoS requirements. Incorporating with congestion control scheme which consists of dynamic arbitration and adaptive routing path selection, high priority traffic is directed to less congested areas and is given preference to available resources. Simulation results show that average latency of high priority and overall traffic is improved dramatically for various traffic patterns. Cost evaluation results also show that the proposed router architecture requires negligible cost overhead but provides better performance for both advanced mesh NoC platforms.



## 1. Introduction

Multi-core integrated circuit designs have been proposed and proven a prevailing architecture recently. Multiprocessors (CMPs) such as 64-core SoC and 80-core NoC architecture [6,42] were presented to pave the way to network-based interconnection network design. NoC interconnection scheme has been demonstrated as a better solution because of superior performance and fault tolerance characteristics [7,11]. NoC interconnection architecture uses a distributed control mechanism, providing a scalable interconnection network.

A multiprocessor system platform called Network-based Processor Array (NePA) has recently developed [5], in which processors are interconnected by using an on-chip two-dimensional (2D) mesh network. NePA is a deadlock-free and livelock-free network that implements wormhole packet switching technique and utilizes an adaptive minimal routing algorithm. Because of the limitation of traditional 2D mesh topology, additional alternative routing resources which provide more network tolerance are employed to further improve the performance of the NePA architecture. Moreover, diagonal links for the 2D mesh network are proposed to improve throughput and performance because of the emergence of X-architecture routing technique in chip manufacturing [21,39]. The proposed NoC architecture referred to as Diagonally-linked Mesh (DMesh) employs diagonal express links among routers on a baseline NePA network [20]. Diagonal links not only reduce the distance between source and destination nodes, but also help to alleviate network congestion so that network performance is enhanced dramatically [47].

Adaptive routing algorithms have been employed in multichip interconnection networks as means to improve network performance and to tolerate link or router failures. However, congestion in interconnection networks is a well-known phenomenon. This work utilized a congestion control scheme to provide adaptive routing arbitration control and thus exploit available routing resources efficiently. This allows avoiding control beyond target baseline adaptive routing algorithm which can adaptively balance traffic load and increase NoC overall throughput by intelligently allocating existing resources. Quality-of-Service (QoS) provision supports differentiated service classes among various applications and can further improve utilization efficiency of network bandwidth. Instead of original approaches of adding multiple buffers or virtual channels (VCs), the proposed mechanism featuring congestion avoidance scheme and QoS ability facilitates differentiated service transmission while maintaining a high throughput tolerance.

The rest of the paper is organized as follows. Section 2 summarizes related work in on-chip interconnection networks. Section 3 describes an overview of NePA and DMesh NoC platforms. Section 4 proposes an innovative router design with congestion-aware and QoS-aware router scheme. Section 5 shows performance and cost evaluation of the proposed architecture. Concluding remarks are provided in Section 6.

* Corresponding author.
*E-mail addresses:* chifengw@uci.edu (C. Wang), nader@uci.edu (N. Bagherzadeh).

0141-9331/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.micpro.2013.09.006

## 2. Related work

### 2.1. Congestion control

Congestion management was proposed to prevent networks from saturation and improve the throughput in NoC. Buffer and link status are two of the most popular ways to indicate the existence of network congestion [37,40,41]. A congestion-aware routing algorithm is targeted to evenly distribute traffic load over the network. For instance, a self-optimized routing strategy [40] decides a favorable path for incoming packets based on buffer load information. A proximity congestion awareness technique is proposed to avoid congested areas based on the use of stress values which are passed from neighboring switches [35]. Both techniques attempt to divert packets from hot spots in the network. A contention-aware input selection algorithm which gives priority to incoming packets from congested areas was proposed to alleviate congestion in upstream area [48]. An application-aware congestion control algorithm in on-chip bufferless networks is proposed by making proper throttling decisions [36]. A streamlined method is used in network interfaces to reduce the network congestion by global congestion information. An adaptive scheduler is equipped to reduce additional traffic to congested areas [14]. An approach using combination of local and non-local network information to determine the optimal path to forward a packet was recently proposed [16,17,27].

Adaptive input–output selection router architecture which utilizes an adaptive minimal and non-minimal routing algorithm relying on the congestion condition of neighboring routers to circumvent the congested areas was proposed [12,15]. Weighted round robin arbitration mechanism is adopted in input selection to give preference to input port with light traffic in order to avoid possible network congestion in the downstream routers. An adaptive output-selection method using both minimal and non-minimal paths based on the congestion condition of neighboring routers was introduced. Congestion flag is asserted when buffer size reaches pre-defined threshold and increasing buffer occupation rate is detected.

In order to design a lightweight congestion-aware NoC router, an approach based on dynamic port arbitration to resolve congestion and adaptive output path selection to distribute traffic load efficiently was devised [45,46]. Complicated buffer monitor methods will cause more overhead for multi-port designs such as NePA and DMesh routers. Different from buffer occupation measurement, the number of blocked buffers meaning active FIFOs with waiting packets serves as congestion index and this approach lowers the congestion measurement complexity. Upstream traffic from more congested routers is given preference to direct packets toward light-congested destination. The proposed router can effectively enhance network throughput by utilizing simple congestion index statistics to determine the network congestion status so that implementation complexity can be reduced. This work utilized this methodology to relax congestion situation.

### 2.2. QoS provision

QoS is commonly achieved by providing each traffic class with a separate virtual channel, either in a time-division multiplexing [19,32] or dynamic virtual channel allocation [23,24] manner. Networks providing guaranteed throughput (GT) and best effort (BE) services use VC reservation methodology. The Æthereal is a NoC that provides GT that is connection-oriented and BE that uses non-reserved time slots, and it supports statical and dynamic allocation of slots [18]. The MANGO [8] is another NoC that provides connectionless BE routing and connection-oriented guaranteed services (GS). Another implementation example adopts deterministic dimension order source routing strategy and assigns priority to GT traffic. BE packets are allocated in a round robin manner if GT associated virtual channels (VCs) are empty [44]. A customized QoS NoC (QNoC) which classifies service into four classes: signaling; real-time; RD/WR and block transfer was proposed. There are individual buffers to store different classes of traffic and bandwidth are allocated accordingly [9]. A memory-efficient on-chip network architecture with intelligent memory controller inside network interface was proposed to realize QoS by better resource utilization and reordering mechanism, including out-of-order delivery and a priority-based router to decrease the network latency [13].

Although multiple VCs are implemented to support more service levels, these designs increase switch complexity and arbitration delay. An area-efficient design using two VCs at switches was presented to provide full QoS support, which demonstrates a more than acceptable performance and meets the low cost need of NoC stringent implementation requirement [30,28]. The trend of providing sufficient adaptive routing or fault tolerance is to increase the number of ports instead of increasing the number of VCs per port was also indicated [28,33].

### 2.3. Hybrid mechanism

Routers with congestion management and QoS provision generally require a high number of buffers at switch ports, so the implementation cost is high and therefore prohibits their adoption of NoCs. An interconnection network architecture combining both technologies has been proposed recently [29,31]. Regional Explicit Congestion Notification (RECN) which needs no VCs and QoS-aware design with two VCs dramatically reduce resources requirement and design complexity. The combined architecture demonstrates cost efficiency and performance improvement.

This work integrated QoS provision and congestion-aware routing algorithm in an efficient way to facilitate packet transmission and enhance throughput for NoC routers.

## 3. High performance NoC platform

Exploring fully adaptive routing ability enables extensive routing flexibility and thus enhance network throughput. The double-y routing algorithm has been proposed as the solution inside a chip to support fully adaptive wormhole routing and maintain feasible design complexity [34]. There are two approaches of employing two additional vertical channels. One is using VC technique, and the other is using additional physical channel to form double-y networks. Although the former approach can reduces port numbers of routers, it might deteriorate transmission performance when workload increases. Adding additional physical ports relaxes the congestion problem and sustains fully adaptive routing benefit. NePA platform uses two extra ports for forming independent eastbound and westbound subnetworks [4]. Beyond that, DMesh platform is constructed by integrating four additional diagonal links to NePA to further take advantage of diagonal architecture and explore more routing space.

### 3.1. NePA system platform

NePA platform is a scalable, flexible and reconfigurable high performance NoC platform [2] which is based on a 2D mesh topology as shown in Fig. 1 and uses the wormhole packet switching technique. The router is connected with its four neighboring routers via six 64-bit bidirectional links, including two horizontal and four vertical links. A key feature of the NePA architecture is

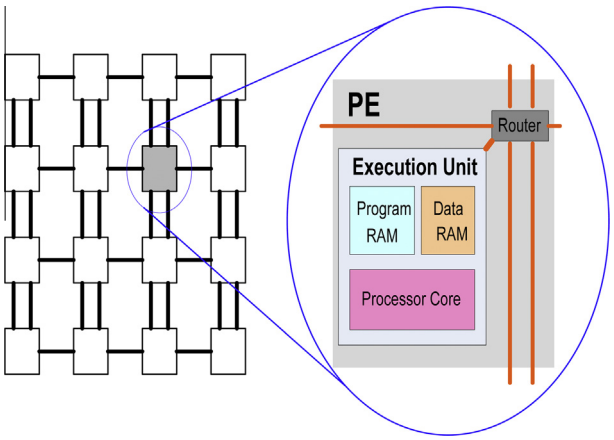



**Fig. 1.** A 4 ×4 NePA platform with two additional vertical links.

the use of two separate vertical links which provides insolated sub-networks and classifies packets into eastbound and westbound traffic. Within each subnetwork, adaptive minimal routing is performed to prevent cycles in the resource dependence graph and guarantees deadlock freedom [10]. Utilizing an adaptive XY routing scheme can increase network performance and provide fault-tolerant routing ability [4]. The router adaptively selects an alternative output port for packets when an output port is congested or the output buffer is full. Therefore, the link utilization is well balanced and network performance also improves.

