Efficient multicast schemes for 3-D Networks-on-Chip

CrossMark

Xiaohang Wang [a,b,*], Mei Yang [c], Yingtao Jiang [c], Maurizio Palesi [d], Peng Liu [b], Terrence Mak [e], Nader Bagherzadeh [f]

[a] Intelligent Chips and Systems Research Centre, Guangzhou Institute of Advanced Technology, Chinese Academy of Sciences, Guangzhou, Guangdong, PR China
[b] Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, Zhejiang, PR China
[c] Department of Electrical and Computer Engineering, University of Nevada, Las Vegas, USA
[d] University of Enna, Kore, Italy
[e] Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, PR China
[f] Electrical Engineering and Computer Science, University of California, Irvine, USA

ARTICLE INFO

ABSTRACT

3-D Networks-on-Chip (NoCs) have been proposed as a potent solution to address both the interconnection and design complexity problems facing future System-on-Chip (SoC) designs. In this paper, two topology-aware multicast routing algorithms, Multicasting XYZ (MXYZ) and Alternative XYZ (AL + XYZ) algorithms in supporting of 3-D NoC are proposed. In essence, MXYZ is a simple dimension order multicast routing algorithm that targets 3-D NoC systems built upon regular topologies. To support multicast routing in irregular regions, AL + XYZ can be applied, where an alternative output channel is sought to forward/replicate the packets whenever the output channel determined by MXYZ is not available. To evaluate the performance of MXYZ and AL + XYZ, extensive experiments have been conducted by comparing MXYZ and AL + XYZ against a path-based multicast routing algorithm and an irregular region oriented multiple unicast routing algorithm, respectively. The experimental results confirm that the proposed MXYZ and AL + XYZ schemes, respectively, have lower latency and power consumption than the other two routing algorithms, meriting the two proposed algorithms to be more suitable for supporting multicasting in 3-D NoC systems. In addition, the hardware implementation cost of AL + XYZ is shown to be quite modest.

1. Introduction

Networks-on-Chip (NoCs) have emerged as the mainstream on-chip network architecture to efficiently interconnect a large number of (16 or more) processing cores in a many-core system. However, with the continued increase of the number of processing cores on a chip [1] enabled by the rapid advances in CMOS technology, standard NoCs will find that they are difficult to keep up the bandwidths and/or power consumption requirements as imposed by future super large SoCs. To address this pressing interconnect challenge, three dimensional NoCs have been proposed [2]. 3-D integration can considerably help reduce the lengths of global interconnects, resulting in significantly lower interconnection delay and power consumption as well as smaller chip area [3]. Of the existing 3-D integration approaches, including wire bounded, microbump, through-silicon via (TSV) and contactless [3], the TSV technology can offer a high density of vertical interconnects.

With the development of diverse applications and programming models on chip multiprocessors (CMPs), collective communications [4] (one-to-many communications and one-to-all communications) are becoming more common. For example, in CMPs with cache coherent shared memory, the cache coherence protocols involve one-to-many communications to either keep different requests in order or invalidate shared data at different cache nodes [5]. In [6], it has been observed that 5–12% of the network traffic is one-to-many in nature when SPEC, TPC and SPLASH-2 benchmarks are running in PHARMSim, a full system simulator [7]. Also reported in [6], with the employment of the GEMS full system simulator [8], if hardware multicast support is replaced by the use of multiple unicast (so called software multicast [4]), the run time of three applications (i.e., barnes, ocean, and radiosity from the SPLASH-2 benchmark) will be multiplied by as much as 1.4× to 2.2×.

These studies [4,9] clearly indicate that efficient support of one-o-many communications in many-core systems, particularly hardware multicast support, will benefit a wide range of applications

* Corresponding author. Address: Intelligent Chips and Systems Research Center, Guangzhou Institute of Advanced Technology, Chinese Academy of Science, Guangzhou 511458, PR China. Tel.: +86 13588196024.
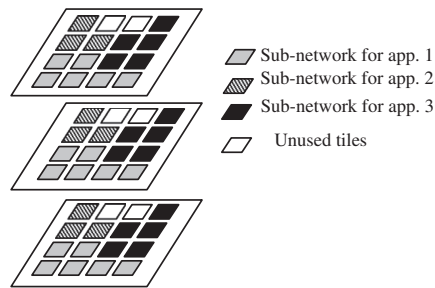E-mail addresses: xh.wang@giat.ac.cn, baikeina@gmail.com (X. Wang).

Fig. 1. Illustration of irregular regions caused by virtualization in a 3-D NoC-based many-core system.

with improved network performance and reduced power consumption. Unfortunately, up to date, very few NoC router designs actually support multicasting [5,6,10], and even fewer for 3-D NoC-based many-core systems [11].

The need of efficient hardware collective communication support is complicated by topological irregularity which might result from virtualization [12]. Virtualization at the NoC level basically allows a single NoC-based CMP to be shared by multiple applications with each mapped to a different region of the chip [13] either statically [14] or dynamically [15,16]. Fig. 1 shows an example where three applications are allocated to three tile regions but with irregular shapes. We call the tiles allocated to one application as sub-network throughout the paper. Applying virtualization to 3-D NoCs faces the challenge of traffic isolation [10], i.e., communication between nodes in a virtualized sub-network shall be limited to the sub-network only. These irregular sub-networks with traffic isolation request negate the application of multicast algorithms designed for regular network topologies. Note that topology irregularity can also result from the presence of faulty links/nodes in an NoC.

To the best of our knowledge, in the literature, there has been no work addressing irregular sub-network oriented multicasting for 3-D NoC systems. In our previous work [17,18], an irregular sub-network-oriented multicasting strategy was proposed for 2-D NoC only. In this paper, we extend the method to cover 3-D NoC with multicasting for both regular and irregular topologies, which first appears in [19].

The organization of the paper is as follows. Section 2 reviews the related work, and Section 3 introduces the NoC architectural and energy models. Section 4 presents the MXYZ algorithm for regular 3-D NoCs, while Section 5 details the strategy to support multicast routing in irregular sub-network and the AL + XYZ algorithm. Section 6 presents the hardware implementation of the proposed algorithms. The experimental results of MXYZ and AL + XYZ are reported in Section 7. Finally, Section 8 concludes the paper.

