IEICE TRANS. INF. & SYST., VOL.E95–D, NO.4 APRIL 2012

PAPER

Support Efficient and Fault-Tolerant Multicast in Bufferless Network-on-Chip

Chaochao FENG[†a)], ***Student Member***, **Zhonghai LU**[††], **Axel JANTSCH**[††], **Minxuan ZHANG**[†], ***Nonmembers***, ***and*** **Xianju YANG**[†], ***Student Member***

SUMMARY In this paper, we propose three Deflection-Routing-based Multicast (DRM) schemes for a bufferless NoC. The DRM scheme without packets replication (DRM_noPR) sends multicast packet through a non-deterministic path. The DRM schemes with adaptive packets replication (DRM_PR_src and DRM_PR_all) replicate multicast packets at the source or intermediate node according to the destination position and the state of output ports to reduce the average multicast latency. We also provide fault-tolerant supporting in these schemes through a reinforcement-learning-based method to reconfigure the routing table to tolerate permanent faulty links in the network. Simulation results illustrate that the DRM_PR_all scheme achieves 41%, 43% and 37% less latency on average than that of the DRM_noPR scheme and 27%, 29% and 25% less latency on average than that of the DRM_PR_src scheme under three synthetic traffic patterns respectively. In addition, all three fault-tolerant DRM schemes achieve acceptable performance degradation at various link fault rates without any packet lost.

key words: *bufferless network-on-chip, deflection routing, multicast, fault-tolerance*

1. Introduction

As hundreds of processors can be integrated into a single chip, the design focus changes from a traditional processing-centric to a communication-centric one [1]. Network-on-Chip (NoC) provides a promising solution to address the global communication in Multi-Processors System-on-Chip (MPSoC) [2]. Compared with the traditional bus-based interconnection architecture, NoC has higher performance and better scalability. State-of-the-art NoCs have already provided high throughput and low latency for one-to-one (unicast) traffic. In order to run various scientific applications on an NoC-based MPSoC, fast and efficient collective communications (such as multicast and broadcast) must also be supported. In an MPSoC system, the cache-coherent shared memory protocols (e.g. directory-based or token-based) require one-to-many or one-to-all communications to maintain the ordering of requests or invalidate shared data on different cache nodes [3]. It has been stated that the multicast traffic accounts for 5-10% of the total network traffic in communication traces of different cache coherence protocols and has a serious impact on system performance [4]. Thus, supporting efficient multicast in NoC can improve the performance of the MPSoC system significantly.

Furthermore, with the technology scaling down to the nanometer domain, shrinking transistor sizes, smaller interconnect features, lower power voltages and higher operating frequencies seriously affect the reliability of CMOS VLSI circuits [5]. It is necessary to provide reliable communication in NoC through fault-tolerant routing. However, due to the existing faulty links, the network changes from regular topology to irregular topology, which makes it difficult to implement fault-tolerant multicast communication without deadlock and livelock.

Previous on-chip interconnection networks adopt the wormhole/virtual-channel router which contains buffers in each input port to buffer the packets (or flits) transmitted in the network. Although buffers in each router can help to improve the bandwidth efficiency of the network, they also consume significant energy and chip area [6]. Recently, bufferless NoC has become a potential alternative to the traditional virtual-channel-based NoC [6]–[8]. The bufferless router utilizes deflection routing to remove the need for buffers, which has two benefits: reduced hardware cost and simplicity in design. Unfortunately, in bufferless NoC, packets do not have to wait in a router and deflection routing makes the routing path unpredictable, so it is impossible to apply existing multicast algorithms used in networks with buffers.

To support efficient multicast communication in bufferless NoC, we propose three Deflection-Routing-based Multicast (DRM) schemes— DRM scheme without packets replication (DRM_noPR), DRM scheme with packets replication at the source node (DRM_PR_src) and DRM scheme with packets replication at both source and intermediate nodes (DRM_PR_all). We also provide fault-tolerant supporting in these schemes through a reinforcement-learning-based method to reconfigure the routing table to tolerate permanent faulty links in the network. The DRM_noPR scheme behaves like a path-based multicast method, while the DRM_PR_src and DRM_PR_all schemes replicate multicast packets at the source or intermediate node according to the destination position and the state of output ports. Simulation results illustrate that the DRM_PR_all scheme achieves 41%, 43% and 37% less latency on average than that of the DRM_noPR scheme and 27%, 29% and 25% less latency on average than that of the DRM_PR_src scheme under three synthetic traffic patterns respectively. In the pres-

Manuscript received April 18, 2011.
Manuscript revised November 15, 2011.

[†]The authors are with National Laboratory for Parallel and Distributed Processing, School of Computer, National University of Defense Technology, China.
[††]The authors are with Department of Electronic Systems, Royal Institute of Technology, Sweden.
a) E-mail: fengchaochao@nudt.edu.cn

DOI: 10.1587/transinf.E95.D.1052

ence of faulty links, all three fault-tolerant DRM schemes achieve acceptable performance degradation at various link fault rates without any packet lost.

The rest of paper is organized as follows. The related work is reviewed in Sect. 2. Section 3 describes the bufferless NoC architecture. The detailed deflection-routing-based multicast schemes are proposed in Sect. 4. In Sect. 5, simulation experimental results are presented and analyzed, followed by the conclusion and future work in Sect. 6.

2. Related Work

Existing multicast mechanisms can be classified as *unicast-based*, *path-based* and *tree-based*. In unicast-based multicast scheme [9], a multicast packet is divided into a sequence of unicast packets at the source node and sent to destination nodes separately. It can be simply implemented on the existing unicast routers without making any changes. However, such a scheme suffers from large network latency and high power consumption due to the fact that multiple copies of the same packet are injected into the network.

Path-based multicast routing selects a path to avoid deadlock and thus routes the multicast packet to each destination sequentially along the path until the last node is reached. A connection-oriented multicast has been proposed in [10] to send a single copy of multicast packets to multiple destinations along a pre-established path. It is simple to implement path-based multicast in hardware. However, if the number of destination nodes is large, a multicast packet will travel a long path which leads to a higher latency than tree-based multicast.