### 3.2. DMesh system platform

DMesh network is constructed by integrating diagonal links to NePA, as presented in Fig. 2. DMesh network is composed of two sub-networks: E-subnet and W-subnet, represented with dashed arrows and solid arrows in Fig. 2, respectively. E-subnet is responsible for transferring eastbound packets while W-subnet is responsible for westbound traffic. When source PE starts packet transmission, it injects packets into one of the subnetworks depending on the direction of destination PE. Subsequently, packets traverse through one of the sub-networks to their destinations.

## 4. Router architecture

Adaptive routing algorithm approaches improve network performance by adjusting routing based on network situations. Multiple buffers inside each port mitigate Head-of-Line (HOL) blocking effects and enhance throughput and latency. Instead of VC approach, this work employs parallel buffers to solve the HOL blocking issue. To effectively utilize routing resource and improve throughput, Congestion-Aware (CA) routing scheme effectively relax packets from high congested areas and direct them to less congested ones. QoS-aware routing further improves performance for high priority application traffic.

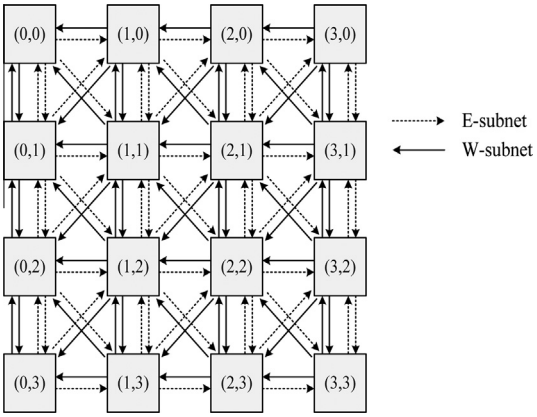



**Fig. 2.** A 4 ×4 DMesh platform with diagonal connection. Dash lines indicate eastbound subnetwork called E-subnet; solid lines indicate westbound subnetwork called W-subnet.

### 4.1. Adaptive routing algorithm

A double-y routing and allowed turns are shown in Fig. 3(a) and (b). Once packets are injected into the networks, they can only transmit in one of the disjoint eastbound and westbound subnetworks, following the minimal XY routing approach. If either *X* or *Y* direction occurs congestion indicated by link status, packets will go through less congested paths toward their destinations adaptively.

NePA/DMesh routers are mainly divided into internal router dealing with injecting and ejecting packets and two sub-routers: E-router and W-router. NePA follows minimal XY routing approach. However, DMesh adopts a quasi-minimal routing approach instead of a minimal one to well balance workloads over the network so that performance and throughput are dramatically improved especially in high workload situations [47]. Diagonal channels will be granted first, then horizontal or vertical channels will be taken if possible diagonal channels are taken or congested. The channels of an internal router have the lowest priority which can prevent further congestion particularly in heavy load situations.

### 4.2. Multiple parallel buffers enhanced architecture

Performance of wormhole routed networks suffers from the HOL blocking effect especially when in high workload scenarios. To tackle this problem, incorporating VC can effectively mitigate performance degradation and improve throughput and latency accordingly [26]. Different VC approaches have been adopted by many five ports routers and demonstrated their effects. However, VC flow control needs abundant control signals among routers to keep track of VC buffer status which causes tremendous overhead for multi-port routers. Traditional congestion monitor is conducted by measuring buffer occupation and passing message among routers. The number of control lines is independent of link bit-width, but one should avoid inefficient design, if control signals make up major portion of wirings. For example, in the case of a router with 4 four-slot VCs inside each input port, two bits are required to indicate available buffer length when each VC size is four flits, needs 8 (4 × 2) wires to indicate buffer occupancy, and 4 input ports totally need 32 bits to indicate current router buffer status. The overall buffer occupancy status of current router can be

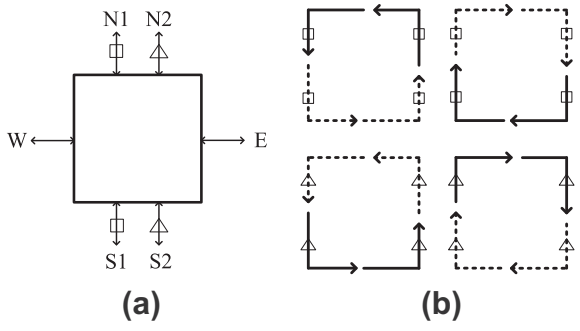



**Fig. 3.** Double-y network router architecture and the associated routing algorithms: (a) a router in double-y network (b) allowed turns by double-y routing. Dotted lines indicate prohibited turns.

viewed as congestion status and is broadcasted to all its neighbors. If all the buffer information of current router is broadcasted to all its neighbors, 128 ($32 \times 4$) bits are required. Therefore, detail congestion information from neighboring routers allows arbiter to choose the most congested input port from upstream routers and less congested output port from downstream routers by congestion condition information. To further mitigate wiring overhead, especially for multi-port designs such as NePA and Dmesh, coarse buffer status signaling with reduced wiring cost should be adopted. That is also one of the motivations for designing a light weight congestion detection and broadcasting mechanism.

Different from VC flow control, a new routing-independent Parallel Buffer (PB) structure and its management scheme were proposed to enhance network channel utilization but keep design overhead moderate [3]. An enhanced router example of NePA E-router is shown in Fig. 4. Each added channel keeps the merit of adaptive minimal routing strategy instead of mapping to dedicated outputs in a fixed pattern. This scheme works independently to explore more routing resources so that the channel utilization and maximum throughput are achieved accordingly. The proposed architecture maintains the routing flexibility to deliver packets toward paths with less congested possibility. Therefore packets can bypass blocked output ports and keep heading to destination with minimal routing paths.

### 4.3. Dynamic congestion-aware router architecture

A proposed lightweight dynamic arbitration mechanism has shown that its effectiveness in congestion detection and management [46,47]. The mechanism can especially reduce the wiring requirement because it only delivers congestion index calculated by the number of active FIFOs with waiting packets instead of detailed buffer or link information. The purpose of dynamic arbitration is to alleviate traffic congestion by allowing packets coming from hot spots to move first and use less congested routers to advance. Resources contention in the congested region is reduced accordingly. A congestion-aware routing procedure is described in Algorithm 1 to detail the approach. First, priority queue $PQ_i$ and $PQ_o$ associated with available input and output ports are established based on congestion indices from neighboring routers. Each input port is corresponding to a congestion index from its upstream router, and each output port is associated with a congestion index from its downstream router. Available input and output ports have associated keys identifying congestion status in upstream and downstream routers to decide the priority of each port. The arbiter matches input–output pair from $PQ_i$ which indicates highly congested traffic and from $PQ_o$ which indicates less congestion.

**Algorithm 1.** CA and QoS management mechanism

Input: $Packet_i$, *Cogestion_idx_in*
Output:*Cogestion_idx_out*
*PBn*: nonempty parallel buffers
*QoSn*: predefined service classes
$PQ_i$: a priority queue that holds M input ports
$PQ_o$: a priority queue that holds N output ports
$P_i$: input port index
$P_o$: output port index
1: Packets are inject into either E-subnet or W-subnet
2: Construct $PQ_i$ for input ports by *Cogestion_idx_in*
3: Construct $PQ_o$ for available output links by *Cogestion_idx_in*
4: Sort $PQ_i$ and $PQ_o$ according to congestion status
5: **for all** *QoSn* **do**
6: **for all** $P_o$ in $PQ_o$ **do**
7: **for all** $P_i$ in $PQ_i$ **do**
8: **for all** *PBn* in each $P_i$ **do**
9: **if** input–output pair is available **then**
10: route packet from selected input to output port
11: remove $P_i$ and $P_o$ from $PQ_i$ and $PQ_o$
12: **else**
13: restore $P_i$ and $P_o$ into $PQ_i$ and $PQ_o$
14: **end if**
15: **end for**
16: **end for**
17: **end for**
18: **end for**
19: Calculate congestion index from buffers with waiting packets
20: Transmit *Cogestion_idx_out* to neighbor routers

### 4.4. QoS-aware router architecture

Application workloads are classified into different service levels and indicated in the header field. Header parsing unit interprets associated QoS header field and gives routing preference to high

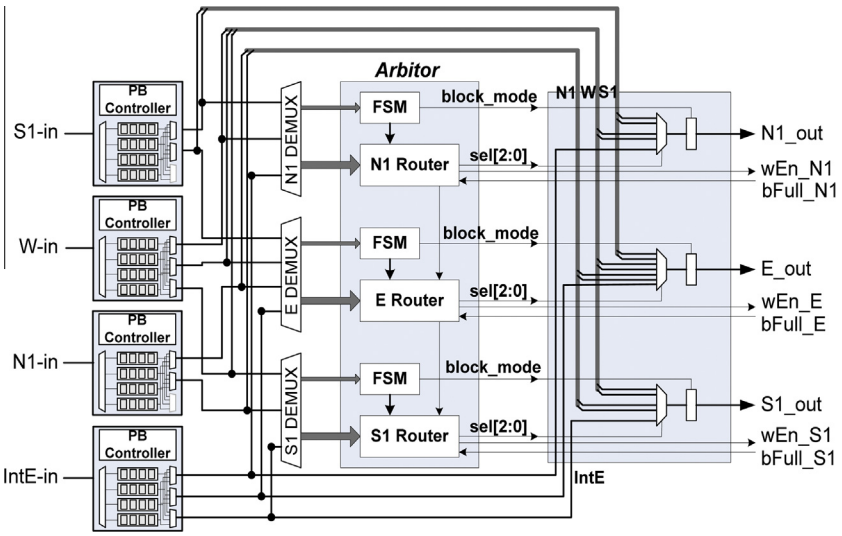



**Fig. 4.** An enhanced NePA E-router architecture with parallel buffers.

priority packets. Less congested output and high congested input are selected to advance high priority traffic first. For multiple PBs router mechanism, resource sharing and specific channel reservation for GT traffic are two strategies to differentiate resource allocation procedure. The former is described in Algorithm 1, all buffers are shared by all traffic including GT and BE. For resource reservation solution, one or multiple buffers are dedicated to GT traffic and others are assigned to BE traffic [24].