2. Related work

Multicast communication has been an extensively studied subject in computer networks and interconnection networks [4]. The simplest multicasting scheme is to send a multicast packet as multiple unicast packets. However, such a simple scheme cannot be adopted in NoCs, as it suffers from very large network latency and high power consumption [6]. As a matter of fact, due to the tight power and area constraints pertaining to NoCs, besides low network latency, support of multicasting in NoC imposes additional requirements for low power consumption and small chip footprint.

Taking these power and area constraints into consideration, below we first survey existing multicasting schemes for both 2-D mesh-based and irregular 2-D mesh-based NoCs. As the last part of this section, the only existing multicasting scheme for 3-D NoCs is reviewed.

2.1. Existing multicasting schemes for regular 2-D NoCs

The multicast of regular 2-D mesh topology has been studied as in [4]. Generally, there are two types of multicast routing strategies, the path-based [11,20] and the tree-based [5,6,21–23] multicasting. In path-based multicast routing, multicast packets traverse along the Hamiltonian paths to reach the destinations. Path-based multicast is attractive for its simplicity in hardware design. However, if the destination nodes are widely spread, path-based multicast may suffer from a longer packet latency because packets need to traverse non-minimal paths to form the Hamiltonian paths [6]. It is shown in [6] that path-based multicast may increase network latency by 48% as compared to a tree-based multicast algorithm.

Tree-based multicast routing [5,6,21–23], on the other hand, is to deliver a packet along a common path to the farthest and replicate packets (branch) for a unique set of destination nodes when necessary to achieve a minimal route to reach each destination. Compared with path-based multicast, tree-based multicast routing might have lower packet latency per destination node [6]. Several tree-based multicast approaches have been proposed for regular 2-D mesh-based NoCs. The virtual-circuit-tree-based multicast (VCTM) [6] sets up a multicast path for each multicast communication using a context addressable memory (CAM) at each router. VCTM avoids sending redundant packets as multiple unicast routing strategy does. However, CAM in VCTM might result in high power consumption and area costs. These approaches in [21] and [24] extend the unicast XY routing to support multicasting, where a packet will always be sent to the X direction first and get replicated if there are destinations in the Y direction. The approach in [24] is referred as multicast XY (MXY) in this paper. In [22,23], tree-based adaptive multicast routing approaches are proposed. The region partition multicast (RPM) [5] selects the replication points for multicast packets based on the distribution of destinations in the network partition. Each node partitions the whole network into at most eight regions according to its position. Replication decisions are made by checking the regions where the destination nodes fall into. The simulation results in [5] show that RPM, compared with VCTM, improves the average packet latency by 50% and saves the router and link power by 25%.

2.2. Existing multicasting schemes for irregular 2-D NoCs

Those multicast/unicast routing algorithms for regular topologies, unfortunately, are not suitable for irregular topologies, because certain part of a routing path found by these algorithms may not belong to the irregular sub-network (i.e., dissatisfying the traffic isolation requirement). In the literature, routing tables are typically used to route packets in irregular topologies [25]. However, if the traffic pattern is not predictable, as for the traffic generated by cache coherence protocols [9], building routing tables may incur additional delay.

The bLBDR routing [10] proposed for collective communication in irregular sub-networks support multicasting by broadcasting in sub-networks. In bLBDR, connectivity bits are used to define different sub-networks. However, the broadcast nature of this scheme makes the network tend to be easily congested, which apparently results in higher power consumption.

In our previous work [17,18], an irregular sub-network oriented multicast strategy was proposed for 2-D NoCs. A regular mesh oriented multicast routing algorithm is used as the basic routing algorithm. When this basic routing algorithm fails to find an output channel because of the irregularity, the multicast packet will be forwarded or replicated to an alternative output channel.

2.3. Existing multicasting approaches for 3-D regular NoCs

Up to date, few work with exception in [11,26] touches on multicasting support for 3-D regular NoCs. In a broader sense, it is a path-based multicast routing algorithm. The network is partitioned according to the distribution of the destination set while trying to balance out the numbers of destinations in each partition. For each partition, a multicast packet is generated which follows a Hamiltonian path. Several partitioning methods have been considered to balance the destination number in each of the partitions. For example, the hybrid partition (HP) method further divides the partitions that include a large number of destinations into several partitions in order to ensure that each of the partitions has roughly the same destination number as any of others.

3. 3-D NoC models

In this section, a generic 3-D NoC architecture and its energy model for its communications are described.

3.1. 3-D NoC architecture

The target 3-D NoC architecture is a vertically stacked mesh-based system with a total of $N \times N \times N_z$ tiles. That is, the 3-D NoC system has N_z layers and each layer has $N \times N$ tiles. Each tile can be indexed by the coordinate (x, y, z) with $1 \leqslant x$, $y \leqslant N$ and $0 \leqslant z \leqslant N_z - 1$. As shown in Fig. 2, each tile is composed of a router and IP core(s) [2]. The IP core(s) could be a processor core, or a memory unit (e.g., L1 cache or L2 unit/bank), or a combination of
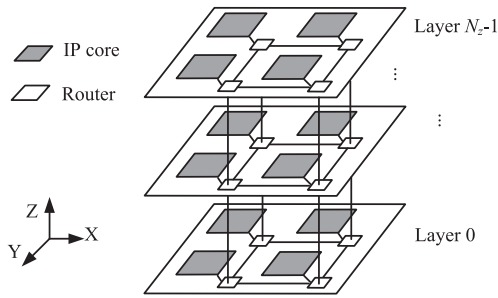

Fig. 2. 3-D NoC archietcture.

them (for example, a tile can be composed of a processor, L1 data and instruction cache, and an L2 bank). Each router (shown in Fig. 3) has seven ports, i.e., East, West, North, South, Local, Up, and Down. A 7×7 crossbar switch is used as the switching fabric of the router. The arbitration unit arbitrates the connection requests sent from the input ports so that each output port receives data from at most one input port. At each input port, multiple virtual channels (VCs) are used. The VC allocation unit controls the virtual channel allocation. The unicast routing unit implements the unicast routing algorithm and the multicast routing unit implements the multicast routing algorithm, respectively.