In tree-based multicast algorithm, the multicast packet is delivered along a common path as far as possible and is replicated at intermediate router to generate branches of the tree when necessary. The Virtual Circuit Tree Multicasting (VCTM) mechanism [4] is a typical tree-based method. Before sending multicast packets, VCTM needs to send a setup packet to build a multicast tree stored in a VCT table. Similar as VCTM to construct the multicast tree through a setup process, two power-efficient tree-based multicast algorithms, which consume less power than the XY tree-based algorithm, have been proposed in [11]. However, the setup process increases the multicast latency and extra storages are needed to maintain the tree information, which makes it not scalable to large networks. The Recursive Partition Multicast (RPM) [12] selects intermediate replication nodes based on the global distribution of destinations in the network. This scheme provides more path diversities and performs better than VCTM, but it is not implemented in hardware. A fully-adaptive multicast mechanism is proposed on the Multicast Rotary Router (MRR) [13]. The mechanism behaves like a tree-based multicast policy at low or medium loads and like a path-based one when the network reaches the saturation point. However, it also needs large buffers to store routing information and is too complex to implement in hardware.

Supporting fault-tolerance is essential for NoC to achieve reliable communication. Fault-tolerant routings have been extensively studied in NoC [14]–[16]. However, research on fault-tolerant multicast for NoC is still in its infancy. Several previous works focus on fault-tolerant multicast for general interconnection networks (e.g. unicast-based [17] and tree-based [18]). For NoC, bLBDR [3] is proposed to perform multicast operations using a broadcast within a sub-network. It can provide limited fault-tolerant capability to handle rectangular faulty blocks, however, the broadcast nature makes the network congested.

To the best of our knowledge, no other bufferless NoC supports multicast and achieves fault-tolerance at the same time. In this paper, we focus on supporting efficient multicast in bufferless NoC, propose three deflection-routing-based multicast (DRM) schemes and also provide fault-tolerant supporting in DRM to improve the reliability.

3. Bufferless NoC Architecture

3.1 General Bufferless Router Architecture

The bufferless NoC architecture in this paper is based on a 2D mesh topology, Nostrum NoC [19]. Figure 1 shows the general bufferless router architecture. There are n input/output ports in each router. For 2D mesh, n is 5. Each input port has only one input register, so the packet is not buffered in the router. Deflection routing is used to route packets. When two or more packets will be routed through a common productive port, through which leads to a shortest path to the destination, only one packet can occupy the productive port, other packet(s) will be deflected to a non-productive port. In order to limit the number of misroutings to avoid livelock, arriving packets must be sorted by priority before output port allocation. The priority is decided according to the number of hops the packet has been routed in the network, which means the oldest packet has the highest priority. The router can handle at most four packets at the same cycle and make output port allocation for each packet from the highest priority to the lowest.

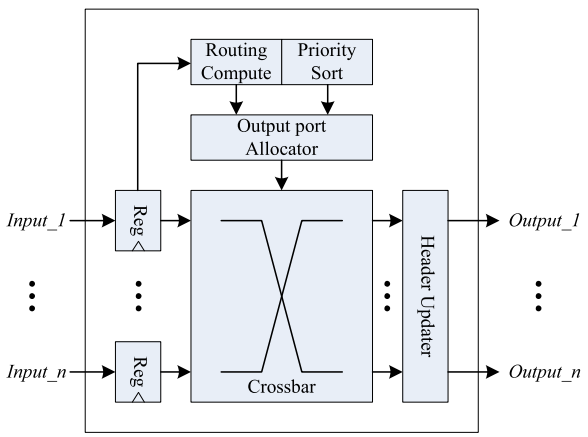
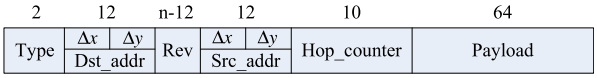
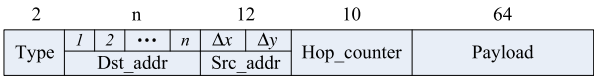


Fig. 1 General bufferless router architecture.



(a) Unicast packet



(b) Multicast packet

Fig. 2 Unicast and multicast packets format.

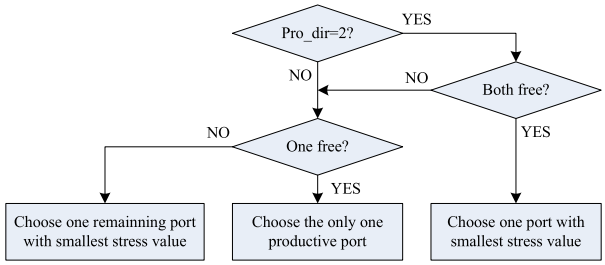


Fig. 3 Routing decision process for deflection routing.

3.2 Packet Format

The router supports two packet types: unicast and multicast. Figures 2 (a) and (b) show the format of two packet types respectively. The packet fields are explained as follows:

- ***Type*** field (2 bits): indicate the type of the packet ("01": unicast packet; "10": multicast packet; "00"/"11": invalid packet).
- ***Dst_addr*** field: for unicast packet, relative address to the destination is used to encode this field. It has 12 bits (6 bits for row and column addresses respectively). For multicast destination address, bit string encoding is used. The bit string is an n-bit vector, where n is the number of nodes in the network. A bit of '1' in this string means the corresponding node is one of destinations.
- ***Rev*** field: in order to maintain the same bit width of the two packet types, a reserved field, which has n-12 bits (n is the length of *Dst_addr* field in multicast packet), is introduced in the unicast packet.
- ***Src_addr*** field (12 bits): denote the relative address to the source node (6 bits for row and column addresses respectively).
- ***Hop_counter*** field (10 bits): record the number of hops the packet has been routed. It is used as packet priority to avoid livelock.
- ***Payload*** field: the packet payload has 64 bits, which can be straightforwardly extended to contain more bits according to different applications and architecture needs.

The header update module (shown in Fig. 1) is responsible for updating the address and hop counter fields of the two packet types. When a multicast packet reaches a destination, the bit in the destination bit string corresponding to that destination node must be reset to '0' and the *Src_addr* is updated with the relative address from the next router to the source node. When a unicast packet has passed a router, the *Dst_addr* and *Src_addr* fields are updated with the relative address from the next router to the destination/source node. In both cases, 1 hop is added to the hop counter field.