According to different resource deployment, QoS-aware router can be further classified into the following mechanisms.

- **Single Buffer Mechanism (SBM):** There is only one associated buffer with each input port. Arbiter routes GT traffic first based on available routing resources and congestion information. Although GT packets gain preference to be served, they might be blocked by BE ones due to shared buffer. Only the arbiter provides differentiated routing arbitration to enable preliminary QoS.
- **Multiple Buffers Mechanism (MBM):** All buffers are shared by all service classes. Although it has the potential performance degradation of GT traffic because of being blocked by BE traffic. This situation can be relaxed by multiple PBs design and dynamic adaptive routing. On one hand multiple PBs can store GT traffic to avoid blocking, and on the other the GT traffic can be advanced owing to preferably allocated resource. Buffer utilization is maximized whatever GT rate is set in this case.
- **Multiple Buffers Mechanism with Reserved Channel (MBMRC):** MBMRC reserves specific buffers to store GT traffic. Static buffer allocation for QoS is popular in assigning routing resource [9]. Different from that, GT buffers are reserved for GT traffic and other buffers are shared by all traffic to improve buffer utilization in our simulation. In two PBs case, $PBnum_0$ is dedicated to GT traffic, and $PBnum_1$ is shared by both GT and BE traffic. Guaranteed bandwidth is reserved for GT so as to provide better performance. It might cause resource utilization inefficiency when considering low GT traffic case.

Congestion and QoS aware mechanism differentiates traffic transmission and maximize network resource utilization efficiency. GT packets benefit from multiple PBs design and preferably adaptive routing. Congestion management helps to mitigate possible performance degradation in high workload cases because GT packets tend to be directed to less congested areas.

### 4.5. Starvation prevention

Resource allocation imbalance and high GT injection rate might cause starvation situation for BE traffic. However, setting 25% GT ratio limitation can avoid starvation at most cases [44]. In our high-throughput router architecture, experimental results have shown that the starvation can be avoided by limiting the ratio of GT traffic to be around 50% when considering multiple PBs without GT channel reservation for NePA. This is realistic as it is expected

**Table 1**
Simulation parameters.

| Characteristics | Baseline | Variations |
|---|---|---|
| Topology | 2D Mesh | – |
| Network size | 8x8 | 16x16, 32x32 |
| Routing | CA adaptive | Fixed priority |
| QoS mechanism | MBM | SBM, MBMRC |
| Router ports | 7(NePA) | 5(NoC5), 11(DMesh) |
| Parallel buffers/port | 2 | 1, 4 |
| Buffer size (flits)/PB | 4 | 2, 8, 16 |
| Packet length (flits) | 4 | 1, 8, 16 |
| Flit size (bits) | 64 | – |
| Traffic workload | Synthetic | Local, Hot spot |
| Simulated period (cycles) | 100,000 | – |

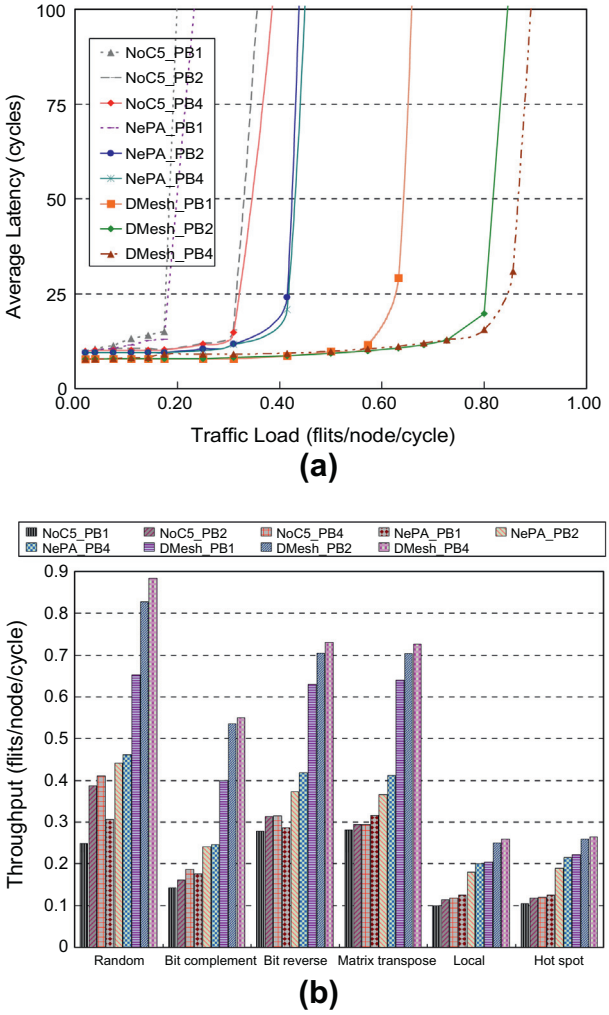



**Fig. 5.** Latency and throughput comparison among NoC5, NePA and DMesh platforms. (a) Latency comparison under uniform random traffic. (b) Throughput comparison for different PB numbers and router architectures under various traffic traces. NoC5/NePA/DMesh_PBn ($n$ = 1, 2, 4), $n$ means the number of parallel buffers.

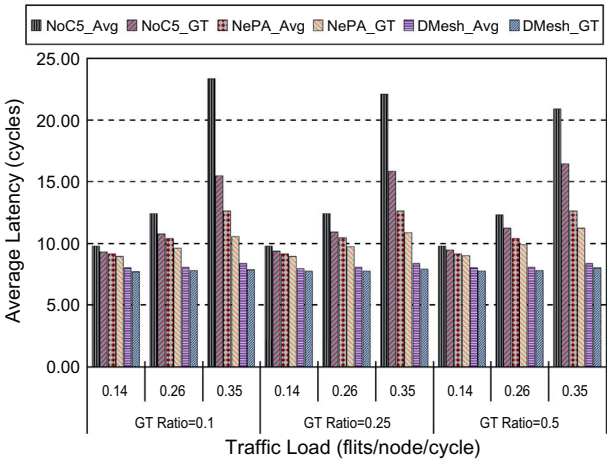



**Fig. 6.** Overall and GT traffic average latency comparison among NoC5, NePA and DMesh under Random traffic (2 PBs/port, 4 flits/buffer).

that GT class should only be assigned to a small portion of traffic. This can be easily achieved by self-disciplined processors.

Other than passive expectation of GT ratio limitation, QoS level boost mechanism is proposed to eliminate starvation. A timer for each buffer is needed to record how long packets have been blocked. When blocking time reaches the predefined threshold, BE packets are boosted to GT traffic in order to accelerate transmission. Starvation can be aggressively prevented in this manner.

## 5. Evaluation

The methodology used to analyze the performance and feasibility is discussed in this section. Different control factors such as network platform, PB allocation, routing algorithm, GT ratio, traffic pattern and workload will be evaluated and discussed.

### 5.1. Experimental setup

The proposed QoS and Congestion Aware (QoSCA) NoC platform is developed by a System-C based cycle accurate simulator. Table 1 lists detailed simulation configuration. Different mesh network sizes and router designs such as five-port (NoC5), seven-port (NePA), eleven-port (DMesh) routers are considered. Wormhole packet switching is adopted, and packets are composed of 64-bit flits. Traffic generator produces various synthetic traffic traces for evaluating the performance, including {*Random, Bit complement, Bit reverse, Matrix transpose*} traffic patterns [10]. Additionally, local and global hot region traffic conditions are also considered, labeled as {*Local, Hot spot*}. *Local* traffic features 80% of the total injected traffic with traverse distance less than four hops. *Hot spot* traffic features that 10% of the nodes receive 68% of the total injected traffic. These patterns define the spatial distribution of packets. A self-similar traffic generation technique was implemented to apply temporal distribution to transmitted packets [25,43]. Self-similar traffic can be generated by aggregating a large number of packet sources which exhibit a long-range dependence property [38]. ON/OFF state is imposed on source node to control traffic generation during simulation time. The length of time a node spends in the ON or OFF state is determined by the Pareto distribution [1]. Shape parameters, $\alpha_{ON}$ and $\alpha_{OFF}$, are used to calculate ON and OFF periods. $T_{ON} = U^{-1/\alpha_{ON}}$ and $T_{OFF} = U^{-1/\alpha_{OFF}}$, where U is a uniformly distributed value in the range of $(0,1]$, $\alpha_{ON} = 1.9$ and $\alpha_{OFF} = 1.25$ are set in the simulation.

An open-loop interconnection network measurement setup was used in simulations [10]. Packets are stored in an infinite queue at the source node after they are generated, and wait until they are injected into the network. This method isolates the packet generation from the network behavior which indicates the packet generation is independent of the network condition. Each simulation executes 10,000 clock cycles for warm-up and then continues for 100,000 cycles during which router performance is conducted.

### 5.2. Performance evaluation

Performance evaluation is based on transmission time and traffic load among source and destination pairs. Latency and throughput are major performance evaluation metrics. System performance evaluation was conducted among various NoC platforms, network sizes, router architectures, arbitration schemes, traffic configurations and resource assignment mechanisms to demonstrate the effectiveness of the proposed mechanism.

