Adaptive Partition-based Multicast Routing Scheme for Mesh-based Networks-on-Chip

Zheng Wang¹, Huaxi Gu¹, Yintang Yang², Haibo Zhang³, Yawen Chen³

¹State Key Laboratory of ISN, Xidian University, Xi'an, China

²Institute of Microelectronics, Xidian University, Xi'an, China

³The University of Otago, Otago, New Zealand

zhengwang@stu.xidian.edu.cn, {hxgu, ytyang}@xidian.edu.cn, {haibo, yawen}@cs.otago.ac.nz

Abstract—Network-on-Chips (NoCs) have become the mainstream for Chip Multi-Processors (CMPs) design. Multicast, a one-to-many communication pattern, is widely used in barrier/clock synchronization, multithreading programs and cache coherence protocols for CMPs. Even though several multicast routing algorithms have been proposed for CMPs, few can adaptively deal with heavy traffic loads. With the increase of multicast traffic load, deterministic routing schemes suffer from long latency and low throughput, whereas adaptive routing algorithms can improve the routing performance by providing multiple redundant paths. In this paper, we proposed a novel multicast routing algorithm based on partition to reduce the latency of multicast packets, by finding multiple routing paths and adaptively choosing available output ports based on the size of buffer space in downstream routers. We evaluate our scheme through simulations, and results show that, under various configurations, both latency and energy consumption have been significantly reduced in comparison with recent multicast routing schemes.

Keywords—Networks-on-Chip; multicast; adaptive; partition

I. INTRODUCTION

With the development of transistor technology, hundreds and even thousands of cores are going to be integrated in a single Chip Multi-processor (CMP) [2]. The efficiency of communication between cores depends on many factors, which has become a critical problem in the design of high-performance CMPs [3]. Traditional bus-based interconnection approaches cannot provide enough bandwidth, and thus is only suitable to deal with communication among a handful of cores. To overcome the drawback of bus interconnection, Network-on-Chips (NoCs) have been proposed as a promising solution to address the challenges of on-chip multi-core processor design [4] [5]. In NoCs, each tile can be an implementation of processors, DSP blocks, memory blocks, and other embedded reconfiguration modules [7]. Then the

regular tiles are connected into a network with a certain topology such as mesh, torus, and etc. [6]. NoCs is a flexible and scalable interconnection architecture that can connect over hundred and even thousands of cores. The throughput and bandwidth of NoCs are far larger than traditional buses, e.g., Intel Teraflop 80-core and Tiler 64-core [9] have gained large bandwidth by using NoCs. Also NoC has the advantage of lower power consumption in comparison with bus-based interconnections [4] [8].

In CMPs different cores need to exchange data with each other frequently. Hence, the performance of a CMP heavily depends on the underlying packet routing scheme. Communications in NoCs can be classified into unicast (one-to-one) traffic or multicast (one-to-many) traffic [10]. The support of multicast communication in NoCs is particularly important for applications such as reduction, permutation, segmented scan, replication [12], barrier synchronization [11], clock synchronization, multithreading programs, and cache coherence protocols of distributed shared-memory architectures in CMPs [13]. Traditional multicast routing schemes simply use multiple unicasts to realize a multicast, which is not efficient since a multicast packet with N destinations needs N unicast packets. The lack of efficient multicast routing support has become a bottleneck for NoCs since multicast traffic has a significant impact on the performance of NoCs [14].

In this paper, we proposed a new adaptive partition-based multicast routing algorithm without routing table for mesh-based NoCs based on the LADPM scheme proposed in [1]. Although LADPM is a novel and efficient approach to route multicast traffic, it lacks adaptation to traffic load. For each multicast packet, a router in LADPM provides only one output port and ignores buffer state and traffic condition of the downstream neighbor routers, which can lead to significant performance degradation under heavy traffic load. The basic principle of our scheme is to route each multicast packet to the corresponding port based on the relative position of each destination node. When there are more than one available routing paths without causing misrouting, the router makes routing decisions adaptively from these candidate ports based on the buffer state of the downstream routers. Simulation shows that, in an 8×8 2D-mesh network under high injection rates, our scheme can reduce the end-to-end communication delay by 43% and 54% in comparison with LADPM when the number of destination nodes is set to 8 and 4, respectively.

The initial idea of our routing algorithm was published in [15]. In this paper, most constraints on adaptive routing has been removed to achieve better adaptivity, and we further added new contributions that include (a) an efficient algorithm for routing multicast traffic using recursive partition; (b) a novel

and easy method for minimizing the link usage of a multicast tree; and (c) a minimal adaptive routing algorithm that can balance multicast traffic load.

The rest of this paper is organized as follows. Section II describes the related work and the background of multicast routing in NoCs. The proposed multicast routing algorithm is presented in detail in Section III. The simulation results and the comparison with previous work are presented in Section IV. Finally, Section V concludes this paper.

II. RELATED WORK

Multicast routing is a research topic that has been well studied for wired and wireless networks. Previous work on multicast routing can be classified into two categories: path-based and tree-based algorithms.

In path-based routing algorithms, each multicast packet is transmitted along a fixed path that contains all the destinations. The packet is duplicated at the intermediate router with one copy sent to the local IP cores. In this scheme, deadlock never occurs since there is no branch in the routing path. However, path-based routing algorithms suffer from long latency because all packets must be transmitted through routers in a fixed order.