Assume wormhole switching is used here. To support multicast, a multicast packet could be replicated to multiple output channels according to the decision of the multicast routing unit. The replication is done inside the crossbar where a packet is forwarded to multiple output channels. A packet is broken into flits which might occupy multiple buffers of the routers along the multicast tree. Asynchronous replication [27] scheme is used, where multiple replicated flits are allowed to be forwarded independently. If one replicated flit is blocked on its destined output channel, the replicated flits to other output channels can still be forwarded asynchronously.

3.2. Energy model

3.2.1. Dynamic energy/power

The average dynamic energy consumption for a unicast communication which sends one bit from a source tile s to a destination tile t can be represented as,

$$E_{Unicast}^{s,t} = \eta E_{Rbit} + \eta_H E_{LHbit} + \eta_V E_{LVbit} \tag{1}$$

where η is the number of routers traversed from tile s to tile t, E_{Rbit} is the bit energy consumed by the router, E_{LHbit} is the bit energy consumed on each horizontal link, and E_{LVbit} is the bit energy consumed on each vertical link. η_H and η_V represent the numbers of horizontal and vertical links on the communication path, respectively. Following the wire model in [28], $E_{LHbit} = d_H V_{dd}^2 C_{wireH}/2$ and $E_{LVbit} = d_V V_{dd}^2 C_{wireV}/2$, where d_H and d_V are the lengths of the respective horizontal and vertical links, V_{dd} is the supply voltage, and C_{wireH} and C_{wireV} are the respective wire capacitances of horizontal and vertical links.

The average energy consumption for a multicast communication which sends one bit from a source tile s to a set of destination tiles $\overline{D}$ can be represented as,
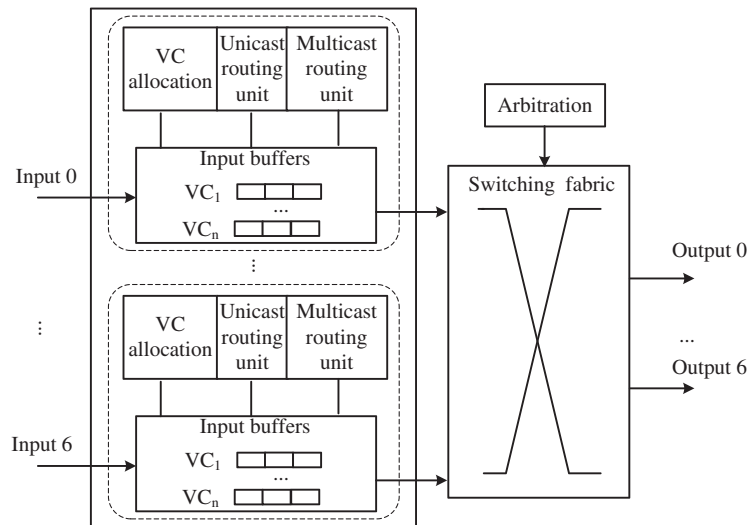

Fig. 3. Router micro-architecture for 3-D NoCs.

$$E_{Multicast}^{s,\overline{D}} = \eta E_{Rbit} + \eta_{MH} E_{LHbit} + \eta_{MV} E_{LVbit} \qquad (2)$$

where η is the total number of routers, and η_{MV} and η_{MH} represent the total numbers of horizontal and vertical links traversed on the multicast path from tile s to all tiles in $\overline{D}$, respectively.

3.2.2. Leakage power

Leakage power of any component is due to the sub-threshold current I_{sub0} and the gate leakage current I_{g0}. The leakage power of a component the j-th component can be represented as

$$P_j^l(\tau) = I_{sub0j} + I_{g0j} = A_j \alpha e^{\beta(T(\tau)-T_{ref})} = R e^{\beta(T(\tau)-T_{ref})}$$

where j is the index of a tile, A_j is the area of the tile j, $T(\tau)$ is the temperature of the tile, T_{ref} is the reference temperature (for example, 383 K), R is the leakage power at the reference temperature, and α and β are the scaling factors among CMOS technology nodes [29].

4. Multicast support for regular 3-D NoCs

In this section, the assumptions/definitions and data structures are first introduced, and they are applicable to both regular and irregular 3-D NoC multicasting algorithms to be presented in this paper. In the second part of this section, we shall present a multicasting algorithm for regular 3-D NoCs.

4.1. Assumptions and definitions

Assumption 1. The applications can be statically or dynamically mapped to the NoC-based many-core system. For dynamic mapping [15], a global manager processor (GM) is responsible for the resource management.

Assumption 2. In each packet, the destination addresses are encoded to form a bit string [4].

Regions are defined to differentiate the destination locations.

Definition 1. *Regions.* At each tile with coordinate (x, y, z), the network is partitioned into 10 disjoint regions, $R_0, R_1, \ldots, R_9$, such that,

$$\text{Tiles with coordinate}(\mathbf{x_0}, \mathbf{y_0}, \mathbf{z_0}) = \begin{cases} \in R_0, & \text{if } x_0 > x \text{ and } y_0 < y \\ \in R_1, & \text{if } x_0 > x \text{ and } y_0 < y \\ \in R_2, & \text{if } x_0 > x \text{ and } y_0 < y \\ \in R_3, & \text{if } x_0 > x \text{ and } y_0 = y \\ \in R_4, & \text{if } x_0 > x \text{ and } y_0 > y \\ \in R_5, & \text{if } x_0 > x \text{ and } y_0 > y \\ \in R_6, & \text{if } x_0 > x \text{ and } y_0 > y \\ \in R_7, & \text{if } x_0 > x \text{ and } y_0 = y \\ \in R_8, & \text{if } x_0 = x \text{ and } y_0 = y \text{ and } z_0 > z \\ \in R_9, & \text{if } x_0 = x \text{ and } y_0 = y \text{ and } z_0 < z \end{cases} \qquad (3)$$

Fig. 4 shows a partition example where a network of size of $4 \times 4 \times 3$ nodes is partitioned into 10 disjoint regions with respect to tile 25. A nodes index, say i, is uniquely determined from its coordinate (x, y, z) as, $i = x + y \times N + z \times N \times N$. The partition of the network into regions helps identify the destination locations, which then helps determine how to replicate packets.

Two groups of bit vectors are defined at each router in the proposed multicast routing algorithms.