#### 5.2.1. Throughput enhanced router design

How port numbers (two additional vertical ports for NePA and diagonal links for DMesh) and PB configuration influence performance in terms of latency and throughput are shown in Fig. 5. The five-port counterpart adopts dimension order routing and all the cases implement the congestion-aware scheme in the simulation. Performance improves as routing resources increase. Multiple ports provide additional routing flexibility and bandwidth and multiple PBs alleviate congestion situation so as to accelerate data transmission. NePA outperforms NoC5 routers and DMesh outperforms NePA because of express diagonal link employment. NoCs with more ports meet the expectation of performance improvement at the same buffer level. It is also noticed that adding more PBs cannot account for definite performance improvement. NePA and DMesh with less PBs work better than NoC5 with four or eight PBs. DMesh_PB2 improves accommodated throughput more than NePA_PB2 and NePA_PB4, and NePA_PB2 improves it more than NoC5_PB2 and NoC5_PB4. Even DMesh_PB1 accommodates more traffic than NePA and NoC5 with multiple PBs. The result reflects that adding extra routing resource benefits overall system performance owing to routing flexibility. The observation from different traffic patterns comes to the same conclusion, and the improvement is significant especially in symmetric traffic scenarios such as {*Bit reverse*} and {*Matrix transpose*}.

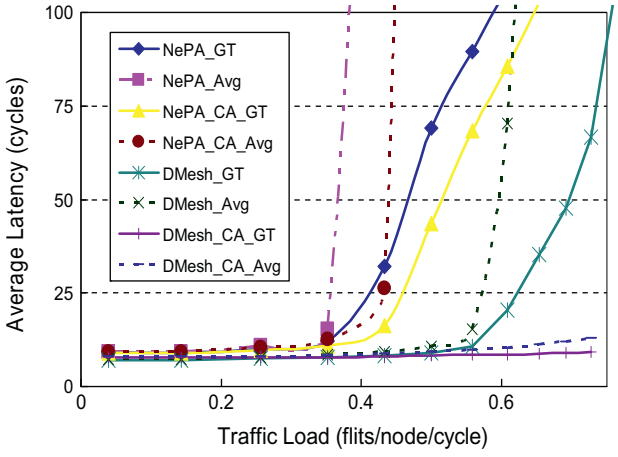



**Fig. 7.** Overall and GT traffic average latency comparison between original and congestion-aware NePA/DMesh architectures (GT ratio = 0.25, MBM is used, 2 PBs/port, 4 flits/buffer).

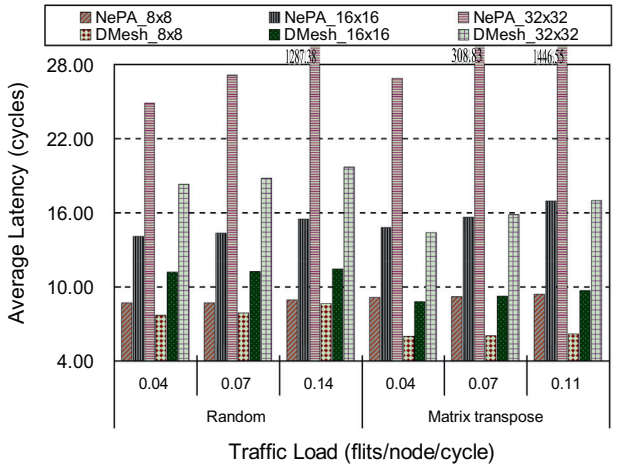



**Fig. 8.** GT traffic average latency comparison between NePA and DMesh under Random and Matrix transpose cases (2 PBs/port, 4 flits/buffer).

*5.2.2. NoC platforms for QoS*

Additional channel bandwidth in NePA increases traffic accommodation and reduces latency. DMesh further takes advantage of the effect and improves performance by inserting diagonal links to accelerate long distance communication as well as lower transmission burden from local routers. The improvement becomes significant as the workload increases, as shown in Fig. 6. Besides GT traffic is guaranteed to gain the best performance, the overall average latency is dramatically improved for DMesh, even better than GT traffic in NePA and NoC5. The conclusion can be easily observed in congested situations for different GT ratio cases.

*5.2.3. Congestion management effectiveness for QoS*

GT traffic can take advantage of available resources with less congestion and achieve significant performance improvement. Other than that, paths taken by GT traffic have less possibility of being blocked which can guarantee packet advancement. Routers with a fixed and CA routing arbitration are compared to demonstrate the effectiveness of CA mechanism, as shown in Fig. 7. CA mechanism effectively enhances GT and overall transmission performance. Among them, DMesh_CA provides the best performance and tolerated throughput.

The CA scheme significantly improves the average latency of GT traffic as well as that of overall traffic, especially in high workload situations. By employing a congestion control scheme, transmitted flits can eventually find available paths to dedicated destination in an acceptable time. GT traffic gains the preference to routing resources and achieves better performance than the other. Overall latency also benefits from well-balanced traffic.

*5.2.4. Network size for QoS*

Larger NoC platforms help to clearly explore and observe long distance effect and importance of congestion avoidance mechanism. Performance comparison between NePA and DMesh under {*Random*} and {*Matrix transpose*} traffic is shown in Fig. 8. The improvement becomes more significant as network size increases because potential congestion situations might deteriorate transmission performance in platforms with more routers. For {*Matrix transpose*} traffic where source–destination pairs are located in diagonal positions, the remarkable latency improvement emphasizes the major virtue of diagonal express links. It is noted that DMesh maintains consistent transmission quality over different network sizes.

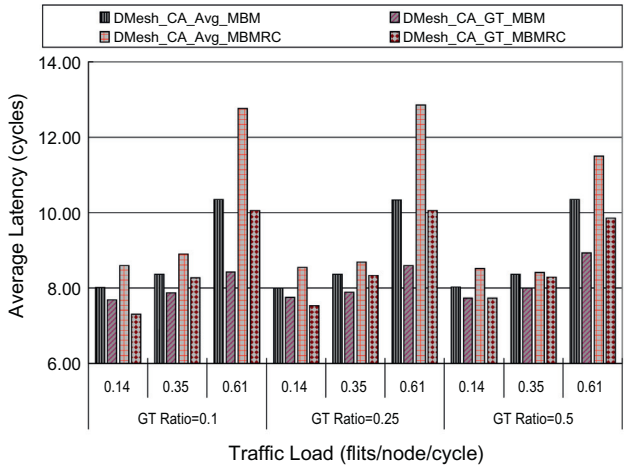
*5.2.5. Resource allocation effectiveness for QoS*

Multiple PBs help to alleviate congestion situations and further improve performance. Traffic associated with different service classes can be isolated and stored in different buffers to mitigate order error effect [31]. GT traffic therefore can be routed first and prevented from being blocked by BE traffic. The impact of different PB configurations for random traffic trace was investigated in Table 2. Preliminary QoS provision still can be achieved even for NoCs with single buffer cases. For NoC5 and NePA, GT packets achieve relatively better performance than overall packets. GT traffic might be blocked by BE traffic and hinder its advancement. Multiple PBs can effectively separate GT traffic from BE one and further provide privileged bandwidth to GT traffic. NePA/DMesh with multiple PBs demonstrate significant performance improvement over other cases. DMesh benefits from express links, so GT traffic even BE traffic if the resource is available can accelerate transmission and shorten latency dramatically.

It is observed that in the case of low GT ratio and high workload, the reserved channel might be under-utilized and the performance of overall and GE traffic suffers from poor resource allocation. A detailed investigation between mechanisms with reserved channels and without reserved channels has been performed. Average and GT latency analysis for DMesh has been conducted and shown in Fig. 9(a). It is noted that GT traffic latency of MBMRC outperforms

**Table 2**
Overall, GT and BE traffic average latency comparison among NoC5/NePA/DMesh architectures under various PB setups, GT ratios and workloads (MBM is used for PB = 2 and PB = 4).