In tree-based routing algorithms, each multicast packet is transmitted along a spanning tree with the source node as the root and the destination nodes as the leaves. Each path from the root to a destination node is a shortest path. This reduces the latency of transmitting a multicast packet. However, the branches of the spanning tree have to compete with each other, which can cause deadlocks [13]. Hence, deadlock-freedom methods must be designed. A hardware-based approach called MRR [16] was proposed by Abad et al. for multicast routing. It uses a multicast rotary router to route multicast traffic to avoid deadlocks. However, in some cases, it may cause long latency. In [14] T. Krishna et al. proposed a tree-based algorithm called FANOUT for routing multicast traffic. This method generates multicast trees randomly for all nodes in the network without taking into account the position of each multicast node. Since it considers all nodes in the network, FANOUT is a good scheme for balancing broadcast traffic. But it could occupy more links than other tree-based algorithms for ordinary multicast traffic that contains only a few nodes of the whole network, which makes it not suitable to route ordinary multicast traffic. Another tree-based multicast routing scheme called RPM [17] was designed based on partition and makes routing decisions based on the global distribution of the destination nodes. It uses two virtual networks to transmit up and down stream separately to avoid deadlocks. The defect of RPM's routing

scheme can lead to frequent inefficient routing decisions. Our work is also a tree-based routing scheme, but it uses a novel mechanism to make routing decisions based on the locations of each node to avoid inefficient routing paths.

Based on the adaptivity of the algorithms, the existing multicast routing schemes can be classified into deterministic algorithms and adaptive algorithms. The deterministic multicast routing algorithms make routing decisions based on the distribution of the destination nodes and the location of the source node without considering other factors such as congestion, traffic load and router buffer state. For a deterministic routing algorithm, the path generated by the routing algorithm for the same set of destinations is always the same. All the algorithms mentioned above are deterministic routing schemes. The performance of these deterministic routing algorithms decreases significantly under heavy traffic loads. Unlike the deterministic routing algorithms, the adaptive routing algorithms select output port from more than one candidate ports, and a selection module is used to choose one proper output port based on the network state. Some factors such as congestion information, buffer state of downstream router, and deadlock-free must be taken into account in the procedure of selecting one output port from the candidate ports. The adaptive routing algorithms can provide low latency on-chip packet transmission under heavy traffic loads. In [19] M. Ebrahimi et al. proposed an adaptive routing algorithm using Q-learning named HARAQ. This routing algorithm is a non-minimal routing algorithm that can provide several shortest alternative paths between the source and the destination nodes. The turn model is very flexible even if the 180 degree turns are permitted. They also proposed another non-minimal adaptive routing algorithm called HiPFaR that is fault-tolerant in [20]. A minimal adaptive routing algorithm called MAR [24] was also proposed by M. Ebrahimi et al for 3D NoCs. It is based on Hamiltonian path and can be used with three different partitioning methods – Two Block Partitioning (TBP), Recursive Partitioning (RP), and Vertical Block Partitioning (VBP) proposed by M. Ebrahimi. The routing algorithm MAR with partitioning method RP combines recursive partitioning and adaptivity like our proposed routing algorithm. So we also made a brief comparison with it in Section IV. A path diversity aware adaptive routing algorithm was proposed by P. Tsai et al. in [18]. This scheme can predict the latency condition of the output channels by taking into account the number of output ports and buffer state. Many other adaptive routing algorithms including IX/Y [21], APSRA routing [22], RABC routing [23], and ABACUS [25] were also proposed recently. These adaptive routing algorithms

were designed for routing only unicast traffic and some of them are non-minimal adaptive routing algorithms that may cause extra latency for transmitting a packet.

LADPM [1] is a newly proposed multicast routing based on RPM. Similar to RPM, LADPM uses two virtual networks to deal with the deadlock problem, thereby reducing the possibility of misrouting and the latency of RPM. This routing algorithm divides the whole network into at most 8 parts. The packets in the network are classified into two categories based on the relative location of the destination nodes. These two kinds of packets are transmitted in two separated virtual networks and two simple and efficient turn models – North-Last and West-Last turn models [27] are used in these two virtual networks respectively to avoid deadlocks. North-Last and West-Last turn models are no cycle dependency by forbidding two turns respectively to avoid deadlock. The simulation results in [1] show a significant performance improvement than RPM.

III. PROPOSED ROUTING SCHEME

In this section, we present a self-adaptive partition-based algorithm for routing multicast traffic based on LADPM.

A. Flit Format

Fig. 1 shows the flit¹ format of our multicast algorithm. A multicast packet consists a head flit, a sequence of body flits and a tail flit. The head flit contains the set of destinations of the multicast packet. The destination addresses are stored in fields DST_1 to DST_n using the bit-string coding method [26], where n is the number of nodes in the whole network. If the value of field DST_i is set to 1, this means node i is included in destination set. Besides the destination set, FLIT TYPE and VN are two one bit fields that are included in the head flit. FLIT TYPE is to indicate whether this flit is a head flit, a body flit or a tail flit. The VN filed is used to identify the virtual network since the proposed routing algorithm uses two virtual networks to avoid deadlocks. The detail of virtual network and deadlock avoidance will be introduced in the following sections.

¹ A flit is a fixed-length unit of a packet, and it is the smallest unit of flow control.

Fig. 1 Flit Format

B. VN Bit Calculation

For a multicast packet, different routing schemes will lead to different link utilization. For different distributions of multicast destinations, we need to use different routing schemes to achieve minimum link utilization. In our multicast routing algorithm, different multicast packets are routed using two different multicast routing schemes. The packets can be classified into two categories based on the different distribution of the destination nodes. Two types of traffic are transmitted in two separated virtual networks to avoid deadlock. We use one bit field VN in head flit to indicate which virtual network the packet belongs to. And packet transmission from one virtual network to another one is not permitted for deadlock avoidance. Once the packet is injected into the network, the VN field will never be changed by any intermediate router in the whole transmission process. The detail of routing schemes in two virtual networks will be presented in Section III.C.