(1) The destination set inclusion bit vectors. For a multicast packet, the destination addresses are checked to determine which regions that the destination nodes belong to. At the multicast routing unit of each input port, the following three types of bit vectors are used to realize the destination set inclusion function in hardware.
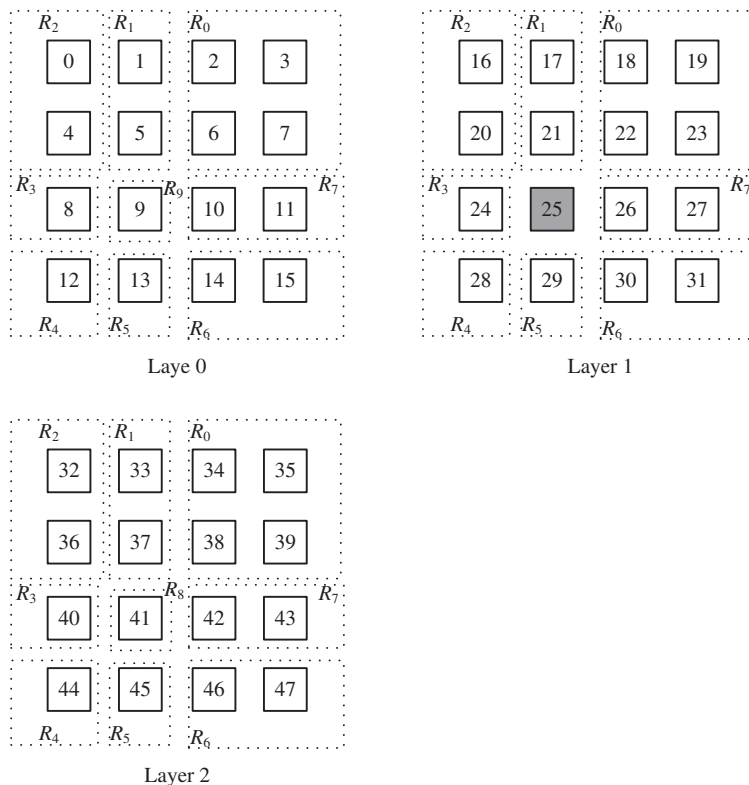

Laye 0

Layer 1

Layer 2

Fig. 4. The disjoint regions of tile 25 for a 3-D NoC with $4 \times 4 \times 3$ nodes.

	47	...	39	38	37	36	35	34	...	23	22	21	20	19	18	...	7	6	5	4	3	2	1	0
Packet head Destinations bit string (D)	0	...	0	0	0	0	0	0	...	0	0	0	1	0	1	...	0	0	0	0	0	1	0	1
Bit mask for R_0 (BM_R_0)	0	...	1	1	0	0	1	1	...	1	1	0	0	1	1	...	1	1	0	0	1	1	0	0
Destinations in R_0 (IN_R_0)	0	...	0	0	0	0	0	0	...	0	0	0	0	0	1	...	0	0	0	0	0	1	0	0
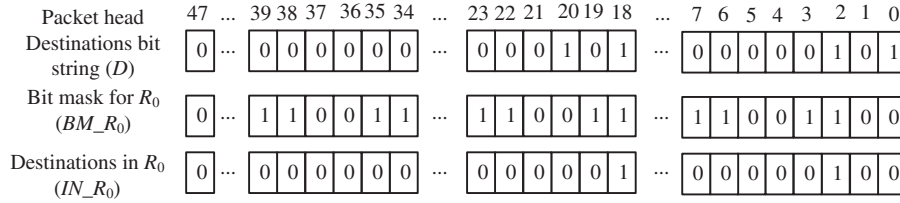
Fig. 5. Bit vectors at node 25 assuming the destinations are 0, 2, 18 and 20. The network is shown in Fig. 4.

- Input destination bit vector D. To encode all nodes in the network, an $(N \times N \times N_z)$-bit vector will be used. The i-th bit of the vector is 1 if the i-th node is inside the destination set.
- Bit mask vector for each region. There are ten bit masks, BM_R_0, ..., BM_R_9, each with $N \times N \times N_z$ bits. The m-th bit of BM_R_i is 1 if node m is in region R_i of the current node. BM_R_i's ($i = 0..9$) are set offline.
- Destination bit vector for each region. Ten bit vectors IN_R_0, ... IN_R_9, each with $N \times N \times N_z$ bits, are used to represent the destinations within each region. For example, IN_R_i ($i = 0..9$) is obtained by performing logical AND operation of D and the corresponding BM_R_i.

Consider the network in Fig. 4 and a multicast example with the source node 25 and four destinations 0, 2, 18 and 20. Fig. 5 shows the three bit vectors at node 25.

(1) The output destination bit vectors. Six bit vectors $N\_DestSet$, $S\_DestSet$, $E\_DestSet$, $W\_DestSet$, $Up\_DestSet$, and $Down\_DestSet$, each of $N \times N \times N_z$ bits, are used to represent the destinations on each output direction.

4.2. Multicast XYZ (MXYZ) algorithm

With the definitions above, a tree-based, dimension-ordered multicast routing algorithm MXYZ is proposed to support multicasting in regular 3-D NoC systems. The packet is delivered along the X direction to the farthest. Then the packet is replicated or forwarded along the Y direction as necessary to reach a unique set of destination nodes. The packet can be further replicated or forwarded along the Z direction as necessary until the packet is delivered to every destination node.

Fig. 6 lists the MXYZ pseudo code assuming the destination bit vectors IN_R_i's are obtained. In MXYZ, if there are destinations in R_0, R_2, R_3, R_4, R_6 or R_7, packets are forwarded or replicated in the X direction. Therefore, bits in $E\_DestSet$ corresponding to destinations in $R_0/R_6/R_7$ or $W\_DestSet$ corresponding to destinations in $R_2/R_3/R_4$ are set accordingly. If there are destinations in R_1 or R_5, packets are forwarded or replicated in the Y direction. Therefore, bits in $N\_DestSet$ corresponding to destinations in R_1 or $S\_DestSet$ corresponding to destinations in R_5 are set accordingly. If there are destinations in R_8 or R_9, packets are forwarded or replicated in the Z direction, i.e., bits in $Up\_DestSet$ corresponding to destinations in R_8 or bits in $Down\_DestSet$ corresponding to destinations in R_9 are set accordingly.