| Load (flits/node/cycle) | | Random | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | GT ratio = 0.1 | | | GT ratio = 0.25 | | | GT ratio = 0.5 | | |
| | | 0.14 | 0.26 | 0.35 | 0.14 | 0.26 | 0.35 | 0.14 | 0.26 | 0.35 |
| PB=1 | NoC5_Avg | 11.79 | 427.73 | 2430.15 | 11.83 | 470.86 | 2511.90 | 11.88 | 498.33 | 2519.39 |
| | NoC5_GT | 10.63 | 41.36 | 57.50 | 10.82 | 47.86 | 69.55 | 11.12 | 66.75 | 107.60 |
| | NoC5_BE | 11.92 | 470.61 | 2691.21 | 12.17 | 611.33 | 3307.44 | 12.66 | 925.94 | 4828.62 |
| | NePA_Avg | 9.88 | 17.89 | 1141.51 | 9.91 | 19.54 | 1219.16 | 9.90 | 20.49 | 1307.69 |
| | NePA_GT | 9.39 | 13.65 | 49.40 | 9.49 | 14.87 | 57.39 | 9.62 | 16.51 | 88.70 |
| | NePA_BE | 9.94 | 18.36 | 1261.33 | 10.05 | 21.11 | 1596.40 | 10.18 | 24.48 | 2448.75 |
| | DMesh_Avg | 8.02 | 8.07 | 8.36 | 7.99 | 8.06 | 8.37 | 8.03 | 8.05 | 8.36 |
| | DMesh_GT | 7.69 | 7.79 | 7.88 | 7.76 | 7.76 | 7.89 | 7.74 | 7.84 | 8.01 |
| | DMesh_BE | 8.05 | 8.10 | 8.42 | 8.07 | 8.16 | 8.53 | 8.34 | 8.26 | 8.73 |
| PB=2 | NoC5_Avg | 9.78 | 12.42 | 23.35 | 9.78 | 12.40 | 22.12 | 9.78 | 12.34 | 20.91 |
| | NoC5_GT | 9.32 | 10.75 | 15.49 | 9.34 | 10.94 | 15.84 | 9.46 | 11.21 | 16.48 |
| | NoC5_BE | 9.83 | 12.61 | 24.25 | 9.93 | 12.89 | 24.25 | 10.10 | 13.47 | 25.40 |
| | NePA_Avg | 9.18 | 10.42 | 12.65 | 9.18 | 10.44 | 12.65 | 9.19 | 10.41 | 12.61 |
| | NePA_GT | 8.94 | 9.62 | 10.58 | 8.95 | 9.72 | 10.86 | 9.01 | 9.89 | 11.24 |
| | NePA_BE | 9.21 | 10.51 | 12.89 | 9.26 | 10.68 | 13.26 | 9.37 | 10.94 | 14.01 |
| | DMesh_Avg | 7.28 | 7.92 | 8.21 | 7.28 | 7.94 | 8.29 | 7.28 | 8.02 | 8.34 |
| | DMesh_GT | 7.27 | 7.62 | 7.81 | 7.26 | 7.68 | 7.83 | 7.28 | 7.72 | 7.98 |
| | DMesh_BE | 7.29 | 7.96 | 8.36 | 7.29 | 7.99 | 8.48 | 7.29 | 8.12 | 8.71 |
| PB=4 | NoC5_Avg | 9.77 | 12.30 | 20.22 | 9.78 | 12.28 | 19.28 | 9.77 | 12.20 | 17.81 |
| | NoC5_GT | 9.31 | 10.58 | 14.11 | 9.33 | 10.75 | 13.99 | 9.45 | 11.01 | 14.11 |
| | NoC5_BE | 9.83 | 12.49 | 20.92 | 9.93 | 12.79 | 21.08 | 10.10 | 13.39 | 21.55 |
| | NePA_Avg | 9.18 | 10.41 | 12.44 | 9.18 | 10.41 | 12.30 | 9.19 | 10.39 | 12.32 |
| | NePA_GT | 8.93 | 9.58 | 10.41 | 8.95 | 9.68 | 10.51 | 9.01 | 9.85 | 10.88 |
| | NePA_BE | 9.21 | 10.50 | 12.68 | 9.26 | 10.66 | 12.91 | 9.36 | 10.93 | 13.77 |
| | DMesh_Avg | 7.14 | 7.66 | 8.22 | 7.14 | 7.66 | 8.22 | 7.14 | 7.66 | 8.23 |
| | DMesh_GT | 7.00 | 7.41 | 7.74 | 7.04 | 7.45 | 7.80 | 7.06 | 7.50 | 7.92 |
| | DMesh_BE | 7.15 | 7.69 | 8.28 | 7.17 | 7.74 | 8.36 | 7.22 | 7.82 | 8.54 |



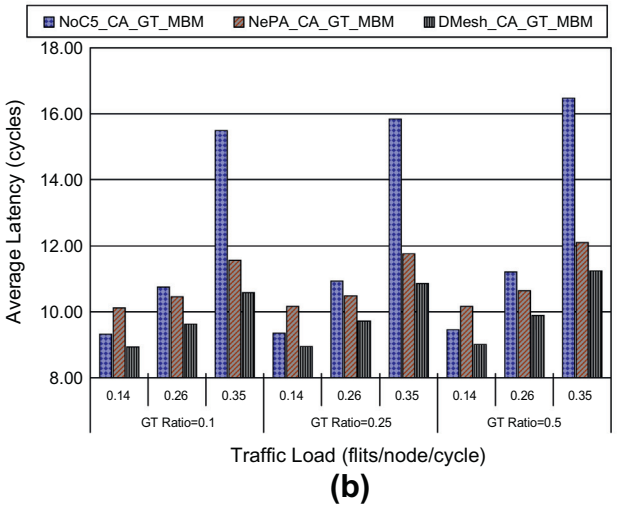




**Fig. 9.** Latency comparison for DMesh architecture with and without reserved channels under Random workload (2 PBs/port, 4 flits/buffer). (a) Overall and GT traffic average latency comparison between adaptive buffer assignment and reserved channel. (b) GT traffic average latency comparison using adaptive buffer assignment among NoC5, NePA and DMesh platforms.

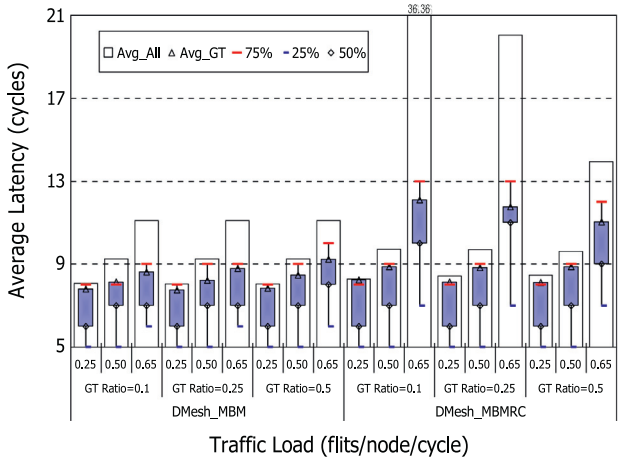



**Fig. 10.** GT traffic average latency statistical analysis comparison between adaptive and reserved buffer assignment for various traffic loads and GT ratio under Random traffic, 25/50/75% are the statistical results showing 25/50/75% of received GT packets are under the listed latency value. (2 PBs/port, 4 flits/buffer).

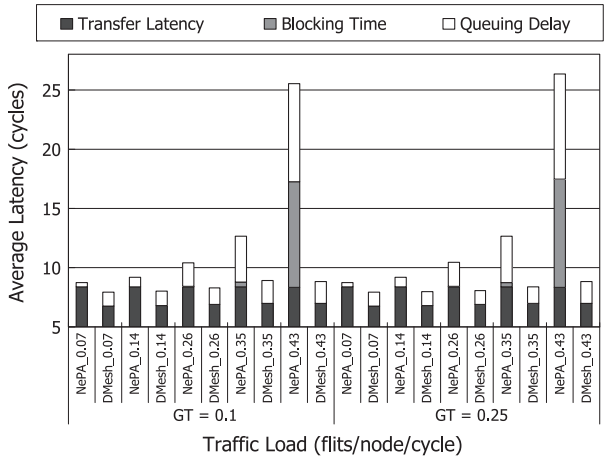



**Fig. 11.** GT traffic average latency delay breakdown comparison between NePA and DMesh for various traffic loads and GT ratio (2 PBs/port, 4 flits/buffer).

that of MBM under low workload condition (0.14 flits/node/cycle) for different GT ratios. As workload increases, reserved GT channels are not sufficient to accommodate GT traffic so that GT traffic will take the shared buffers and therefore hinder BE traffic. This situation in MBMRC case is worse than MBM. Because MBM can efficiently and flexibly allocate buffers to both GT and BE in order to prevent them from under-utilization, GT traffic can achieve the best performance while maintaining tolerable average latency. NoC5 and NePA also demonstrate the same tendency from our observation. Fig. 9(b) shows GT latency comparison among NoC5, NePA and DMesh with MBM under various GT ratios and workload conditions.

A statistical timing analysis between MBM and MBMRC has been conducted in Fig. 10. Statistical results demonstrate that average latency of MBM 75% GT is comparable to that of MBMRC under light and medium workload cases. The overall and GT traffic latency improvement of MBM over MBMRC becomes significant when heavy workload is presented. MBM maintains consistent performance under various workloads because MBM not only favors GT data by differentiated traffic management, but also tremendously reduces system average latency by sophisticated resource allocation strategy and contention elimination.

### 5.2.6. Latency breakdown analysis

Latency breakdown analysis takes a closer look at understanding the bottleneck of network transmission and attempt to reveal the effect of transmission delay mitigation for the proposed router architecture. Fig. 11 shows the timing analysis between NePA and DMesh platforms under various GT ratios and workloads. Transfer Latency ($T_L$) calculates the latency between source and destination nodes; Blocking Time ($T_B$) shows blocking time when flits are stuck in buffers to wait for transmission; Queuing Delay ($T_Q$) represents waiting time between generation time and injection time [47].

$T_L$ and $T_B$ are obviously shortened for DMesh because diagonal links employment reduces physical transmission distance and mitigate congestion possibility. Under 0.03 and 0.07 flits/node/cycle workload cases, $T_Q$ in DMesh is traded for deciding better routing paths so that overall latency is still better than NePA. Unlike NePA, $T_Q$ in DMesh increases moderately when in high workloads. High-throughput DMesh routers result in better load balance and data distribution, so total latency is significantly reduced, especially in congested situations.

### 5.2.7. Different traffic cases evaluation for QoS

A comprehensive performance comparison for various traffic patterns was shown in Table 3. The results indicate consistent

conclusion that the CA mechanism collaborating with QoS provision with two PBs can effectively provide guaranteed bandwidth with GT traffic to ensure its performance, even in local and hot spot cases.

The following conclusion can be made based on simulation results.

- Parallel buffer architecture alleviates congestion and allows reserving channel for GT traffic.
- Adaptive routing approach and congestion-aware mechanism improve overall throughput by well balancing transfer tasks. CA mechanism can be designed to give preference to GT traffic so as to ensure guaranteed bandwidth and performance.
- The extra routing resources from NePA and DMesh effectively enhance QoS provision especially in high workload cases.
- QoS provision was validated by different network platforms, routing algorithms, buffer deployment, GT ratios, workloads and traffic patterns.