The value of the VN field in the head flit is called the VN bit. If more horizontal links can be shared, the VN bit is set to 0; while if more vertical links can be shared the VN bit is set to 1. The basic principle of deciding the value of the VN bit is to calculate the link utilization of the packet under XY and YX routing schemes. First of all, we build two multicast trees – XY tree and YX tree. The XY and YX routing methods are applied in these two trees, respectively. Fig. 2 gives an example for using two multicast trees. In the XY tree, all of the paths from source node to destination nodes follow XY routing algorithm. While in YX tree, all the paths follow the YX routing algorithm. After constructing the XY and YX trees, the cost of each multicast tree is calculated. The cost is the number of used links in the multicast tree. If the cost of XY tree is smaller, the VN bit is 0; otherwise, the VN bit is 1.

The VN bit for injecting multicast packets to the proper virtual network is calculated in the source node. And this value is stored in the VN field of head flit. Besides the use for identifying virtual network, this method of calculating VN bit can be used to find alternative routing paths. An intermediate router can also calculate the VN bit but this bit will not be written in the VN field and it's only for finding

alternative paths. If an intermediate router calculates the VN bit of one of the 8 parts (see Section III.C), and the cost of XY tree and YX tree are equal, this means there are alternative paths at this node. And the router can select one path from two or more paths based on the buffer conditions. In Fig. 2, the cost of XY tree is 13 while the cost of YX tree is 16. Thus the VN bit calculated by source IP core in tile 24 is 0. This means more horizontal links can be shared in the transmission of the multicast packet.

Fig. 2 (a) XY tree (b) YX tree

C. Enhanced Adaptive Partition-based Multicast

The proposed routing scheme is a partition-based routing scheme for 2D mesh network. If we define the source node as the origin of the Coordinate System, the first quadrant is part-0, the positive y-axis is part-1 and the rest parts can be divided in a similar manner as shown in Fig. 3. It divides the whole mesh network into at most 8 parts based on the location of the source node. If the source node locates at the center of the network, there are 8 parts. If the source node locates at the border of network, there are 5 parts. There are only 3 parts if the source node locates at the corner of the network. Fig. 3 is an example of 8-part partition when the source is located in the center.

Fig. 3 An example of network partition

Most multicast routing algorithms are not adaptive, and they generate only one multicast tree for each multicast packet, that is, the path that a packet is transmitted from the source to the destinations is fixed. These schemes generate the path without considering the network congestion information and the buffer state. These routing schemes such as RPM [17] and LADPM [1] lack flexibility.

The key idea of improving the performance of multicast routing is to find alternative routing paths without increasing the path length. We use the method that mentioned in Section III.B to identify the chance of adaptive routing. If there are more than one path available, the routing scheme chooses the one with the maximum number of empty channel buffer slots.

North-Last and West-Last based turn models [27] are used in virtual network 0 and virtual network 1, respectively. In the North-Last algorithm, more horizontal links can be shared, whereas in the West-Last algorithm, more vertical links can be shared. The following gives the detailed description of these two routing schemes based on the examples given in Fig. 4 and Fig. 5. The dark gray areas in these two figures indicate parts with destinations while the light gray areas means parts with no destination.

North-Last:

- As illustrated in Fig. 4 (a), Part-0 and Part-2 have destinations. The traffic from the source to nodes in Part-0 is transmitted through the east output port of the router in the source tile, whereas the traffic from the source to nodes in Part-2 is transmitted through the west output port.
- In Fig. 4 (b), Part-2, Part-3, Part-5 and Part-6 have no destination whereas Part-4 has destinations. The router needs to calculate the value of the VN bit for Part-4. If it returns 1, the south output port will be used for transmitting the traffic from the source to the destinations in Part-4. If it returns 0, the west output port will be used for transmitting the traffic. Otherwise, the traffic to the destinations in Part-4 is transmitted through the south or west output port that has a bigger Virtual Channel (VC) buffer.
- In Fig. 4 (c), if Part-2, Part-3, Part-5 or Part-6 are not all empty and Part-4 has destinations, the traffic to Part-4 is transmitted through the south output port if Part-2 and Part-3 have no destination node; otherwise the west output port will be used. If Part-0, Part-4, Part-5 and Part-7 are not all empty and Part-6 has destinations, the traffic to Part-6 is transmitted through the south output port if Part-0 and Part-7 have no destination nodes; otherwise the east output port will be used.
- In Fig. 4 (d), Part-0, Part-4, Part-5 and Part-7 have no destinations and Part-6 has destinations. The router needs to calculate the value for the VN bit for Part-6. If it returns 1, the south output port will

be used for transmitting the traffic from the source to the destinations in Part-6. If it returns 0, the east output port will be used for transmitting the traffic. Otherwise, the traffic to the destinations in Part-6 is transmitted through the south or east output port that has a bigger VC buffer.

Fig. 4 North-Last Partition

West-Last:

- As illustrated in Fig. 5 (a), the traffic to the nodes in Part-2 is transmitted through the north output port, and the traffic to the nodes in Part-4 is transmitted through south output port.
- In Fig. 5 (b), if Part-1, Part-2, Part-6 and Part-7 have no destination and Part-0 has destinations, the router needs to calculate the value of the VN bit for Part-0. If it returns 1, the north output port will be used for transmitting the traffic from the source to the destinations in Part-0. If it returns 0, the east output port will be used for transmitting the traffic. Otherwise, the traffic to the destinations in Part-0 is transmitted through the north or east output port that has a bigger VC buffer.
- In Fig.5 (c), if Part-1, Part-2, Part-6 and Part-7 are not all empty and Part-0 has destinations, the traffic to Part-0 is transmitted through the east output port if Part-1 and Part-2 have no destination

nodes, otherwise the north output port will be used. If Part-0, Part-4, Part-5 and Part-7 are not all empty and Part-6 has destinations, the traffic to Part-6 is transmitted through the south output port if Part-4 and Part-5 have no destination nodes. Otherwise, east output port will be used.