4. Deflection-routing-based Multicast (DRM) Schemes

In this section, we propose three different Deflection-Routing-based Multicast (DRM) schemes and also provide fault-tolerant supporting in DRM schemes to protect against permanent faulty links in the network.

4.1 Baseline Deflection Routing

Deflection routing is a non-minimal fully adaptive routing algorithm, which can be used in any topology as long as the number of input ports is equal to the number of output ports. A load-aware routing selection mechanism [20] is introduced in the original deflection routing to balance the traffic load in the network. The stress value, which is the number of packets processed by neighboring routers in the last 4 cycles, is transmitted between neighboring routers. In 2D mesh, when a packet reaches a router, there are at most two productive directions to the destination node. The productive direction is calculated according to the row and column relative addresses to the destination. The routing decision process is shown in Fig. 3. If both productive directions are free, the router will choose one of them with the smallest stress value to route the packet. If the productive directions are occupied by other higher priority packets, the router will choose a remaining free port with the smallest stress value to deflect the packet away from the shortest path to the destination node.

4.2 DRM without Packets Replication

Due to the fact of unpredictable routing path in deflection routing, it is impossible to apply existing multicast algorithms used in virtual-channel/wormhole router. We propose a non-deterministic path-based DRM scheme without packets replication (DRM_noPR). Different from the original path-based multicast, the multicast packet will be routed to each destination along a non-deterministic path in the DRM_noPR scheme. When a multicast packet arrives at a router, the router always selects a destination with the minimum manhattan distance to the current router from the destination bit string as the best candidate to calculate the productive direction(s). The packet does not have to choose another best candidate after arriving at the first best candidate. As the packet may be deflected way from the shortest path to the destination, the best candidate may change dynamically during the routing process.

The pseudo code of the routing computation function for the DRM_noPR scheme is shown in Fig. 4. The function first collects multicast destination node IDs into a vec-

Routing computation function for DRM without packets replication
Input: Multicast destination address *dst_addr*
Coordinate of current node (x_c, y_c)
Output: Productive direction set {$d_{productive}$}

1: $i \leftarrow 0$
2: ***for*** j in 0 to N-1 ***loop*** //N is the number of nodes in the network
3: ***if*** $dst\_addr[j]$ = '1' ***then***
4: $dst\_id\_vector[i] \leftarrow j$
5: $i \leftarrow i + 1$
6: ***end if***
7: ***end loop***
8: $best\_candidate\_id \leftarrow dst\_id\_vector[0]$
9: $(x_d, y_d) \leftarrow$ get_dst_coordinate($dst\_id\_vector[0]$)
10: $distance \leftarrow$ get_manhattan_distance(x_d, y_d, x_c, y_c)
11: ***for*** j in 1 to i-1 ***loop***
12: $(x_d, y_d) \leftarrow$ get_dst_coordinate($dst\_id\_vector[j]$)
13: ***if*** distance > get_manhattan_distance(x_d, y_d, x_c, y_c) ***then***
14: distance $\leftarrow$ get_manhattan_distance(x_d, y_d, x_c, y_c)
15: $best\_candidate\_id \leftarrow dst\_id\_vector[j]$
16: ***end if***
17: ***end loop***
18: {$d_{productive}$} $\leftarrow$ get_productive_direction($best\_candidate\_id$, x_c, y_c)

Fig. 4 Routing computation function for DRM without packets replication.

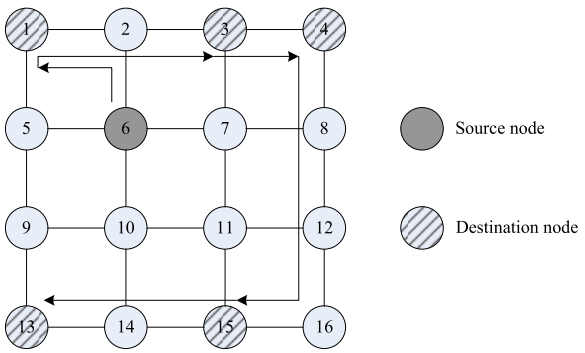


Fig. 5 Routing example of DRM without packets replication.

tor *dst_id_vector* (Step 1-7), then sets the first element of *dst_id_vector* as the best candidate node ID and calculates the manhattan distance between the current node and the best candidate node (Step 8-10). After that, the best candidate node with the minimum manhattan distance to the current node is found (Step 11-17). Finally, the productive direction(s) is calculated according to the position of the best candidate node to the current node (Step 18).

Figure 5 shows a multicast routing example of the DRM_noPR scheme. Node 6 sends a multicast packet to nodes 1, 3, 4, 13 and 15. Node 1 is chosen as the first best candidate since it has the minimum manhattan distance 2 to the source node 6. After the packet is sent to node 1, node 3 is chosen as the second best candidate. Without contention, the multicast path requires 11 link traversals in total. The path shown in Fig. 5 is not the only one path since the packet may be deflected due to contention.

4.3 DRM with Adaptive Packets Replication

As the number of multicast destination size increases, the long path of DRM without packets replication will lead to large multicast latency. To solve this problem, we propose two DRM schemes with adaptive packets replication: one can only replicate packets at the source node (DRM_PR_src) when the multicast packet is being injected into the network; the other can replicate packets at both source and intermediate nodes (DRM_PR_all). The multicast packet is replicated according to a region partition of the network. A region partition policy is proposed to divide the 2D mesh network into at most 4 regions according to the position of the source node, as shown in Fig. 6. Different from the region partition policy in [12] dividing the network into at most 8 regions, each region in our partition policy only corresponds to one direction. For the region partition policy in [12], if some destinations in a region can be reached through more than one direction, replication priority rules must be used for each direction to avoid redundant replication. While our region partition policy does not need replication priority rules and multicast packets can be replicated without destination overlapping.

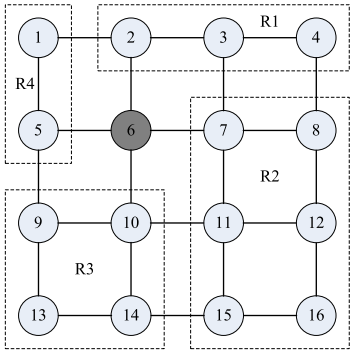


Fig. 6 Region partition in 2D mesh NoC.