### 5.3. Implementation cost evaluation

Feasibility is evaluated by hardware cost estimation in terms of area and power consumption. CA routers for NePA and DMesh platforms have been designed at Register-Transfer Level (RTL) in *Verilog*™ HDL. A logic description of NePA and DMesh router component has been obtained using *Synopsys*™ Design Compiler and *TSMC*™ 65 nm CMOS generic process technology to perform logic synthesis and analyze hardware cost. Synthesis condition is set to 800 *MHz* and switching activity is set to 10%.

The DMesh microarchitecture is illustrated in Fig. 12. Two separate E_router and W_router process output ports of eastward and westward traffic. Int_router is deployed to process packets ejecting to local PEs. There are parallel FIFOs and one PB controller associated with each input port. Header Parsing Unit (HPU) interprets destination information from the header flit and Arbiter Logic (AL) decides routing path, performs arbitration and manages the

**Table 3**
Overall, GT and BE traffic average latency comparison among NoC5/NePA/DMesh architectures under various traffic patterns, GT ratios and workloads (MBM is used in the simulation).

| Load (flits/node/cycle) | | PB = 2 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | GT ratio = 0.1 | | | GT ratio = 0.25 | | | GT ratio = 0.5 | | |
| | | 0.07 | 0.14 | 0.26 | 0.07 | 0.14 | 0.26 | 0.07 | 0.14 | 0.26 |
| Bit complement | NoC5_Avg | 12.13 | 14.29 | 1996.25 | 12.13 | 14.28 | 2449.41 | 12.13 | 14.27 | 3604.54 |
| | NoC5_GT | 11.78 | 12.87 | 1356.34 | 11.87 | 13.01 | 1643.40 | 11.98 | 13.33 | 2987.65 |
| | NoC5_BE | 12.17 | 14.45 | 2065.84 | 12.21 | 14.69 | 2717.60 | 12.28 | 15.20 | 4253.34 |
| | NePA_Avg | 11.65 | 13.09 | 596.65 | 11.65 | 13.11 | 721.07 | 11.66 | 13.10 | 848.84 |
| | NePA_GT | 11.54 | 12.20 | 32.52 | 11.57 | 12.25 | 39.78 | 11.62 | 12.47 | 78.64 |
| | NePA_BE | 11.66 | 13.19 | 657.90 | 11.68 | 13.39 | 947.51 | 11.70 | 13.73 | 1601.71 |
| | DMesh_Avg | 9.40 | 9.65 | 10.68 | 9.40 | 9.66 | 10.64 | 9.44 | 9.67 | 10.70 |
| | DMesh_GT | 9.15 | 9.43 | 9.77 | 9.32 | 9.44 | 9.86 | 9.38 | 9.53 | 10.02 |
| | DMesh_BE | 9.42 | 9.67 | 10.79 | 9.42 | 9.73 | 10.90 | 9.50 | 9.81 | 11.40 |
| Bit reverse | NoC5_Avg | 9.83 | 24.16 | 1655.26 | 9.83 | 24.19 | 1821.98 | 9.83 | 24.19 | 2071.28 |
| | NoC5_GT | 9.50 | 14.33 | 1064.92 | 9.51 | 14.29 | 1266.85 | 9.64 | 15.68 | 1662.40 |
| | NoC5_BE | 9.86 | 25.23 | 1719.31 | 9.94 | 27.50 | 2006.43 | 10.02 | 32.61 | 2473.86 |
| | NePA_Avg | 9.07 | 9.54 | 10.55 | 9.08 | 9.55 | 10.56 | 9.08 | 9.55 | 10.56 |
| | NePA_GT | 9.08 | 9.26 | 9.84 | 9.07 | 9.35 | 9.94 | 9.10 | 9.40 | 10.05 |
| | NePA_BE | 9.07 | 9.57 | 10.63 | 9.08 | 9.62 | 10.77 | 9.06 | 9.70 | 11.06 |
| | DMesh_Avg | 7.09 | 7.22 | 7.62 | 7.11 | 7.24 | 7.86 | 7.14 | 7.34 | 7.91 |
| | DMesh_GT | 6.88 | 7.02 | 7.22 | 6.92 | 7.12 | 7.55 | 6.99 | 7.22 | 7.76 |
| | DMesh_BE | 7.11 | 7.25 | 7.72 | 7.17 | 7.29 | 7.97 | 7.38 | 7.44 | 8.19 |
| Matrix transpose | NoC5_Avg | 9.89 | 29.68 | 1742.15 | 9.89 | 29.68 | 1741.91 | 9.89 | 29.68 | 1741.62 |
| | NoC5_GT | 9.45 | 13.96 | 1358.40 | 9.51 | 14.63 | 1436.66 | 9.67 | 15.75 | 1502.86 |
| | NoC5_BE | 9.94 | 31.39 | 1783.93 | 10.02 | 34.65 | 1843.76 | 10.11 | 43.45 | 1979.52 |
| | NePA_Avg | 9.30 | 9.61 | 10.65 | 9.30 | 9.61 | 10.65 | 9.30 | 9.61 | 10.62 |
| | NePA_GT | 9.18 | 9.44 | 10.06 | 9.21 | 9.39 | 10.09 | 9.29 | 9.44 | 10.19 |
| | NePA_BE | 9.32 | 9.63 | 10.72 | 9.33 | 9.68 | 10.84 | 9.32 | 9.77 | 11.05 |
| | DMesh_Avg | 6.50 | 6.55 | 6.77 | 6.54 | 6.55 | 6.84 | 6.55 | 6.65 | 6.90 |
| | DMesh_GT | 6.03 | 6.15 | 6.60 | 6.04 | 6.47 | 6.67 | 6.39 | 6.47 | 6.67 |
| | DMesh_BE | 6.53 | 6.61 | 6.79 | 6.57 | 6.72 | 6.93 | 6.71 | 6.81 | 7.03 |
| | | 0.05 | 0.10 | 0.15 | 0.05 | 0.10 | 0.15 | 0.05 | 0.10 | 0.15 |
| Local | NoC5_Avg | 9.71 | 19.10 | 319.84 | 9.54 | 18.64 | 292.01 | 9.35 | 17.34 | 291.40 |
| | NoC5_GT | 8.00 | 9.17 | 10.76 | 7.91 | 9.21 | 10.90 | 8.01 | 9.46 | 11.61 |
| | NoC5_BE | 9.91 | 20.17 | 353.65 | 10.09 | 21.77 | 384.58 | 10.73 | 25.25 | 570.97 |
| | NePA_Avg | 7.65 | 7.94 | 15.47 | 7.62 | 7.88 | 12.49 | 7.57 | 7.90 | 12.25 |
| | NePA_GT | 7.10 | 7.48 | 8.82 | 7.16 | 7.49 | 8.68 | 7.17 | 7.48 | 8.86 |
| | NePA_BE | 7.72 | 7.99 | 16.21 | 7.77 | 8.00 | 13.74 | 7.98 | 8.30 | 15.65 |
| | DMesh_Avg | 6.08 | 6.18 | 6.34 | 6.09 | 6.18 | 6.36 | 6.06 | 6.16 | 6.26 |
| | DMesh_GT | 6.07 | 6.11 | 6.19 | 6.08 | 6.16 | 6.22 | 6.05 | 6.07 | 6.20 |
| | DMesh_BE | 6.08 | 6.20 | 6.35 | 6.09 | 6.19 | 6.40 | 6.09 | 6.16 | 6.32 |
| Hot spot | NoC5_Avg | 11.27 | 58.60 | 607.86 | 11.59 | 46.62 | 598.87 | 11.84 | 41.62 | 589.30 |
| | NoC5_GT | 9.17 | 10.25 | 12.84 | 9.01 | 10.63 | 13.01 | 9.21 | 11.18 | 14.49 |
| | NoC5_BE | 11.49 | 64.00 | 672.24 | 12.46 | 58.79 | 791.72 | 14.57 | 72.33 | 1141.88 |
| | NePA_Avg | 8.12 | 9.97 | 18.79 | 8.13 | 10.18 | 15.43 | 8.23 | 10.17 | 14.28 |
| | NePA_GT | 7.94 | 8.37 | 9.70 | 7.92 | 8.48 | 9.77 | 7.90 | 8.52 | 9.82 |
| | NePA_BE | 8.14 | 10.15 | 19.80 | 8.19 | 10.73 | 17.27 | 8.55 | 11.81 | 18.68 |
| | DMesh_Avg | 6.31 | 6.52 | 6.87 | 6.31 | 6.43 | 6.84 | 6.31 | 6.54 | 6.93 |
| | DMesh_GT | 6.07 | 6.31 | 6.63 | 6.16 | 6.39 | 6.67 | 6.20 | 6.44 | 6.74 |
| | DMesh_BE | 6.34 | 6.54 | 6.90 | 6.36 | 6.45 | 6.90 | 6.42 | 6.64 | 7.12 |

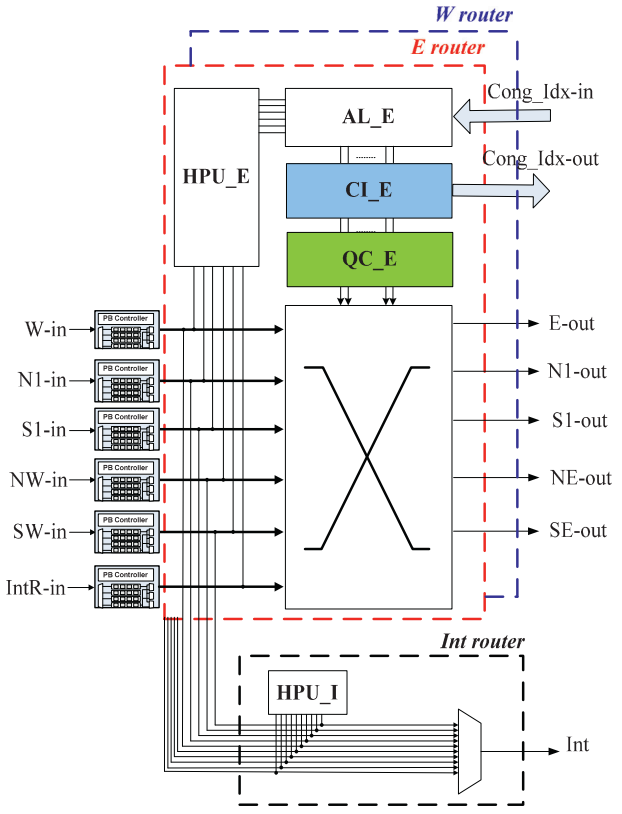



**Fig. 12.** DMesh Router MicroArchitecture.

crossbar switch. Switching unit takes care of practical packet traversal from input FIFOs to output links. Congestion Index (CI) logic calculates the number of flits that are blocked and passes the congestion index to neighboring routers. Quality Control (QC) logic performs traffic classification and priority decision to optimize network resource utilization.