- In Fig. 5. (c), if Part-0, Part-4, Part-5 and Part-7 have no destination and Part-6 has destinations, the router needs to calculate the value of the VN bit for Part-6. If it returns 1, the south output port will be used for transmitting the traffic from the source to the destinations in Part-6. If it returns 0, the east output port will be used for transmitting the traffic. Otherwise, the traffic to the destinations in Part-6 is transmitted through the south or the east output port that has a bigger VC buffer.

Fig. 5 West-Last Partition

The pseudo code of the proposed multicast routing scheme is given below.

/*VN is VN bit in head flit */

/*Current node LOCAL */

/*Multicast Destinations Set D */

/*Destinations Subset in Part-0 to Part-7: $D_0, D_1 \dots D_7$ */

/*Destinations Subset for 4 Directions: D_E, D_W, D_S, D_N*/

/*The credit is the unused buffer of downstream router*/

If (LOCAL is in D)

Send one copy to LOCAL;

Remove LOCAL from D;

If (VN==0)//Use North-Last turn model in virtual network 0

Add D₀|D₇ to D_E;

Add D₁ to D_N;

Add D₂|D₃ to D_W;

Add D₅ to D_S;

If ((D₃|D₅==0)&&(D₄!=0))

If (VN bit of D₄ is 1)

Add D₄ to D_S;

Else if ((VN bit of D₄ is 0)|| (west credit > south credit))//compare the available buffer size of two downstream routers

Add D₄ to D_W;

Else

Add D₄ to D_S;

If ((D₃|D₅!=0)&&(D₄!=0))

If (D₂|D₃==0)

Add D₄ to D_S;

Else

Add D₄ to D_W;

If ((D₅|D₇==0)&&(D₆!=0))

If (VN bit of D₆ is 1)

Add D₆ to D_S;

Else if ((VN bit of D₆ is 0)|| (east credit > south credit))//compare the available buffer size of two downstream routers

Add D₆ to D_E;

Else

Add D₆ to D_S;

If ((D₅|D₇!=0) &&(D₆!=0))

If (D₀|D₇==0)

Add D₆ to D_S;

Else

Add D₆ to D_E;

Else// use West-Last turn model in virtual network 1

Add D₁|D₂ to D_N;

Add D₃ to D_W;

```

Add D4|D5 to DS;

Add D7 to DE;

If (( D1| D7 ==0)&&( D0!=0))

    If ( VN bit of D0 is 1)

        Add D0 to DN;

        Else if(( VN bit of D0 is 0)|| (east credit > north credit)) //compare the available buffer size of two downstream routers

            Add D0 to DE;

        Else

            Add D0 to DN;

If (( D1| D7 !=0)&&( D0 !=0))

    If (D1 | D2 ==0)

        Add D0 to DE;

    Else

        Add D0 to DN;

If ((D5|D7 ==0)&&( D6 !=0))

    If (VN bit of D6 is 1)

        Add D6 to DS;

        Else if( ( VN bit of D6 is 0)|| (east credit > south credit)) //compare the available buffer size of two downstream routers

            Add D6 to DE;

        Else

            Add D6 to DS;

If ((D5|D7 !=0) &&( D6 !=0))

    If ( D4 |D5 ==0)

        Add D6 to DS;

    Else

        Add D6 to DE;

Send one copy to DE, DW, DS, DN;

```

For the destinations in Part-0, Part-4 and Part-6, two different turn models can be applied based on the distribution of the destination nodes in these three parts. If there are more than one available paths for the destination nodes in these 3 parts, the router selects the direction that downstream router has a larger buffer. As a result, the routing results for the same destinations may not be the same. This routing scheme has a larger possibility to avoid making routing decision in congested areas, which can achieve good performance under heavy traffic load.

D. Proposed multicast routing algorithm

Fig. 6 (a) XY tree (b) YX tree

In this section, we introduce the details of the routing procedure. When an IP core wants to send a multicast packet, it firstly calculates the VN bit of the packet using the method introduced in Section III.B and sets the VN bit in the VN field of the head flit. Then the flits of the packet are transmitted to the router that is directly linked to this IP core. The router transmits the flits in the corresponding virtual network based on the VN field value in the head flit using the routing algorithm given in Section III.C. All packets whose VN bit is 0 are injected into Virtual Network 0 (VN0), while the others are injected into Virtual Network 1 (VN1).