For the DRM_PR_src scheme, when a multicast packet is being injected into the network from the local input port, if there is only one free output port, the packet will not be replicated and routed the same way as the DRM scheme without packets replication. If the number of free output ports is more than one and destinations fall into different regions corresponding to the free output ports, the destinations will be grouped according to the region partition and the multicast packet will be replicated and routed through different output ports. For the DRM_PR_all scheme, if the number of free output ports exceeds the number of packets to handle, the multicast packet arriving at the intermediate router can be replicated according to the destination distribution.

The pseudo code of adaptive packets replication algorithm is shown in Fig. 7. If there is no free output port left, the multicast packet will not be replicated. *Free_port* is a 4-bit vector, each bit of which indicates the corresponding output port is free or not. *Output_prio*, which is calculated according to the stress value, sorts the priority of the output port from the highest priority to the lowest (e.g. *output_prio*[0] and [3] represent output directions with the highest and lowest priority respectively). The router al-

Adaptive packets replication algorithm
Input: Multicast destination address *dst_addr*,
Number of free output ports *free_port_num*,
Free output port status *free_port*[4],
Output port priority *output_prio*[4]
Output: Replicated packet address *replicate_pkt_addr*[4]

```
1:  copy_num ← 0
2:  if free_port_num = 0 then
3:    Do not replicate the multicast packet
4:  else
5:    for i in 0 to 3 loop
6:      if free_port[output_prio[i]] = '1' and copy_num < free_port_num then
7:        replicate_pkt_addr[output_prio[i]] ← replicate_addr_gen(dst_addr,output_prio[i])
8:        dst_addr ← dst_addr and (not replicate_pkt_addr[output_prio[i]])
9:        if replicate_pkt_addr[output_prio[i]] ≠ 0 then
10:         copy_num ← copy_num + 1
11:       end if
12:     end if
13:   end loop
14: end if
```

Fig. 7 Adaptive packets replication algorithm.

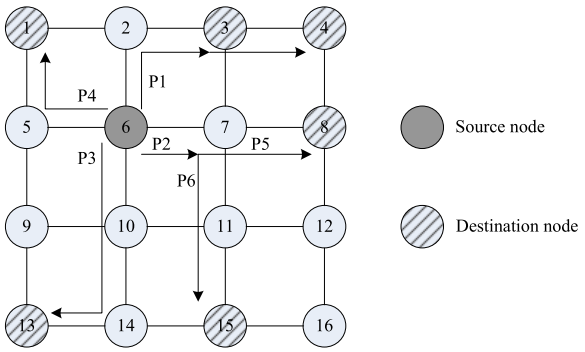


Fig. 8 Routing example of DRM with packets replication.

ways chooses a free output port with the highest priority (the smallest traffic load) to replicate and delivery a copy of the multicast packet. The function *replicate_addr_gen* generates the address of the replicated packet. If a multicast destination falls into the region corresponding to the output direction *output_prio*[*i*], it will be added into the address of the replicated packet. After replication, the original destination address will be updated and the number of packet copies will increase by 1.

Figure 8 shows a multicast routing example of the DRM schemes with adaptive packets replication. Node 6 sends a multicast packet to nodes 1, 3, 4, 8, 13 and 15. Nodes 3 and 4 belong to region 1, nodes 8 and 15 belong to region 2, nodes 13 and 1 belong to region 3 and 4 respectively. For the DRM_PR_src scheme, at most 4 packets (P1, P2, P3 and P4) can be replicated if all output ports are free. For the DRM_PR_all scheme, if the *East* and *South* output ports are free at router 7, P2 can be replicated as P5 and P6 further.

4.4 Fault-Tolerance Supporting in DRM

To support fault-tolerance, we consider the faults as completely broken links (permanent faults). In each router, a 4-bit fault vector is used to represent the fault states of its four links. Deflection router requires the number of input ports should be equal to the number of output ports, so broken links are assumed to be bidirectional. In [21], a reconfigurable fault-tolerant deflection routing (FTDR) algorithm based on reinforcement learning has been proposed to handle faulty regions which do not disconnect the whole network. Each router contains an $n \times 4$ routing table which is construct by the minimum number of hops to all destinations in 2D mesh network from its four output ports (n is the number of nodes in the network). Assuming each routing table entry has d bits, the routing table has a cost of $n \times 4 \times d$ bits. As the size of the network increases, the routing table size will increase. To reduce the cost of routing table, the network can be virtually partitioned into small regions and use the hierarchical routing table to route packets as described in [21]. Figure 9 shows a routing table of router 5 in a 3×3 2D mesh. The routing table is reconfigured by Eq. (1). $Q^x(d, y)$ denotes the minimum number of hops from x to d through neighbor y. When router x sends a packet to d through y, y will return 1 plus the minimum number of hops from itself to d back to x to reconfigure the corresponding routing table entry of x.

	North	East	South	West
Number of hops to R1	2	4	4	2
Number of hops to R2	1	3	3	3
Number of hops to R3	2	2	4	4
Number of hops to R4	3	3	3	1
Number of hops to R5	0	0	0	0
Number of hops to R6	3	1	3	3
Number of hops to R7	4	4	2	2
Number of hops to R8	3	3	1	3
Number of hops to R9	4	2	2	4

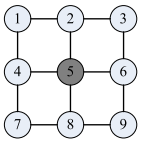


Fig. 9 Routing table of router 5 in a 3×3 mesh.

$$Q_t^x(d, y) = 1 + \min_z Q_{t-1}^y(d, z) \tag{1}$$

We extend the FTDR algorithm for the three DRM schemes to support multicast and fault-tolerance at the same time. The fault-tolerant DRM schemes can tolerate permanent faulty links without any packet lost. For both unicast and multicast packets, after receiving the packet, the router will send 1 plus the minimum number of hops from itself to destination back to the packet sender to reconfigure its routing table. In original DRM schemes, the productive direction(s) for the multicast packet is calculated by finding one destination with the minimum manhattan distance to the current node as the best candidate. Due to the fact that existing faulty links make the network change from regular topology to irregular topology and the manhattan distance is no longer the minimum distance to the destination, the DRM schemes with fault-tolerance calculate the productive direction(s) for the multicast packet by searching for table entries corresponding to all destinations of the multicast packet and finding an entry with the minimum number of hops to one of destinations as the best candidate to get the productive direction(s).