The replication rules are included in this pseudo code algorithm in Fig. 6. For example, suppose there are destinations in both regions R_1 and R_5. Consider the following two rules from Fig. 6,

- $DestSet = IN\_R1$ OR $N\_DestSet$, and
- $S\_DestSet = IN\_R5$ OR $S\_DestSet$.

In this way, there would be two packets generated. The first packet includes the destinations in R_1 and the second includes the destinations in R_5.

MXYZ(IN_R_0, ..., IN_R_9)
Input: IN_R_i ($0 \leq i \leq 9$): Destination bit vectors for each region.
Output: $N\_DestSet$, $E\_DestSet$, $W\_DestSet$, $S\_DestSet$, $Up\_DestSet$, $Down\_DestSet$: Output destination bit vectors corresponding to each output direction.
Function: Find the output channels according to the distribution of the destination set.
Procedure body:
begin
    // Bit vector IN_R_i ($0 \leq i \leq 9$) is 1 if R_i includes any multicast destination
    // Bit vectors $N\_DestSet$, $E\_DestSet$, $W\_DestSet$, $S\_DestSet$, $Up\_DestSet$,
    // and $Down\_DestSet$ are initialized to 0

    // if vectors IN_R_0 or IN_R_6 or IN_R_7 are non-zero
    if (IN_R_0 OR IN_R_6 OR IN_R_7)
        // For destinations in R_0, R_6 and R_7, enclose them to East direction
        $E\_DestSet = IN\_R_0$ OR IN_R_6 OR IN_R_7 OR $E\_DestSet$

    // if vector IN_R_2 or IN_R_3 or IN_R_4 are non-zero
    if (IN_R_2 OR IN_R_3 OR IN_R_4)
        // For destinations in R_2, R_3 and R_4, enclose them to West direction
        $W\_DestSet = IN\_R_2$ OR IN_R_3 OR IN_R_4 OR $W\_DestSet$

    // if vector IN_R_1 is non-zero
    if (IN_R_1)
        // For destinations in R_1, enclose them to North direction
        $N\_DestSet = IN\_R_1$ OR $N\_DestSet$

    // if vector IN_R_5 is non-zero
    if (IN_R_5)
        // For destinations in R_1, enclose them to South direction
        $S\_DestSet = IN\_R_5$ OR $S\_DestSet$

    // if vector IN_R_8 is non-zero
    if (IN_R_8)
        // For destinations in R_8, enclose them to Up direction
        $Up\_DestSet = IN\_R_8$ OR $Up\_DestSet$

    // if vector IN_R_9 is non-zero
    if (IN_R_9)
        // For destinations in R_9, enclose them to Down direction
        $Down\_DestSet = IN\_R_9$ OR $Down\_DestSet$
  end

Fig. 6. Pseudo code of MXYZ.

Fig. 7(a) illustrates how MXYZ works on a $4 \times 4 \times 3$ network. The source tile 6 sends multicast packets to five destinations: tiles 0, 4, 12, 28, 44.

- At the source tile 6, two packets are generated, i.e., one to the West output channel (packet A) and the other to the East output channel (packet B).
- Packet A follows the XY routing path to reach the destination tile 0.
- Packet B reaches the destination tile 4 and then two packets are replicated, one (packet C) to the Up output channel targeting to the destination tile 28, and one to the North output channel targeting to the destination tile 12.

Finally, at the destination tile 12, a packet is replicated to the Up output channel targeting to the destination tile 44 (packet D).

As a tree-based approach, MXYZ is compared against the path-based hybrid partition (HP) multicast routing algorithm [11]. In order to present a consistent numbering for both MXYZ
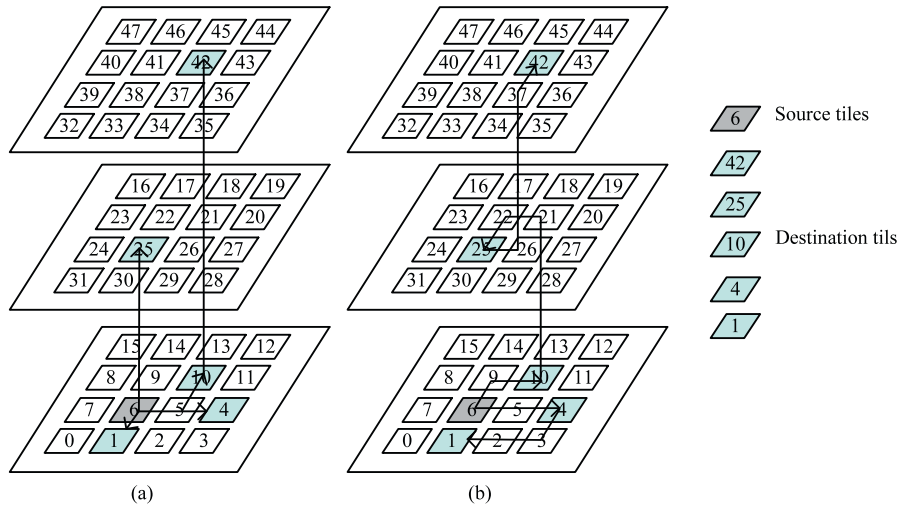
Fig. 7. Comparison of HP and MXYZ in terms of hop counts by an example. (a) The communication paths generated by MXYZ routing. (b) The communication paths generated by HP routing.
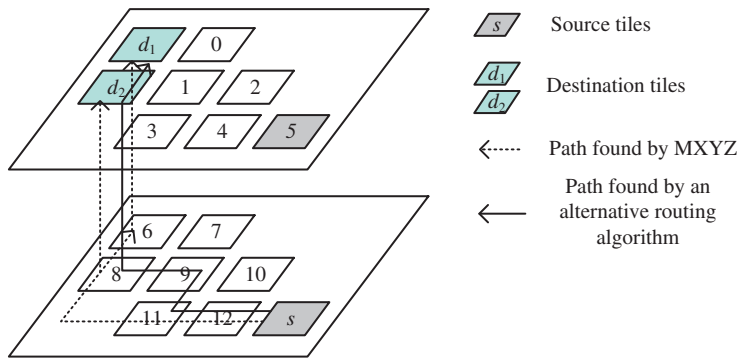

Fig. 8. An example illustrating the irregular sub-network oriented multicasting strategy.

and HP, the Hamiltonian path numbering is used for both multicast routing algorithms, although in the latter of the paper the conventional numbering is used. Fig. 7 (b) shows that for the same source and destination set, there are totally 13 hops on the communication paths of HP. In contrast, in Fig. 7(a), there are only 7 hops on the communication paths of MXYZ. The larger hop count of HP is due to the fact that the Hamiltonian paths formed in HP might lead to non-minimal communication paths. Consequently, the packets will traverse on paths with more hops, resulting in higher energy consumption and higher packet latency. For example, the packets need to traverse along tiles 6, 5, 4, 3, 2 to reach the destination tile 1 (Fig. 7(b)). Instead, in Fig. 7(a), a packet could be replicated to the South output at tile 6 directly to reach the destination tile 1. By this way, the hop count of the paths can be reduced. Similarly, to reach the destination tile 42, the packet needs to traverse more hops in HP than that in MXYZ.