We use a simple example to show how the routing algorithm works. In Fig. 7, the source node 24 tries to send a multicast packet to the destination nodes 0, 15, 20, 27, 31 and 41. The source node uses the method mentioned in Section III.B to calculate the VN bit of this packet. The XY and YX trees are constructed in Fig. 6. Then the cost of each tree is calculated from Fig. 6. The cost of XY tree is 14 while the cost of the YX tree is 19. Since the XY tree uses fewer links than the YX tree, more horizontal links can be shared during the transmission and the VN bit of the packet is 0. The value 0 is written in VN field of head flit. The packet is transmitted in virtual network 0. At source node 24, the packet to node 20, 27 and 41 are transmitted through the east output port. The packet to node 31 is transmitted through the north output port. The destination nodes 0 and 15 that are in part-4 are transmitted through the west output port. At intermediate node 23, the XY tree and YX tree from node 23 to destination nodes 0 and 15 are constructed. And the cost of these two trees are equal. In this case, both of the west output port and the south output port can be used for transmitting the packet to node 0 and 15 at intermediate node 23. The intermediate router 23 selects the output port that has a larger VC buffer size. In Fig. 7, the blue arrow represents the path that the router selects. The green arrow means the alternative paths that are not chosen by routers. The similar case occurred in node 8 and node 15. Note that all of the paths are in virtual network 0. Fig. 7 shows all the paths that can be used to

transmit a packet. There are 2 paths from node 23 to node 15 and 3 paths from node 15 to node 0. With this scheme, the router can dynamically make routing decisions based on the distribution of the destination nodes and the downstream router buffer state. Fig. 8 shows the routing result of LADPM. The path from the source node to each destination node is only one and it is fixed. LADPM algorithm lacks self-adaptive feature. When the network is under high traffic load, network congestion may occur. Without adaptive routing algorithms, the packet may suffer from large latency.

Fig. 7 the proposed routing algorithm

Fig. 8 LADPM

E. Router Architecture

Fig. 9 shows the architecture of the multicast router with credit-based flow control that we use in our proposed routing algorithm. The router is similar to the traditional unicast router that is composed of input queues, Routing Computation (RC) module, Virtual Channel (VC) allocator, Switch allocator and Crossbar. But there are some differences between our router and traditional routers. Each of the input channel in our router is implemented with several virtual channels and each virtual channel belongs to one of the two virtual networks (sub-network). Usually, 8 VCs are used in our algorithm (actually more or fewer VCs are also acceptable in our algorithm) and the first four VCs belong to VN0 while the other four VCs belong to VN1. The packets aimed to each direction are assigned to one VC and this reduces the possibility of blocking. Besides, the credit signal of neighbor routers that indicates the buffer state of neighbor routers is transmitted to both switch allocator and routing computation module while in traditional routers the credit information is only transmitted to the switch allocator. This is because, in our algorithm, the RC module makes routing decision based on both of the distribution of the destination nodes and the neighbor router credit information.

Fig. 9 Router Architecture

F. Deadlock-Freedom Analysis

Since the two turn models in our proposed routing algorithms are used in two separated virtual networks, no cycle dependency exists in the network and our multicast routing algorithm is deadlock-free. Once the multicast packet is injected into one of the virtual network, the VN bit in the head flit will never be changed in intermediate routers, and the transmission from one to the other virtual network is strictly forbidden. Fig. 10 shows the turns used in VN0 and VN1. The hollow arrows in Fig. 10 represent the restricted turns. Since the turns selected in intermediate routers are all allowed turns that Fig. 10 shows, the alternative paths chosen by intermediate routers are also deadlock-free.

Fig. 10 (a) Turns in VN0 (b) Turns in VN1

IV. SIMULATION RESULTS

In this section, we evaluate the proposed routing algorithm in comparison with the original LADPM routing scheme in terms of latency and adaptivity. Since the proposed routing algorithm is a recursive partitioning routing algorithm with adaptivity, we also compared our scheme with ARP [24] - Adaptive MAR routing algorithm - using the RP partitioning method with similar features. The simulation methodology is introduced first, and then the evaluation results are presented and discussed in detail.

We use OPNET [28] as our simulator to evaluate the proposed algorithm with the original LADPM and ARP. OPNET (Optimized Network Engineering Tool) is a comprehensive development

environment for simulation and performance analysis of networks. It uses Discrete Event Simulations (DES) as the method to analyze system performance to make it easier to get the latency, throughput and other performance parameters of the network. Table 1 shows the configuration of the simulations. The topology of our network is an 8×8 2D mesh. We compare the latency performance of the multicast routing algorithms with different number of destinations, VCs, and different unicast traffic percentage and injection rate. The multicast traffic follows a random distribution with each node having equal probability to be selected as the destination. In our simulations, we use the following three unicast traffic patterns: uniform, bit-complement and transpose [29].

Table 1. Simulation Configuration

Topology	8×8 mesh
VCs/IC	4, 8
VC Depth	5, 10
Packet Length (flits)	2, 3, 4, 5
Multicast Traffic	Random Distribution
Unicast Traffic	Uniform, Bit-Complement, Transpose
Operation Frequency	0.2GHz

The latency of our scheme and LADPM under various injection rates with different number of destinations is shown in Fig. 11, and Fig. 12. In all these figures, no unicast traffic is injected. The buffer size is set to 10 flits. To show the effect of the number of VCs on latency, we ran simulations under different VC numbers. The number of VCs is 8 in Fig. 11, while in Fig. 12 there are only 4 VCs in each input channel. Under low injection rates, the performance of our algorithm is almost the same as LADPM. The reason is that there is not much congestion in the network. So the alternative paths in our algorithm do not have much effect on the performance. As the injection rate increases, our routing scheme outperforms LADPM.

Fig. 11 shows the latency of the proposed algorithm and LADPM with 4, 6, and 8 multicast destinations using 8 VCs per input channel and 2 flits packet length. In Fig. 11, when the injection rate is under 0.06 packets/cycle/node with 4 multicast destinations, the latency of these two routing schemes is almost the same. When the injection rate increases from 0.06 packets/cycle/node, the performance of LADPM decreases dramatically. If we define the saturation point as the point where the latency is double of the zero latency [30], the saturation point of LADPM is less than 0.07 while our scheme is about 0.09. Our routing algorithm achieves lower latency under heavy workload, since we use adaptive routing to avoid congestion.