The implementation cost is evaluated in terms of area and power consumption from circuit synthesis reports. Increasing buffer size or virtual channel numbers cannot always account for better transmission performance. From our observation, adding additional routing resources is the plausible approach to this problem. In order to provide the routing flexibility, more buffers and powerful switches are needed to manage various routing directions, and that is the major reason why DMesh outperforms NePA and traditional five-port routers in terms of average latency and throughput. Besides, the proposed router design eliminates the need for having virtual channel control and allocation overhead to compensate powerful switch design. Performance comparisons of NePA (7 ports) with FIFO size = 8/16 and DMesh with FIFO size = 4/8 were conducted. The results demonstrated that increasing alternative ports (DMesh) outperforms increasing FIFO size (NePA). From this point of view, DMesh is an area-efficient design at the same cost level [47].

Feasibility analysis was emphasized by the overhead as compared with baseline NePA/DMesh platforms. Table 4 also shows the area and power comparison with a typical $5 \times 5$ mesh. The CA router employs two adders to calculate congestion indices of eastward and westward sub-routers separately and modify routing arbiter from fixed priority to dynamic priority arbitration which is composed of a priority multiplexer circuit. NoC5_Orion2,[1] one five-port NoC counterpart, is listed for comparison. The implementation cost for CA routers with single buffer (FIFO = 4) was calculated to be 31,524 $\mu m^2$ for NePA_CA and 59,939 $\mu m^2$ for DMesh_CA, which reflects 4.1% and 6% increase over baseline designs. The increase becomes negligible when FIFO = 8, proving that CA routers enhance interconnection network throughput with a cost efficient modification [45].

[1] Area and power consumption is estimated from Orion2 simulator [22]. The configuration parameters are: 65 nm technology, 800 MHz speed, 4 VC each port, 8 flits each buffer and flit size is 64 bits.

**Table 4**
Cost comparison of baseline and CA router design with single FIFO per input port.

| | FIFO = 8 (4VC) | FIFO = 4 | | FIFO = 8 | | FIFO = 4 | |
|---|---|---|---|---|---|---|---|
| | NoC5_Orion2 | NePA | NePA_CA | NePA | NePA_CA | DMesh | DMesh_CA |
| Area ($\mu m^2$) | 170,442 | 30,295 | 31,524 | 48,900 | 49,407 | 56,583 | 59,939 |
| Dynamic power (mW) | 29.33* | 20.27 | 20.40 | 37.33 | 37.34 | 33.68 | 34.16 |
| Leakage power ($\mu W$) | | 139.08 | 147.44 | 227 | 277.75 | 265.02 | 278.88 |

* Total power consumption including dynamic and leakage power.

**Table 5**
Cost comparison between original and QoSCA designs.

| | 2 PBs/port | | 4 flits/buffer | |
|---|---|---|---|---|
| | NePA | NePA_QoSCA | DMesh | DMesh_QoSCA |
| Area ($\mu m^2$) | 47,038 | 52,222 | 84,016 | 92,986 |
| Dynamic power (mW) | 4.66 | 4.73 | 7.69 | 8.52 |
| Leakage power ($\mu W$) | 274.73 | 279.32 | 457.83 | 503.22 |

To evaluate the feasibility for QoS provision with multiple PBs architecture, routers with two PBs each port and four flits each PB are used to estimate the hardware cost. Besides extra buffers, the routing arbiter has to be modified to provide priority arbitration which is composed of a priority multiplexer circuit. Table 5 illustrates implementation cost overhead for both NePA and DMesh platforms. It shows that NePA_QoSCA increases area by 11%, DMesh_QoSCA increases area by only 10.7% and negligible power consumption increase, proving that the proposed mechanism can be achieved with a cost efficient modification from original designs. To maintain moderate cost, DMesh_QoSCA with smaller PB (two flits each PB) or fewer PBs (2 PBs instead of 4 PBs) serves as the best candidate because effectiveness of routing flexibility outweighs buffer size [45]. Simulation results in Section 5.2.1 also reach the same conclusion.

## 6. Conclusion

Current high-performance routers demand congestion management and QoS provision for boosting network performance and supporting differentiated services. Flexible router design and adaptive routing algorithm not only effectively exploit routing resources, but also support more advanced features to accommodate versatile application traffic traces. Fully adaptive routing, parallel buffers and congestion-aware routers alleviate congestion and enhance NoC performance in terms of latency and throughput. QoS-aware routing without multiple PBs provides acceptable performance improvement for GT traffic. Routers with multiple PBs further provide guaranteed transmission performance. Experimental results showed that performance improvement is considerable and implementation cost overhead is moderate for both NePA and DMesh platforms. With alternative links employed among routers, DMesh demonstrated significant performance enhancement for GT and overall traffic.

## References


[1] D.R. Avresky, V. Shurbanov, R. Horst, P. Mehra, Performance evaluation of the ServerNet(R) SAN under self-similar traffic. In: Proc. IPPS/SPDP Parallel and Distributed Processing 13th Int. and 10th Symp. Parallel and Distributed Processing, 1999, pp. 143–147.

[2] J.H. Bahn, N. Bagherzadeh, Design of simulation and analytical models for a 2D-meshed asymmetric adaptive router, IET Computers & Digital Techniques 2 (1) (2008) 63–73.

[3] J.H. Bahn, N. Bagherzadeh. Efficient parallel buffer structure and its management scheme for a robust Network-on-Chip (NoC) architecture, in: 13th International CSI Computer Conference CSICC, Kish Island, Iran, March 2008, pp. 98–105.

[4] J.H. Bahn, S.E. Lee, N. Bagherzadeh, On design and analysis of a feasible Network-on-Chip (NoC) architecture, in: Proc. Fourth Int. Conf. Information Technology ITNG '07, 2007, pp. 1033–1038.

[5] J.H. Bahn, S.E. Lee, Y.S. Yang, J. Yang, N. Bagherzadeh, On design and application mapping of a Network-on-Chip (NoC) architecture, Parallel Processing Letters 18 (2008) 239–255.

[6] S. Bell, B. Edwards, J. Amann, R. Conlin, K. Joyce, V. Leung, J. MacKay, M. Reif, L. Bao, J. Brown, M. Mattina, C.-C. Miao, C. Ramey, D. Wentzlaff, W. Anderson, E. Berger, N. Fairbanks, D. Khan, F. Montenegro, J. Stickney, J. Zook. TILE64 – Processor: a 64-core SoC with mesh interconnect, in: Proc. Digest of Technical Papers. IEEE Int. Solid-State Circuits Conf. ISSCC 2008, 2008, pp. 88–598.

[7] L. Benini, G. De Micheli, Networks on chip: a new paradigm for systems on chip design, in: Proc. Design, Automation and Test in Europe Conf. and Exhibition, 2002, pp. 418–419.

[8] T. Bjerregaard, J. Sparso, A router architecture for connection-oriented service guarantees in the MANGO clockless network-on-chip, in: Proc. Design, Automation and Test in Europe, 2005, pp. 1226–1231.

[9] E. Bolotin, I. Cidon, R. Ginosar, A. Kolodny, QNoC: QoS architecture and design process for network on chip, Journal of Systems Architecture 50 (2004) 105–128.

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

[11] W.J. Dally, B. Towles, Route packets, not wires: on-chip interconnection networks, in: Proc. Design Automation Conf, 2001, pp. 684–689.

[12] M. Daneshtalab, M. Ebrahimi, P. Liljeberg, J. Plosila, H. Tenhunen, Input–output selection based router for Networks-on-Chip, in: ISVLSI, 2010, pp. 92–97.

[13] M. Daneshtalab, M. Ebrahimi, P. Liljeberg, J. Plosila, H. Tenhunen, Memory-efficient on-chip network with adaptive interfaces, IEEE Transactions on CAD of Integrated Circuits and Systems 31 (1) (2012) 146–159.

[14] M. Daneshtalab, M. Ebrahimi, J. Plosila, H. Tenhunen, CARS: congestion-aware request scheduler for network interfaces in NoC-based manycore systems, in: Proceedings of the Conference on Design, Automation and Test in Europe, DATE '13, San Jose, CA, USA, 2013, pp. 1048–1051 (EDA Consortium).

[15] M. Daneshtalab, M. Kamali, M. Ebrahimi, S. Mohammadi, A. Afzali-Kusha, J. Plosila, Adaptive input–output selection based on-chip router architecture, Journal of Low Power Electronics 8 (1) (2012) 11–29.

[16] M. Ebrahimi, M. Daneshtalab, F. Farahnakian, J. Plosila, P. Liljeberg, M. Palesi, H. Tenhunen, HARAQ: congestion-aware learning model for highly adaptive routing algorithm in on-chip networks, in: 2012 Sixth IEEE/ACM International Symposium on Networks on Chip (NoCS), IEEE, 2012, pp. 19–26.