5. Irregular sub-network oriented multicasting in 3-D NoCs

In this section, the irregular sub-network oriented multicasting strategy and such a multicast routing algorithm AL + XYZ are proposed.

5.1. Irregular sub-network oriented multicasting strategy

Consider the irregular sub-network shown in Fig. 8. If the multicast routing algorithm is based on MXYZ, the dashed arrow indicates the routing path from the source tile *s* to the destination tiles
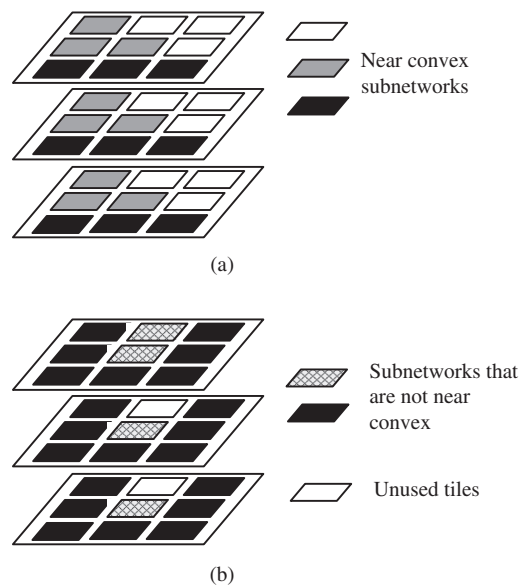

Fig. 9. (a) Near convex sub-networks. (b) Sub-networks that are not near convex.

d_1 and d_2. Apparently, the packet following the dashed path cannot reach the destination because the West output port of tile 11 does not connect to any tile in the sub-network. However, if the path turns to North at tile 11, the packet could reach both destinations (shown as the solid arrow).
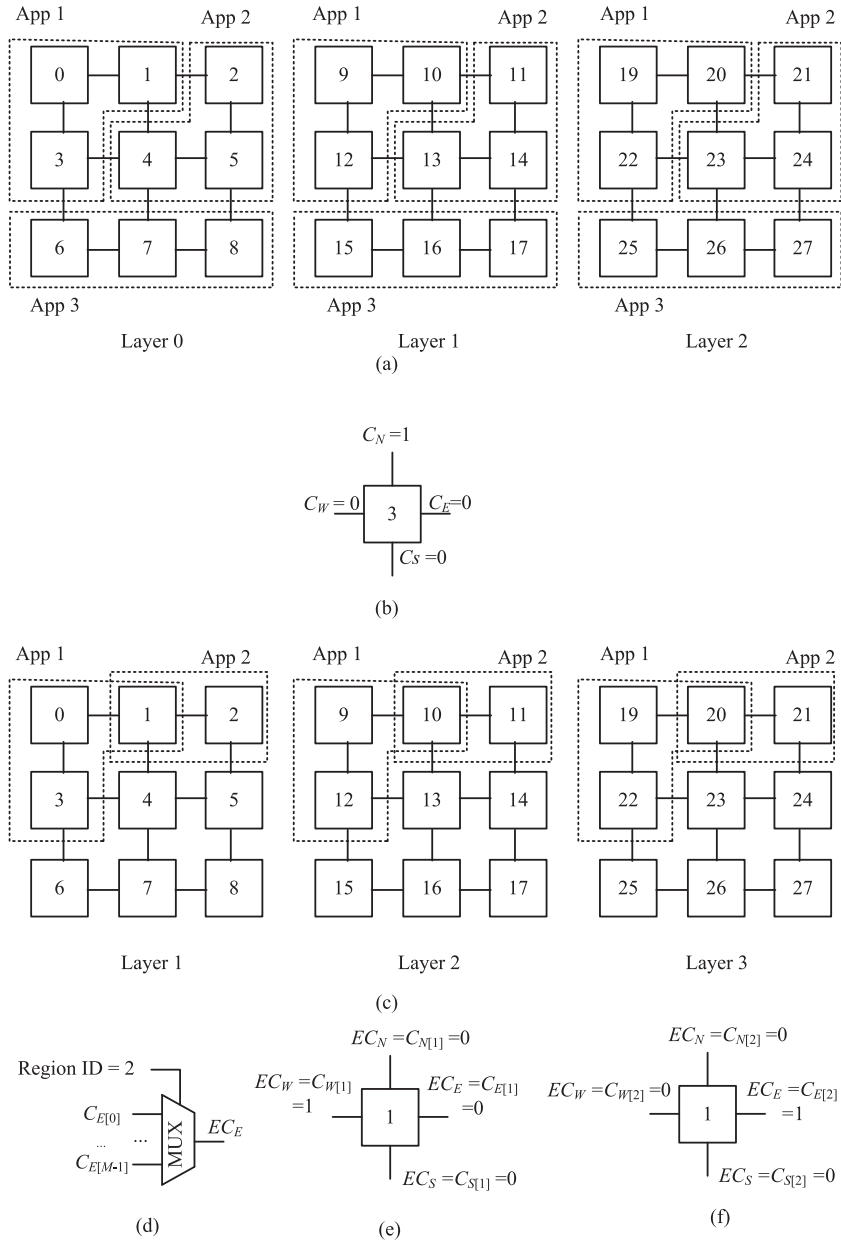
Fig. 10. (a) Sub-networks with 3 applications mapped on. (b) Connectivity bits of node 3. (c) Two overlapped sub-networks sharing node 2. (d) Extended connectivity bit generated using a MUX. (e) Extended connectivity bits of node 1 for Sub-network 1 in (c). (f) Extended connectivity bits of node 1 for Sub-network 2 in (c).