When the number of multicast destinations is set to 6, the network saturates earlier for both algorithms in comparison with 4 destinations, and our algorithm still works better under heavy workloads. The saturation point of our algorithm is more than 0.06, and it's higher than LADPM's. Similar results can be observed from the curves for 8 multicast destinations.

Fig. 11 Latency performance of two algorithms with various destination numbers (8 VCs)

Fig. 12 Latency performance of two algorithms with various destination numbers (4 VCs)

Fig. 12 shows the simulation results with fewer VCs. The simulation results are obtained using almost the same configuration except the number of the VCs in each input channel. Each input channel in this simulation is implemented using 4 VCs. With fewer VCs, the saturation point of each algorithm become earlier than that in the simulation with 8 VCs. Even though a slight performance decrease occurred in this simulation, the proposed routing algorithm still offers an obvious latency reduction under high injection rate. The saturation points of our routing algorithm are higher than LADPM with 4, 6 and 8 multicast destinations, respectively. The proposed routing algorithm performs better than LADPM under various number of destination nodes.

Fig. 13 shows the performance of our routing algorithms for various packet lengths when 100% multicast traffic is evaluated. The configuration is 4 VCs per input channel with 4 multicast destinations. With the increase of the packet length, the latency of both algorithms becomes worse. This is because longer packets lead to a larger possibility of congestion. But the latency of our routing algorithm is still better than LADPM under various packet lengths. The saturation point is higher since adaptive routing is applied.

Fig. 13 Latency Performance of two Algorithms with Various Packet Lengths

Fig. 14 Latency Performance of two Algorithms with Unicast Traffic

Fig. 14 shows the latency performance of LADPM and our proposed routing algorithm with 50% unicast traffics. The buffer length in this simulation is set to 5, and the configuration of this simulation is 4 destinations, 50% unicast traffic with different traffic patterns and 3-flit-long packets. As can be seen from Fig. 14, our routing algorithm outperforms LADPM under each kind of mixed traffics. At low injection rate of each traffic pattern, the performance of these two routing algorithms is similar. For Uniform, Bit-Complement and Transpose traffic patterns, when the injection rate is under about 0.06, 0.05 and 0.08, the latency of these two algorithms is nearly the same. When the injection rate increases, the performance of LADPM degrades significantly. It can be seen from Fig. 14, under 50% uniform unicast traffic, the latency of LADPM increases dramatically and the latency of our algorithm is still less than 40 cycles when the injection rate is larger than 0.065. For bit-complement and transpose traffic patterns, similar trends can also be observed.

The latency performance of our routing algorithm and ARP with various number of destinations and unicast traffic are also evaluated, and the performances are shown in Fig. 15 and Fig. 16. The simulation configuration of Fig. 15 is 4 five-flit-deep VCs per channel and 3-flit-long packet with 100% multicast traffic, and the configuration of Fig. 16 is the same with Fig. 14. In these two simulations, the latency performance of our proposed routing algorithm outperforms ARP significantly. ARP has a larger latency under low traffic loads and a smaller saturation point than our algorithm. The difference between ARP and our proposed algorithm is caused by the partitioning method and the routing algorithm. In this 8×8 2D-Mesh topology, ARP partitions the network into 8 parts and the source node needs to transmit at most 8 packets without combining the packets in the same direction into one

multicast packet. More packets are transmitted in the network, which makes network congestion occur more frequently and leads dramatically increase on latency.

Fig. 15 Latency Performance of the Proposed Routing Algorithm and ARP with Various Destination Numbers

Fig. 16 Latency Performance of the Proposed Routing Algorithm and ARP with Unicast Traffic

We use normalized Adaptivity to measure the adaptive routing ability of our algorithm. Adaptivity denotes the times of a given router detects alternative path for multicast packets, and it is normalized as the normalized Adaptivity under the condition of 4 multicast destinations is set to 1. Fig. 17 shows Normalized Adaptivity under various destination quantities and 0.05 packets/cycle/node injection rate with 100% multicast traffic. The horizontal axis is the number of destinations. The result shows that, with the increase of the number of destination nodes, Normalized Adaptivity decreases. This is because the probability of detecting an alternative path decreases with the increase of the number of destination nodes. When the number of destinations is small, the performance improvement of our proposed routing scheme is more significant. If multicast destinations increase, the latency curves of these two algorithms become closer but it is not obvious, as shown in Fig. 11 .

Fig. 17 Normalized Adaptivity under Various Destination Quantities

Fig. 18 Normalized Energy Consumption under Various Traffic Patterns

Fig. 19 Normalized Energy Consumption under Various Destination Numbers

The normalized energy consumption of the proposed routing algorithm and LADPM based on the power model in ORION 2.0 [31] is also evaluated. We use ORION 2.0 to generate the energy consumption of each procedure, such as buffer read or write, link transmissions, etc. Various statistics including the number of buffer read and write, the activities of virtual channels and switch arbiters, the number of crossbar traversals and the transmission at network links are collected in the simulation to obtain energy consumption. The normalized energy consumption of LADPM and our proposed algorithm under different unicast traffic patterns is shown in Fig. 18. The buffer length in this simulation is 5 flits and the unicast portion is 50%. The injection rate is 0.05 and packet length is 3. As can be seen from Fig. 18, our multicast routing algorithm saves 8.94%, 7.46%, and 11.88% of energy respectively than LADPM under these three different unicast traffic patterns. Fig. 19 illustrates the normalized energy consumption of these two algorithms under various number of multicast destinations with 100% multicast traffic. The simulation configuration is the same with Fig. 18, except that the multicast traffic is 100% and the number of the multicast destinations is varied from 4 to 8 in Fig. 19. With the increase of the number of destinations, the energy saved by the proposed routing algorithm increases from 13.53% to 22.37%. The reason is that the average latency of LADPM under these traffic patterns is larger than the proposed one, which leads to more clock power consumption and static power consumption. The performance gap between these two routing algorithms becomes larger with the increase of the number of the destination nodes.