[17] M. Ebrahimi, M. Daneshtalab, P. Liljeberg, J. Plosila, H. Tenhunen, CATRA: congestion aware trapezoid-based routing algorithm for on-chip networks, in: Proceedings of the Conference on Design, Automation and Test in Europe, DATE '12, San Jose, CA, USA, 2012, pp. 320–325 (EDA Consortium).

[18] K. Goossens, J. Dielissen, A. Radulescu, AEthereal network on chip: concepts, architectures, and implementations, IEEE Design & Test of Computers 22 (5) (2005) 414–421.

[19] K. Goossens, J. van Meerbergen, A. Peeters, R. Wielage, Networks on silicon: combining best-effort and guaranteed services, In: Proc. Design, Automation and Test in Europe Conf. and Exhibition, 2002, pp. 423–425.

[20] W.-H. Hu, S.E. Lee, N. Bagherzadeh. DMesh: a diagonally-linked mesh Network-on-Chip architecture, in: NoCArc, First International Workshop on Network on Chip Architectures to be held in conjunction with MICRO-41, 2008.

[21] M. Igarashi, T. Mitsuhashi, A. Le, S. Kazi, Y.-T. Lin, A. Fujimura, S. Teig, A diagonal-interconnect architecture and its application to RISC core design, in: Solid-State Circuits Conference, 2002. Digest of Technical Papers. ISSCC. 2002 IEEE International, vol. 1, pp. 210–460, 2002.

[22] A.B. Kahng, B. Li, L.-S. Peh, K. Samadi, ORION 2.0: a fast and accurate NoC power and area model for early-stage design space exploration, in: Proc. DATE '09. Design, Automation and Test in Europe Conf. and Exhibition, 2009, pp. 423–428.

[23] N. Kavaldjiev, G.J. Smit, P.T. Wolkotte, P.G. Jansen, Providing QoS guarantees in a NoC by virtual channel reservation, in: K. Bertels, J. Cardoso, S. Vassiliadis (Eds.), Reconfigurable Computing: Architectures and Applications, Lecture Notes in Computer Science, vol. 3985, Springer-Verlag, Heidelberg, Germany, 2006. pp. 299–310.

[24] N. Kavaldjiev, G.J.M. Smit, P.G. Jansen, P.T. Wolkotte, A virtual channel network-on-chip for GT and BE traffic, in: Proc. IEEE Computer Society Annual Symp. Emerging VLSI Technologies and Architectures, vol. 00, 2006.

[25] W.E. Leland, M.S. Taqqu, W. Willinger, D.V. Wilson, On the self-similar nature of Ethernet traffic (extended version), IEEE/ACM Transactions on Networking 2 (1) (1994) 1–15.

[26] X. Lin, P.K. McKinley, L.M. Ni, The message flow model for routing in wormhole-routed networks, IEEE Transactions on Parallel and Distributed Systems 6 (7) (1995) 755–760.

[27] S. Ma, N. Enright Jerger, Z. Wang, DBAR: an efficient routing algorithm to support multiple concurrent applications in networks-on-chip, in: Proceedings of the 38th Annual International Symposium on Computer Architecture, ISCA '11, ACM, New York, NY, USA, 2011, pp. 413–424.

[28] A. Martinez, P. Garcia, F. Alfaro, J. Sanchez, J. Flich, F. Quiles, J. Duato, Towards a cost-effective interconnection network architecture with QoS and congestion management support, Berlin, Germany, 2006, pp. 884–895.

[29] A. Martinez, R. Martinez, F.J. Alfaro, J.L. Sánchez, A low-cost strategy to provide full QoS support in advanced switching networks, Journal of Systems Architecture 53 (2007) 355–368.

[30] A. Martinez-Vicente, P. Garcia, F. Alfaro, J. Sanchez, J. Flich, F. Quiles, and J. Duato, Integrated QoS provision and congestion management for interconnection networks, in: Euro-Par 2007. Parallel Processing, Proceedings 13th International Euro-Par Conference, LNCS, vol. 4641, Berlin, Germany, 2007, pp. 837–847.

[31] A. Martínez, F.J. Alfaro, J.L. Sánchez, J. Duato, Providing full QoS support in clusters using only two VCs at the switches, in: Proceedings of the 12th International Conference on High Performance Computing (HiPC), 2005, pp. 158–169.

[32] M. Millberg, E. Nilsson, R. Thid, A. Jantsch, Guaranteed bandwidth using looped containers in temporally disjoint networks within the nostrum network on chip, in: Proc. Design, Automation and Test in Europe Conf. and Exhibition, vol. 2, 2004, pp. 890–895.

[33] C. Minkenberg, R.P. Luijten, F. Abel, W. Denzel, M. Gusat, Current issues in packet switch design, SIGCOMM Computer Communication Review 33 (2003) 119–124.

[34] L.M. Ni, P.K. McKinley, A survey of wormhole routing techniques in direct networks, Computer 26 (2) (1993) 62–76.

[35] E. Nilsson, M. Millberg, J. Oberg, A. Jantsch, Load distribution with the proximity congestion awareness in a network on chip, in: Proc. Design, Automation and Test in Europe Conf. and Exhibition, 2003, pp. 1126–1127.

[36] G. Nychis, C. Fallin, T. Moscibroda, O. Mutlu, Next generation on-chip networks: what kind of congestion control do we need?, in: Proceedings of the Ninth ACM SIGCOMM Workshop on Hot Topics in Networks, Hotnets '10, ACM, New York, NY, USA, 2010, pp 12:1–12:6.

[37] U.Y. Ogras, R. Marculescu, Prediction-based flow control for network-on-chip traffic, in: Proc. 43rd ACM/IEEE Design Automation Conf., 2006, pp. 839–844.

[38] M.S. Taqqu, W. Willinger, R. Sherman, Proof of a fundamental result in self-similar traffic modeling, SIGCOMM Computer Communication Review 27 (1997) 5–23.

[39] S.L. Teig, The X architecture: not your father's diagonal wiring, in: Proceedings of the 2002 international workshop on system-level interconnect prediction, SLIP '02, ACM, New York, NY, USA, 2002, pp. 33–37.

[40] W. Trumler, S. Schlingmann, T. Ungerer, J. Bahn, N. Bagherzadeh, Self-optimized routing in a Network on-a-Chip, in: M. Hinchey, A. Pagnoni, F. Rammig, H. Schmeck (Eds.), Biologically-Inspired Collaborative Computing, IFIP International Federation for Information Processing, vol. 268, Springer, Boston, 2008, pp. 199–212.

[41] J.W. van den Brand, C. Ciordas, K. Goossens, T. Basten, Congestion-controlled best-effort communication for Networks-on-Chip, in: Proc. Design, Automation & Test in Europe Conf. & Exhibition DATE '07, 2007, pp. 1–6.

[42] S. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, P. Iyer, A. Singh, T. Jacob, S. Jain, S. Venkataraman, Y. Hoskote, N. Borkar, An 80-Tile 1.28TFLOPS Network-on-Chip in 65 nm CMOS, in: Proc. Digest of Technical Papers. IEEE Int. Solid-State Circuits Conf. ISSCC 2007, 2007, pp. 98–589.

[43] G. Varatkar, R. Marculescu, Traffic analysis for on-chip networks design of multimedia applications, in: Proc. 39th Design Automation Conf., 2002, pp. 795–800.

[44] P. Vellanki, N. Banerjee, K.S. Chatha, Quality-of-Service and error control techniques for mesh-based network-on-chip architectures, Integration, the VLSI Journal 38 (2005) 353–382.

[45] C. Wang, N. Bagherzadeh, Scalable load balancing congestion-aware Network-on-Chip router architecture, Journal of Computer and System Sciences 79 (2013) 403–514.

[46] C. Wang, W.-H. Hu, N. Bagherzadeh, Congestion-aware Network-on-Chip router architecture, in: Proc. 15th CSI Int. Computer Architecture and Digital Systems (CADS) Symp., 2010, pp. 137–144.

[47] C. Wang, W.-H. Hu, S.E. Lee, N. Bagherzadeh, Area and power-efficient innovative congestion-aware Network-on-Chip architecture, Journal of Systems Architecture 57 (2011) 24–38.
[48] D. Wu, B.M. Al-Hashimi, M.T. Schmitz, Improving routing efficiency for network-on-chip through contention-aware input selection, in: Proc. Asia and South Pacific Conf. Design Automation, 2006.

**Chifeng Wang** received a Ph.D. degree from University of California, Irvine in 2012 and his BS and MS degrees from National Central University, Taoyuan, Taiwan in 1999 and 2001, respectively. From 2001 to 2007, he was a researcher with Elan Microelectronics Corporation, where he was involved in the research and development of high performance microprocessor and system-on-chip. His research interests are in the areas of power-aware on-chip interconnection networks, low power chip multiprocessor design, especially in QoS and fault-tolerance router design for Network-on-Chip.

*Nader Bagherzadeh* is a professor of computer engineering in the department of electrical engineering and computer science at the University of California, Irvine, where he served as a chair from 1998 to 2003. Dr. Bagherzadeh has been involved in research and development in the areas of: computer architecture, reconfigurable computing, VLSI chip design, network-on-chip, sensor networks, and computer graphics since he received a Ph.D. degree from the University of Texas at Austin in 1987.

Professor Bagherzadeh has published more than 200 articles in peer-reviewed journals and conferences. He has trained hundreds of students who have assumed key positions in software and computer systems design companies in the past twenty years. He has been a PI or Co-PI on more than 4 million dollars worth of research grants for developing next generation computer systems for applications in general purpose computing and digital signal processing.