Figure 10 shows a routing computation example in a 3×3 mesh with two faulty links. Here, router 4 sends a multicast packet to routers 3 and 9. Router 4 checks the routing table entries to R3 and R9 and finds that the minimum num-

	North	East	South	West
Number of hops to R1	1	∞	3	∞
Number of hops to R2	2	∞	4	∞
Number of hops to R3	3	∞	5	∞
Number of hops to R4	0	0	0	0
Number of hops to R5	3	∞	3	∞
Number of hops to R6	4	∞	4	∞
Number of hops to R7	3	∞	1	∞
Number of hops to R8	4	∞	2	∞
Number of hops to R9	5	∞	5	∞

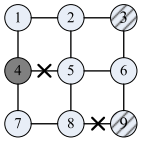


Fig. 10 Routing computation example in a 3×3 mesh with two faulty links.

ber of hops to the two destinations is 3 (from *North* output port to R3). So R3 is considered as the first best candidate and the *North* direction is set as the productive direction.

Besides the routing computation, the packet replication policy is also different from the original DRM schemes. Since the network with faulty links is no longer the regular 2D mesh, the multicast packet replication based on the region partition may lead to misrouting. So for DRM schemes with adaptive packets replication, if a destination of the multicast packet can be added into the replication address along a free output direction i, it must satisfy the following condition: the routing table entry from this direction i to the destination is the minimum number of hops from the current node to the destination.

4.5 Deadlock and Livelock Avoidance

Deflection routing is inherently *deadlock-free* due to the fact that packets never have to wait in a router. However, whenever a packet is deflected, it moves further away from its destination. Thus, livelock must be avoided by limiting the number of misroutings. In our multicast router, both unicast and multicast packets are prioritized based on its age (the number of hops already routed in the network). The age-based priority mechanism guarantees that the oldest packet can always win the link arbitration and eventually advance towards its destination deterministically. Once the oldest packet reaches its destination, another packet becomes the oldest. For the DRM schemes with fault-tolerance, it can be proved that the routing table will converge to the minimum number of hops to destination within a limited time in the presence of fault regions which do not disconnect the network [21]. Thus livelock can be avoided.

5. Experimental Evaluation

In this section, we use a cycle-accurate NoC simulator to evaluate the performance of the three DRM schemes with different synthetic traffic patterns.

5.1 Methodology

The experiments are performed on an 8×8 2D mesh network with 1-cycle router. The routing process can be divided into two steps. At first, the routing computation together with the input and output priority sorting are performed in parallel, and then the output port allocator makes output allocation for each packet based on the results of the first step. For the DRM router with adaptive packets replication, packets replication occurs at the moment of output allocation when at least one free output port left for replication. The DRM router without fault-tolerance uses routing computation function shown in Fig. 4 rather than a routing table to calculate the productive direction. Only the fault-tolerant DRM router uses a routing table for routing computation. Compared with the conventional unicast bufferless router, the DRM routers are more complicated due to increased routing computation for multicast packets and packets replication. In order to simplify the design, we use 1-cycle router without pipeline for the three kinds of DRM routers (DRM_noPR, DRM_PR_src, DRM_PR_all). Although the maximum achievable frequency can be enhanced by pipelining, the pipeline will increase the design complexity and single hop latency. The three kinds of DRM routers differ slightly in their routing computation process, thus it is reasonable to assume that their maximum operating frequencies are close to each other. In addition, as on-chip embedded applications are typically power constrained, routers may not necessarily run up to their maximum frequency. Therefore it can be necessary to have them running with the same reasonable speed.

A packet generator is attached to each router and uses a FIFO to buffer the packets which cannot be injected into the network immediately due to the fact that there is no free output port to route the packet. The traffic workloads contain both unicast and multicast packets. The number of destinations and the percentage of multicast traffic can be configured at the beginning of the simulation. For unicast traffic, three synthetic traffic patterns (uniform random, transpose and bit compliment) are used. In uniform random traffic, each resource node sends packets randomly to other nodes with an equal probability. For transpose traffic, resource node positioned at (x, y) sends packets to destination node (y, x) for all $x \neq y$. In bit compliment traffic, the six-bit source node ID $\{s_i | i \in [0, 5]\}$ sends packets to destination $\{\neg s_i | i \in [0, 5]\}$. For multicast packet, the destination positions are uniformly distributed.

We measure the *average packet latency* T which is calculated by Eq. (2) and measured in *cycle*, where T_{net} is the *network delivery time* (the hop count of a packet being routed) and T_{src} is the time a packet waiting in the source FIFO queue. We also measure the *link utilization*, which is the average percentage of used links in total links. Link utilization can reflect the power consumption indirectly. The more number of links is active, the higher power consumption is consumed.

$$T = T_{net} + T_{src} \tag{2}$$

5.2 Performance with No Faulty Links

In this subsection, we measure the performance of the three DRM schemes in the network with no faulty links. Figures 11 (a)-(c) illustrate the average packet latency of the

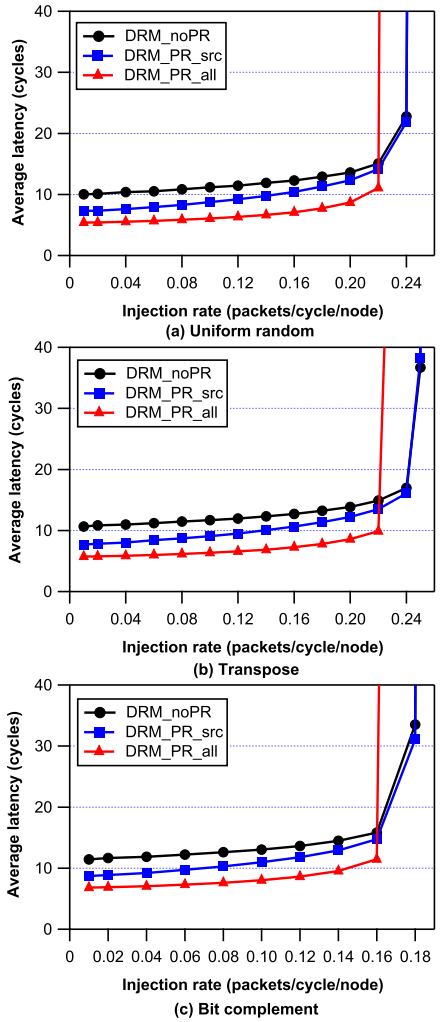


Fig. 11 Average packet latency with three synthetic traffic patterns (10% multicast traffic, 8 destinations).