Fig. 20 Link Workloads under Multicast and Uniform Unicast Traffic

Fig. 21 Link Workloads under Multicast and Bit-Complement Unicast Traffic

Fig. 22 Link Workloads under Multicast and Transpose Unicast Traffic

Fig. 20, Fig. 21 and Fig. 22 show the workloads of each link under different mixed traffic of LADPM and our scheme. The simulation configuration is the same with Fig. 18. In this simulation, we recorded the number of the flits that are transmitted through each link. The simulation results show that our algorithm has better load-balancing as there are fewer links under heavy loads. In Fig. 20, the links that transmit more than 13000 flits by the proposed algorithm are fewer than LADPM. The links that

are under low workloads using the proposed algorithm are more than LADPM. Similar trends can be observed from Fig. 21 and Fig. 22. Hence, our algorithm spreads the workload to a larger network area to avoid hot spots and to balance the workload in the whole network. The adaptivity of the proposed algorithm leads to better load-balancing.

As illustrated above, the proposed routing scheme outperforms LADPM routing algorithm in latency, power consumption and load-balance under different packet lengths, number of VCs, unicast traffic portions and multicast destinations conditions. The adaptive routing scheme of the proposed routing algorithm increases the performance significantly.

V. CONCLUSION

In this paper, we proposed a new adaptive multicast routing algorithm for mesh-based NoC. The turns in each virtual network are fully used to implement the adaptive routing algorithm without introducing deadlocks. This algorithm can find the alternative paths for one multicast packet without increasing the path length. The alternative path can divert the traffic from the congestion area and improve the performance of the network. The simulation results show that the latency and power consumption of our proposed routing algorithm are significantly improved than LADPM under different traffic loads.

ACKNOWLEDGMENT

This work is supported by the National Science Foundation of China Grant No.61472300 and No.61334003, and the 111 Project Grant No.B08038.

REFERENCES

- [1] J. Li, L. Shi, C. Xe and Y. Xu, "Dual Partitioning Multicast for High-performance On-Chip Networks," *Journal of Parallel and Distributed Computing*, 74(1), 2014, pp. 1858-1871.
- [2] T. Bjerregaard and S. Mahadevan, "A Survey of Research and Practices of Network-on-Chip," *ACM Computer Survey*, vol. 38, no. 1, pp. 1-51, 2006.
- [3] J. D. Owens, W. J. Dally, R. Ho, D. N. Jayasimha, S. W. Keckler, and L. S. Peh, "Research Challenges for On-Chip Interconnection Networks," *IEEE Micro*, 27(5), 2007, pp. 96-108.
- [4] W. Dally and B. Towles, "Route packets, not wires: on-chip interconnection networks," *the Design Automation Conference*, pp.684-689, 2001.
- [5] L. Benini and G. De Micheli, "Networks on Chips: A New SoC Paradigm," *Computer*, vol. 35, no. 1, pp. 70-78, Jan. 2002.

- [6] M. Mirza-Aghatabar, S. Koohi, S. Hessabi and M. Pedram, "An Empirical Investigation of Mesh and Torus NoC Topologies Under Different Routing Algorithms and Traffic Models", IEEE 10th Euromicro Conference on Digital System Design Architectures, Methods and Tools, 2007, pp. 19-26.
- [7] S. Kumar, A. Jantsch, J. P. Soininen, and M. Forsell, "A Network on Chip Architecture and Design Methodology," IEEE Computer Society Annual Symposium on Very Large Scale Integration (VLSI) Systems, 2002, pp. 105-112.
- [8] J. Henkel, W. Wolf, and S. Chakradhar, "On-Chip Networks: A Scalable, Communication-Centric Embedded System Design Paradigm," IEEE 17th International Conference on Very Large Scale Integration Design, 2004, pp. 845-851.
- [9] D. Wentzlaff, P. Griffin, H. Hoffmann, L. Bao, B. Edwards, C. Ramey, M. Mattina, C. Miao, J. F. Brown, and A. Agarwal, "On-Chip Interconnection Architecture of the Tile Processor," IEEE Micro, 27(5), 2007, pp. 15-31.
- [10] X. Lin and L. M. Ni, "Multicast Communication in Multicomputer Networks," IEEE Transactions on Parallel and Distributed Systems, vol. 4, no. 10, pp. 1105-1117, Oct. 1993.
- [11] H. Xu, P. K. McKinley and L. M. Ni, "Efficient Implementation of Barrier Synchronization in Wormhole-Routed Hypercube Multicomputers," Journal of Parallel and Distributed Computing, vol. 16, pp. 172-184, 1992.
- [12] H. Xu, P. K. McKinley and L. M. Ni, "ComPaSS: Efficient Communication Services for Scalable Architectures," Supercomputing, 1992, pp. 478-487.
- [13] M. Daneshtalab, M. Ebrahimi, S. Mohammadi, and A. Afzali-Kusha, "Low-distance path-based multicast routing algorithm for network-on-chips," Computers & Digital Techniques, IET, 3(5), 2009, pp. 430-442.
- [14] T. Krishna, L. Peh, B. Beckmann, and S. K. Reinhardt, "Towards the Ideal on-Chip Fabric for 1-to-Many and Many-to-1 Communication," 44th Annual IEEE/ACM International Symposium on Microarchitecture, vol. 2, 2011, pp. 71-82.
- [15] Z. Wang, H. Gu, and J. Zhu, "A Partially Adaptive Dual-Partition Multicast Routing for Mesh-based Networks-on-Chip," 4th IEEE International Conference on Information Science and Technology (ICIST2014), pp.444-447, 2014.
- [16] P. Abad, V. Puente, J. Gregorio, "MRR: Enabling Fully Adaptive Multicast Routing for CMP Interconnection Networks," IEEE 15th International Symposium on High Performance Computer Architecture, 2009, pp. 355-366.
- [17] L. Wang, Y. Jin, H. Kim, and E. Kim, "Recursive Partitioning Multicast: A Bandwidth-Efficient Routing for Networks-on-Chip," Proceedings of the 2009 3rd ACM/IEEE International Symposium on Networks-on-Chip, 2009, pp. 64-73.
- [18] P. Tsai, Y. Kuo, E. Chang, H. Hsin, and A. Wu, "Hybrid Path-Diversity-Aware Adaptive Routing with Latency Prediction Model in Network-on-Chip Systems," IEEE International Symposium on VLSI Design, Automation, and Test, pp. 348-351, April 2013.
- [19] M. Ebrahimi, M. Daneshtalab and F. Farahnakian, "HARAQ: Congestion-Aware Learning Model for Highly Adaptive Routing Algorithm in On-Chip Networks", IEEE/ACM International Symposium on Networks-on-Chip, pp. 19-26, 2012.
- [20] M. Ebrahimi, M. Daneshtalab, and J. Plosila, "High Performance Fault-Tolerant Routing Algorithm for NoC-based Many-Core Systems", Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, pp. 462-469, 2013.
- [21] A. M. Shafiee, M. Montazeri and M. Nikdast, "An Innovational Intermittent Algorithm in Network-on-Chip (NOC)," World Academy of Science, Engineering and Technology, pp. 145-147, 2008.