From this example, a simple but yet effective irregular sub-network oriented multicasting strategy is derived as follows.

- Find the output directions to all the destinations in the destination set using a multicast routing algorithm designed for regular 3-D mesh topology.
- For each output direction, check its corresponding connectivity bit. If it is set, then the packet will be replicated and sent to the output direction; otherwise, use an alternative output direction. That is, if a tile X direction output channel is not available, packets can now be forwarded to its Y direction output channel.

5.2. Assumptions and definitions

Besides the assumptions and definitions introduced in Section 3.1, additional assumptions and two definitions are made particularly for irregular sub-networks.

Assumption 3. The shapes of the sub-networks in consideration should be near convex. That is, a sub-network is composed by N_z layers of planar sub-regions and all planar sub-regions are identical. Each planar sub-region should be convex, i.e., the minimal path of any two tiles should be inside the same sub-region (Fig. 9). Sub-networks with such shapes could be generated by the online incremental mapping algorithm proposed in our work [15].

Fig. 9(a) illustrates the sub-network shapes that are conformed to Assumption 3. Fig. 9(b) shows an example of the sub-networks not conformed to Assumption 3. In Fig. 9(b), the black tiles form a sub-network with planar sub-regions of concave shape. In Fig. 9(b), the crossed tiles form a sub-network where the sub-regions have different shapes.

To support multicasting inside the irregular sub-networks, a set of connectivity bits are added.

AL+XYZ (IN_R_0, ..., IN_R_9)
Input: IN_R_i ($0 \leq i \leq 9$): Destination bit vectors for each region.
Output: $N\_DestSet$, $E\_DestSet$, $W\_DestSet$, $S\_DestSet$, $Up\_DestSet$,
        $Down\_DestSet$: Output destination bit vectors corresponding to each
        output direction.
Function: Find the output channels according to the distribution of
        destination set.
Procedure body:
begin
    // Bit vectors $N\_DestSet$, $E\_DestSet$, $W\_DestSet$, $S\_DestSet$, $Up\_DestSet$,
    // and $Down\_DestSet$ are initialized to 0
    // Bit vectors IN_R_i is 1 if R_i includes any multicast destination ($0 \leq i \leq 9$)

    if (IN_R_0) **begin** // vector IN_R_0 is non-zero
      if ($EC_E == 0$) $N\_DestSet = IN\_R_0$ OR $N\_DestSet$
      else $E\_DestSet = IN\_R_0$ OR $E\_DestSet$
    end

    if (IN_R_1) $N\_DestSet = IN\_R_1$ OR $N\_DestSet$

    if (IN_R_2) **begin** // vector IN_R_2 is non-zero
      if ($EC_W == 1$) $W\_DestSet = IN\_R_2$ OR $W\_DestSet$
      else $N\_DestSet = IN\_R_2$ OR $N\_DestSet$
    end

    if (IN_R_3) $W\_DestSet = IN\_R_3$ OR $W\_DestSet$

    if (IN_R_4) **begin**// vector IN_R_4 is non-zero
      if ($EC_W == 1$) $W\_DestSet = IN\_R_4$ OR $W\_DestSet$
      else $S\_DestSet = IN\_R_4$ OR $S\_DestSet$
    end

    if (IN_R_5) $S\_DestSet = IN\_R_5$ OR $S\_DestSet$

    if (IN_R_6) **begin**// vector IN_R_6 is non-zero
      if ($EC_E == 1$) $E\_DestSet = IN\_R_6$ OR $E\_DestSet$
      else $S\_DestSet = IN\_R_6$ OR $S\_DestSet$
    end

    if (IN_R_7) $E\_DestSet = IN\_R_7$ OR $E\_DestSet$
    if (IN_R_8) $Up\_DestSet = IN\_R_8$ OR $Up\_DestSet$
    if (IN_R_9) $Down\_DestSet = IN\_R_9$ OR $Down\_DestSet$
end

Fig. 11. The pseudo code of AL + XYZ.

Definition 2. *Connectivity bits.* Each router has 4 connectivity bits, C_N, C_E, C_S, and C_W, each defining the connectivity at the specific output direction. Suppose a tile has coordinate (x, y, z), C_N is 1 if the tile and its north neighbor tile $(x, y - 1, z)$ are in the same sub-network. Similarly, C_d is 1 if the tile and its neighbor tile on the d direction are in the same sub-network.

A tile may be shared by several sub-networks. For example, a cache memory may be shared by several applications. Fig. 10(c) shows that tile 1 is shared by two overlapped sub-networks. The connectivity bits in Definition 2 cannot describe the overlapped sub-networks (e.g., two applications share the same memory node). As in [10], to support up to M overlapped sub-networks, each connectivity bit is extended to M bits indexed by the sub-network ID (Fig. 9(c)). For example, in Fig. 9(d), C_E is extended into $C_{E[0]}$, ..., $C_{E[3]}$ if 4 sub-networks need be supported.

Definition 3. *Extended connectivity bits.* To support up to M overlapped sub-networks, each router located at the tile with coordinate (x, y, z) has $4 \times M$ connectivity bits [17], $\{C_{N[0]}, ..., C_{N[M-1]}\}$, $\{C_{W[0]}, ..., C_{W[M-1]}\}$, $\{C_{E[0]}, ..., C_{E[M-1]}\}$, and $\{C_{S[1]}, ..., C_{S[M-1]}\}$. $C_{d[q]} = 1$ (d represents N, E, S, or W, $q = 0, ..., M-1$) if the tile and its neighbor tile on the d direction are in the same sub-

network with ID q. Extended connectivity (EC) bits EC_N, EC_E, EC_S, and EC_W are defined as follows. Given the sub-network ID q, $EC_N = C_{N[q]}$, $EC_W = C_{W[q]}$, $EC_E = C_{E[q]}$, $EC_S = C_{S[q]}$.

Fig. 10(d) shows that a MUX can be used to find the value of an extended connectivity bit from the connectivity bits given a sub-network ID. Fig. 10(e) and (f) show the values of extended connectivity bits of tile 2 for sub-network with ID 1 and 2, respectively. The connectivity bit registers can be set statically or by GM [30] with dynamic mapping.