three DRM schemes with three synthetic traffic patterns at various injection rates respectively. The average packet latency is calculated for both unicast and multicast packets. The multicast traffic is 10% of the total traffic and the number of multicast destinations is 8. It can be observed that the DRM_PR_all scheme achieves the smallest average latency among the three schemes. However, the network with the DRM_PR_all scheme reaches saturation point earlier than the other two schemes. In the case of the network not saturated, the average latency of the DRM_PR_src scheme is 18%, 20% and 17% less than that of the DRM_noPR scheme under three synthetic traffic patterns respectively. The DRM_PR_all scheme achieves 41%, 43% and 37% less latency on average than that of the DRM_noPR scheme and 27%, 29% and 25% less latency on average than that of the DRM_PR_src scheme under three synthetic traffic patterns respectively.

The results are consistent with our expectations. The DRM_PR_src scheme outperforms the DRM_noPR scheme at low or medium traffic loads and performs similar as the

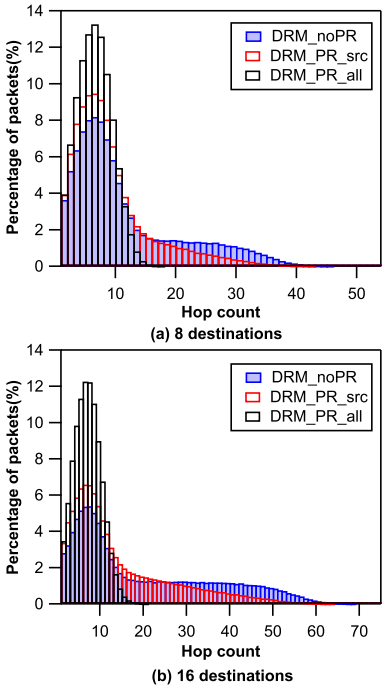


Fig. 12 Packet distribution based on hop count (10% multicast traffic).

DRM_noPR scheme when the network reaches saturation point. The reason lies in that at low or medium traffic loads, when the multicast packet is being injected into the network, more packet copies can be sent through free output ports which can reduce the multicast latency significantly. While at high traffic loads, because of the less chance to replicate packets at the injection time, the DRM_PR_src scheme performs similar as the scheme without packets replication. For the DRM_PR_all scheme, due to the fact that it can replicate multicast packets at both source and intermediate nodes, which increases the network loads, so the throughput of the network is a little bit lower than that of the other two schemes. However, under real application workloads the network is never fully loaded, so the DRM_PR_all scheme is a better alternative multicast solution for bufferless NoC than the other two schemes.

In addition to average latency, we also compare the packet distribution at the injection rate of 0.1 packets/cycle/node for the three schemes based on the hop counts under uniform random traffic with 10% multicast traffic of 8 and 16 destinations in Figs. 12 (a) and (b) respectively. Due to the packets replication at both source and intermediate nodes, the DRM_PR_all scheme can save much more hop counts. The maximum hop counts for the DRM_PR_all scheme are 20 and 25 for multicast traffic of 8 and 16 destinations respectively, while for the DRM_noPR scheme, the maximum hop counts are 54 and 75 respectively. Since as the number of destinations increases, the multicast packet for DRM_noPR scheme will be routed through a long path to reach all destinations. With limited packets replication, the DRM_PR_src scheme also suffers more hop counts than the DRM_PR_all scheme.

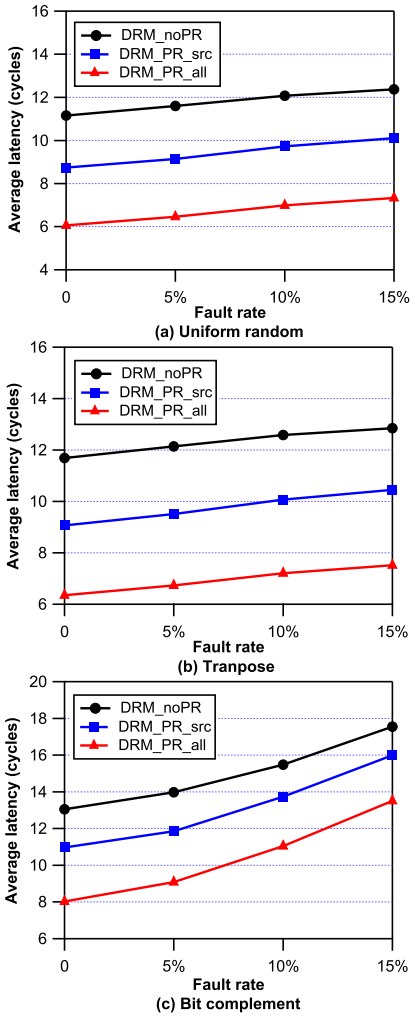


Fig. 13 Average packet latency with three synthetic traffic patterns at various link fault rates (10% multicast traffic, 8 destinations).

5.3 Performance with Faulty Links

In order to evaluate the performance of the three DRM schemes with fault-tolerant capability, we perform the simulation on the network with various link fault rates under three synthetic traffic patterns. Figures 13 (a)-(c) show the average packet latency of the three DRM schemes with link fault rates varying from 0 to 15%. The packet injection rate is 0.1 packets/cycle/node, at which the network does not reach the saturation point. The multicast traffic is 10% of the total traffic and the number of multicast destinations is 8. In the presence of faulty links, the DRM_PR_src scheme can achieve 20%, 20% and 12% less latency on average than that of the DRM_noPR scheme for the three traffic patterns respectively, while the DRM_PR_all scheme can achieve 42%, 43% and 28% less latency on average than that of the DRM_noPR scheme for the three traffic patterns respectively. All three fault-tolerant DRM schemes achieve acceptable performance degradation with the increasing link fault rate.