- [22] M. Palesi, R. Holsmark, S. Kumar and V. Catania, "Application Specific Routing Algorithms for Networks on Chip," IEEE Transactions on Parallel and Distributed Systems, vol. 20, pp. 316-330, 2009.
- [23] M. H. Tang and X. L. Lin, "Routing Algorithms by Breaking Cycles," ICA3PP 2010, pp. 163-173, 2010.
- [24] M. Ebrahimi, M. Daneshmand, P. Liljeberg, J. Plosila, J. Flich and H. Tenhunen, "Path-Based Partitioning Methods for 3D Networks-on-Chip with Minimal Adaptive Routing," IEEE Transactions on Computers, vol. 63, no. 3, pp. 718-733, March, 2014.
- [25] B. Z. Fu, Y. H. Han, J. Ma, H. W. Li and X. W. Li, "An Abacus Turn Model for Time/Space-Efficient Reconfigurable Routing," ISCA'11, pp. 259-270, June 2011.
- [26] C. M. Chiang and L. M. Ni, "Multi-address Encoding for Multicast," Parallel Computer Routing and Communication, pp. 146-160, 1994.
- [27] C. J. Glass and L. M. Ni, "The Turn Model for Adaptive Routing," the 19th Annual International Symposium on Computer Architecture, pp. 278-287, 1992.
- [28] OPNET Technologies Inc. (2010). *OPNET Modeler*, Bethesda, MD [Online]. Available: <http://www.opnet.com>
- [29] W. J. Dally, B. P. Towles, "Principles and Practices of Interconnection Networks," Elsevier, 2004.
- [30] N. E. Jerger, L. S. Peh, and M. Lipasti, "Virtual Circuit Tree Multicasting: A Case for on-Chip Hardware Multicast Support," International Symposium on Computer Architecture, pp. 229-240, June 2008.
- [31] A. B. Kahng, B. Li, L. S. Peh and K. Samadi, "ORION 2.0: A Fast and Accurate NoC Power and Area Model for Early-Stage Design Space Exploration," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 20, no. 1, pp. 191-196, January, 2012.

VITAE

Zheng Wang is a postgraduate student in School of Telecommunications Engineering, Xidian University, China. In 2014, He received his B.E. from Xidian University, China. The main research interests of him include: Network-on-Chip, multicast routing, and network-coding-based routing algorithm for Network-on-Chip.

Huaxi Gu received the B.E. degree, M.E. and PhD in Telecommunication Engineering and Telecommunication and Information Systems from Xidian University, Xian in 2000, 2003 and 2005 respectively. He is full professor in the State key lab of ISN, Telecommunication Department, Xidian University, China. His current interests include interconnection networks, networks on chip and optical intrachip communication.

Yintang Yang received the PhD degree from the School of Technical Physics, Xidian University, majoring in the specialty of semiconductor. He is now a professor in Microelectronics Institute, Xidian

University. He won the title of National Model Teacher and the Chinese Youth Science & Technology Award. He has more than 100 publications in refereed journals and conferences.

Haibo Zhang received the MS degree in Computer Science from Shandong Normal University, China in 2005, and the PhD degree in Computer Science from the University of Adelaide, Australia in 2009. Currently he is a senior lecturer at Computer Science department of University of Otago, New Zealand. His research interests include real-time wireless communications, wireless sensor/ad hoc networks, delay-tolerant networks.

Yawen Chen obtained her PhD degree in Computer Science from The University of Adelaide in Australia in 2008. She received Bachelor degree in 2002 and Master degree in 2004 from Shandong Normal University in China. She is now a Lecturer in the University of Otago in New Zealand. Her research interests include resource optimization and performance evaluation in computer networking.