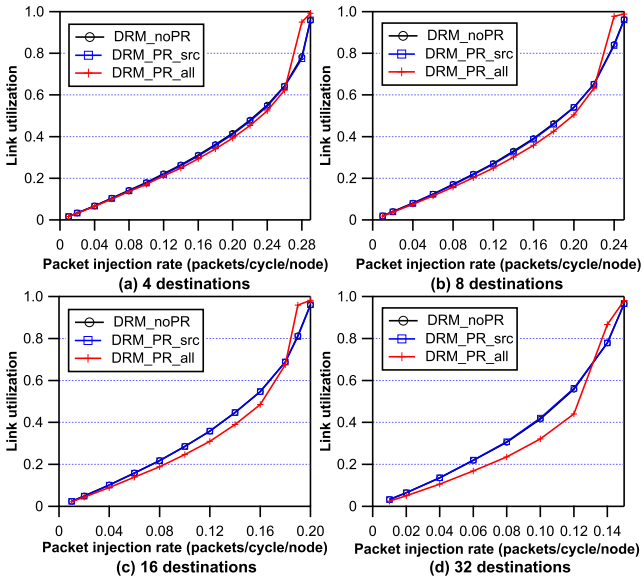


Fig. 14 Link utilization with different number of multicast destinations (10% multicast traffic).

5.4 Link Utilization

Figures 14 (a)-(d) show the link utilization with the number of multicast destinations increasing from 4 to 32 under uniform random unicast traffic with 10% multicast traffic. It can be seen that the DRM_noPR and DRM_PR_src schemes have the similar link utilization. Before the network reaches the saturation point, compared with the other two schemes, the DRM_PR_all scheme can save 5%, 7%, 11% and 22% of link utilization on average for different number of multicast destinations respectively. It can also be concluded that as the number of destinations increases from 4 to 32, the DRM_PR_all scheme can save more power consumption than that of the other two schemes.

5.5 Scalability

We also evaluate the performance of the three DRM schemes with different portion of multicast traffic, different number of multicast destinations and different network sizes under the uniform random unicast traffic pattern. Figure 15 shows the performance variation of the three DRM schemes with the portion of multicast traffic increasing from 10% to 40%. The number of multicast destinations is 8. As can be seen from this figure, the DRM_noPR and DRM_PR_src schemes achieve much larger average latency than the DRM_PR_all scheme with the increasing portion of multicast traffic. Figure 16 reveals the average latency of the three DRM schemes with the increasing number of multicast destinations. The portion of multicast destinations is 10%. As the number of multicast destinations increases from 4 to 32, the average latency of the DRM_PR_all scheme does not vary so much, while the average latencies of the DRM_noPR and DRM_PR_src schemes increase by up to 3

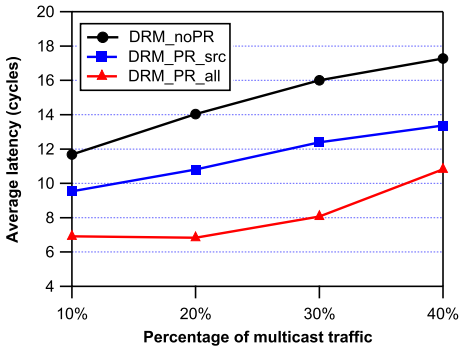


Fig. 15 Average packet latency with different percentage of multicast traffic (8 destinations).

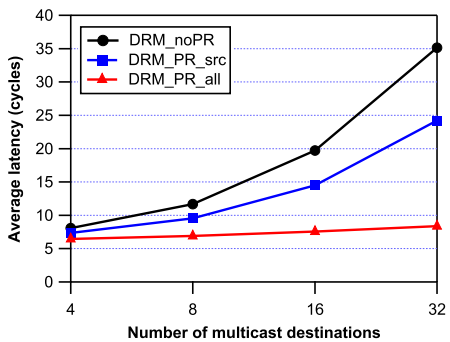


Fig. 16 Average packet latency with different number of multicast destinations (10% multicast traffic).

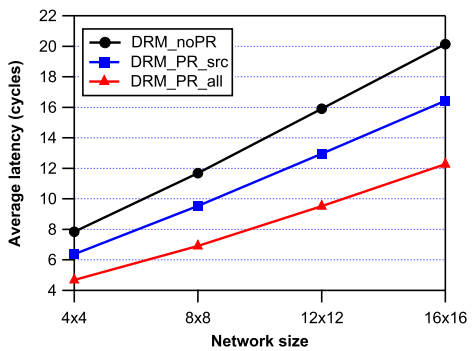


Fig. 17 Average packet latency under different network sizes (10% multicast traffic, 8 destinations).

and 2 times respectively. Figure 17 shows the average latency of the three DRM schemes under different network sizes with 10% multicast traffic and 8 destinations. The DRM_PR_all schemes can achieve 40% and 26% less latency on average than the DRM_noPR and DRM_PR_src schemes respectively. Compared with the other two DRM schemes, the DRM_PR_all scheme is more scalable.

6. Conclusion and Future Work

This paper proposes efficient multicast schemes in the bufferless NoC and also provides fault-tolerant supporting in these schemes to tolerate permanent faulty link without any packet lost. Specific contributions of this paper can be summarized as follows:

- Three deflection-routing-based multicast (DRM) schemes are propose in bufferless NoC to support efficient multicast communication. The DRM scheme without packets replication, which is simple in design, implements multicast through a non-deterministic path. The DRM schemes with adaptive packets replication at the source or intermediate node replicate multicast packets at the source or intermediate node according to the destination position and the state of output ports, which can reduce multicast latency significantly.
- The fault-tolerant DRM schemes, which reconfigure the routing table through a reinforcement learning method during packets transmission, can tolerate permanent faulty link at various fault rates with acceptable performance degradation.

Although the fault-tolerant DRM schemes can handle permanent faulty links efficiently, as the CMOS technology scaling down to 20 *nm* and below, transient faults, such as crosstalk and single-event upsets (SEUs), may cause significant problems in NoC. In future work, we will extend the fault-tolerant DRM schemes to handle transient faults as well.

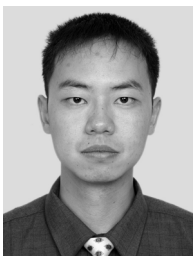
Acknowledgements

The research is partially supported by the National Natural Science Foundation of China, under Grant No.60970036, No.60873212 and No.61003301.

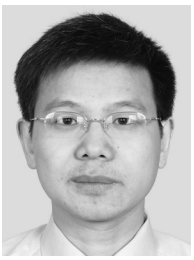
References

[1] T. Bjerregaard and S. Mahadevan, “A survey of research and practices of network-on-chip,” ACM Comput. Surv., vol.38, no.1, pp.1–54, June 2006.
[2] L. Benini and G. De Micheli, “Networks on chips: a new soc paradigm,” Computer, vol.35, no.1, pp.70–78, Jan. 2002.
[3] S. Rodrigo, J. Flich, J. Duato, and M. Hummel, “Efficient unicast and multicast support for cmps,” Proc. IEEE Computer Society, Int. Symposium on Microarchitecture, Washington, DC, USA, pp.364–375, 2008.
[4] N.E. Jerger, L.S. Peh, and M. Lipasti, “Virtual circuit tree multicasting: A case for on-chip hardware multicast support,” Proc. IEEE Computer Society, Int. Symposium on Computer Architecture, Washington, DC, USA, pp.229–240, 2008.
[5] C. Constantinescu, “Trends and challenges in vlsi circuit reliability,” IEEE Micro., vol.23, no.4, pp.14–19, 2003.
[6] T. Moscibroda and O. Mutlu, “A case for bufferless routing in on-chip networks,” Proc. IEEE Computer Society, Int. Symposium on Computer Architecture, pp.196–207, 2009.
[7] M. Hayenga, N.E. Jerger, and M. Lipasti, “Scarab: a single cycle adaptive routing and bufferless network,” Proc. IEEE Computer Society, Int. Symposium on Microarchitecture, pp.244–254, 2009.
[8] Z. Lu, M. Zhong, and A. Jantsch, “Evaluation of on-chip networks using deflection routing,” Proc. ACM, Great Lakes Symposium on VLSI, pp.363–368, July 2006.
[9] P.K. McKinley, H. Xu, L.M. Ni, and A.H. Esfahanian, “Unicast-based multicast communication in wormhole-routed networks,” IEEE Trans. Parallel Distrib. Syst., vol.5, no.12, pp.1252–1265, Dec. 1994.

[10] Z. Lu, B. Yin, and A. Jantsch, "Connection-oriented multicasting in wormhole-switched networks on chip," Proc. IEEE Computer Society, Annual Symposium on Emerging VLSI Technologies and Architectures, Washington, DC, USA, pp.205–210, 2006.
[11] W. Hu, Z. Lu, A. Jantsch, and H. Liu, "Power-efficient tree-based multicast support for networks-on-chip," Proc. 16th Asia and South Pacific Design Automation Conference, Piscataway, NJ, USA, pp.363–368, 2011.
[12] L. Wang, Y. Jin, H. Kim, and E.J. Kim, "Recursive partitioning multicast: A bandwidth-efficient routing for networks-on-chip," Proc. IEEE Int. Symposium on Networks-on-Chip, Washington, DC, USA, pp.64–73, 2009.
[13] P. Abad, V. Puente, and J.A. Gregorio, "Mrr: enabling fully adaptive multicast routing for cmp interconnection networks," Proc. IEEE Computer Society, Int. Symposium on High Performance Computer Architecture, pp.355–366, 2009.
[14] Z. Zhang, A. Greiner, and S. Taktak, "A reconfigurable routing algorithm for a fault-tolerant 2d-mesh network-on-chip," Proc. ACM/IEEE Design Automation Conference, pp.441–446, 2008.
[15] D. Fick, A. DeOrio, G. Chen, V. Bertacco, D. Sylvester, and D. Blaauw, "A highly resilient routing algorithm for fault-tolerant nocs," Proc. Design, Automation and Test in Europe Conference and Exhibition, pp.21–26, 2009.
[16] A. Kohler, G. Schley, and M. Radetzki, "Fault tolerant network on chip switching with graceful performance degradation," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.29, no.6, pp.883–896, 2010.
[17] D. Xiang, Y. Zhang, and J.G. Sun, "Unicast-based fault-tolerant multicasting in wormhole-routed hypercubes," J. Systems Architecture, vol.54, no.12, pp.1164–1178, Dec. 2008.
[18] J. Wu and X. Chen, "Fault-tolerant tree-based multicasting in mesh multicomputers," J. Computer Science & Technology, vol.16, no.5, pp.393–409, 2001.
[19] M. Millberg, E. Nilsson, R. Thid, S. Kumar, and A. Jantsch, "The nostrum backbone-a communication protocol stack for networks on chip," Proc. IEEE Computer Society, Int. Conference on VLSI Design, pp.693–696, 2004.
[20] E. Nilsson, M. Millberg, J. Oberg, and A. Jantsch, "Load distribution with the proximity congestion awareness in a network on chip," Proc. Design, Automation and Test in Europe Conference and Exhibition, pp.1126–1127, 2003.
[21] C. Feng, Z. Lu, A. Jantsch, J. Li, and M. Zhang, "A reconfigurable fault-tolerant deflection routing algorithm based on reinforcement learning for network-on-chip," Proc. 3rd Int. Workshop on Network on Chip Architectures, pp.11–16, 2010.



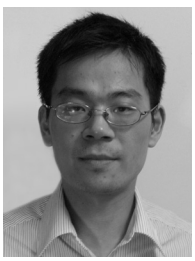
Chaochao Feng born in 1982, Ph.D. candidate at School of Computer, National University of Defense Technology, China. His current research interests include Network-on-Chip and microprocessor design.



Zhonghai Lu born in 1968, Ph.D., Associate professor at Department of Electronic Systems, Royal Institute of Technology, Sweden. His research interests include Network-on-Chip, hardware/software codesign and multiprocessor architecture.

Axel Jantsch born in 1962, Professor at Department of Electronic Systems, Royal Institute of Technology, Sweden. His research interests cover Network-on-Chip, embedded systems, VLSI design and system synthesis and validation.

Minxuan Zhang born in 1954, Professor at School of Computer, National University of Defense Technology, China. His research interests cover computer architecture, microprocessor design and VLSI design.



Xianju Yang born in 1980, Ph.D. candidate at School of Computer, National University of Defense Technology, China. His current research interests focus on microprocessor design.