5.3. Irregular sub-network oriented multicasting routing algorithm

Based on the proposed irregular sub-network oriented multicasting strategy (Section 5.1), an irregular sub-network oriented multicasting routing algorithm named as Alternative XYZ is proposed for 3-D NoCs based on MXYZ. It is important to point out that, based on the proposed strategy, any efficient multicast routing algorithm supporting regular 3-D NoC topologies can be used as the basic multicast routing algorithm (not limited to MXYZ) to develop the irregular sub-network oriented multicast routing algorithm. The pseudo code of the AL + XYZ algorithm running in the MR module is shown in Fig. 11. Assume the destination
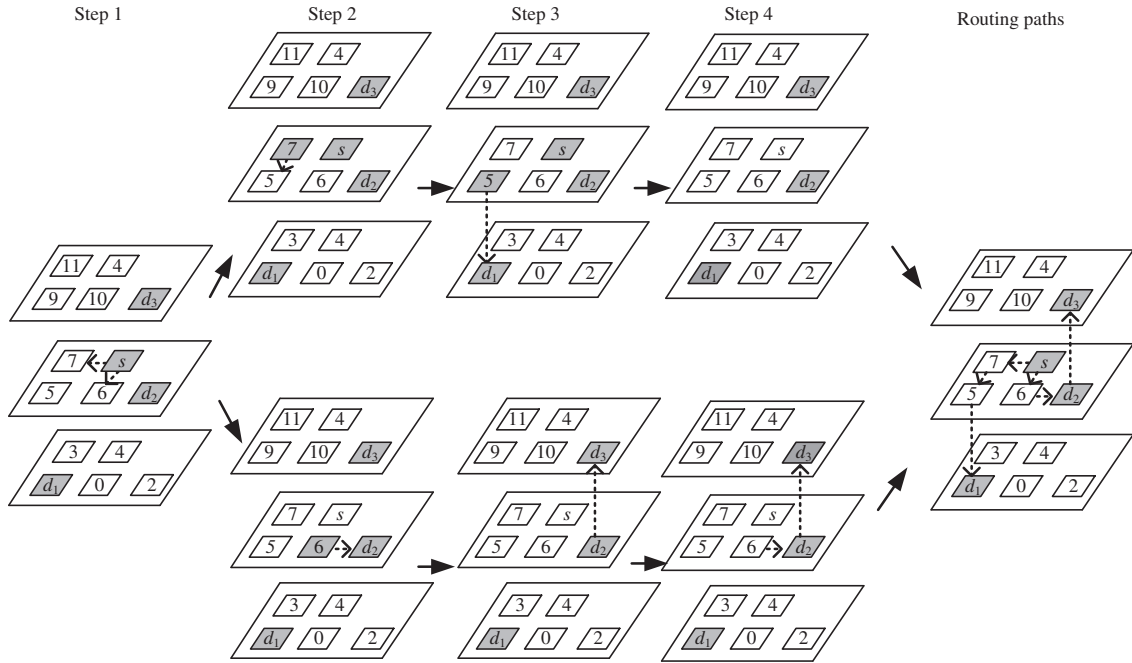
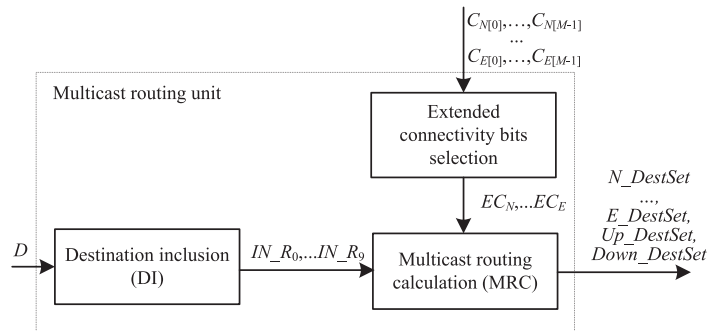**Fig. 12.** Examples showing the routing steps of AL + XYZ.

Fig. 13. Structure of the multicast routing unit.

bit vectors IN_R_i's are obtained using the method described in Section 4.2.

Using MXYZ as the basic routing algorithm, the alternative output direction is on the Y direction. Thus, if there are destinations in regions R_0 and R_2, North ($Y+$) is selected as alternative output direction. If there are destinations in regions R_4 and R_6, South ($Y-$) is selected as alternative output direction. Note that, if destinations are in regions R_1, R_3, R_5, R_7, R_8, R_9, i.e., the $X+$, $X-$, $Y+$, $Y-$, $Z-$, $Z+$ directions, there is no alternative output direction. The reason is that, according to Assumption 3, the sub-network must be near convex which ensures that between any pair of tiles there exists at least one minimal path inside the sub-network. It is clear that for each destination in regions R_1, R_3, R_5, R_7, R_8, R_9, there is only one minimal path to these destinations. Hence, there is no alternative output direction for destinations in those regions. Thus, only the alterative output directions for destinations in regions R_0, R_2, R_4, R_6 are found, which are N, N, S, and S, respectively.

Fig. 12 shows an example of finding the routing path using AL + XYZ for a multicast communication from s to destinations d_1, d_2, and d_3 on an irregular sub-network. In step 1, instead of replicating two packets to West and East (based on output directions generated by MXYZ), node s replicates two packets to nodes 7 (West) and 6 (South). Because node s has extended connectivity bits EC_W equal to 1 and EC_E equal to 0. The alternative output of East is South. In step 2, node 7 forwards the packet to node 5. Node 6 forwards the packets to reach one of the destinations nodes d_2. In step 3, node 5 forwards the packet to one of the destination nodes d_1. Node d_2 forwards the packet to the last destination node d_3.

To avoid deadlocks, the physical network is separated into two virtual networks [5], VN_0 and VN_1. First, each of the virtual networks does not include cyclic path by forbidding some turns. VN_0 does not allow packets to turn to the North direction while VN_1 does not allow packets to turn to the South direction. Second, packets are only allowed to be traversed on one virtual network only. As such, it is ensured that no cycle will be formed in the whole network.

To realize this scheme, packets must be distinguished by the virtual network that they should follow. One bit is thus added in the packet header to identify the virtual network for a packet. At the source router, this bit is initialized for the packets. If the destinations are in the North direction of the source node, the packet follows VN_0, while the packet follows VN_1 if the destinations are in the South direction. If the destinations lie in both sides, the
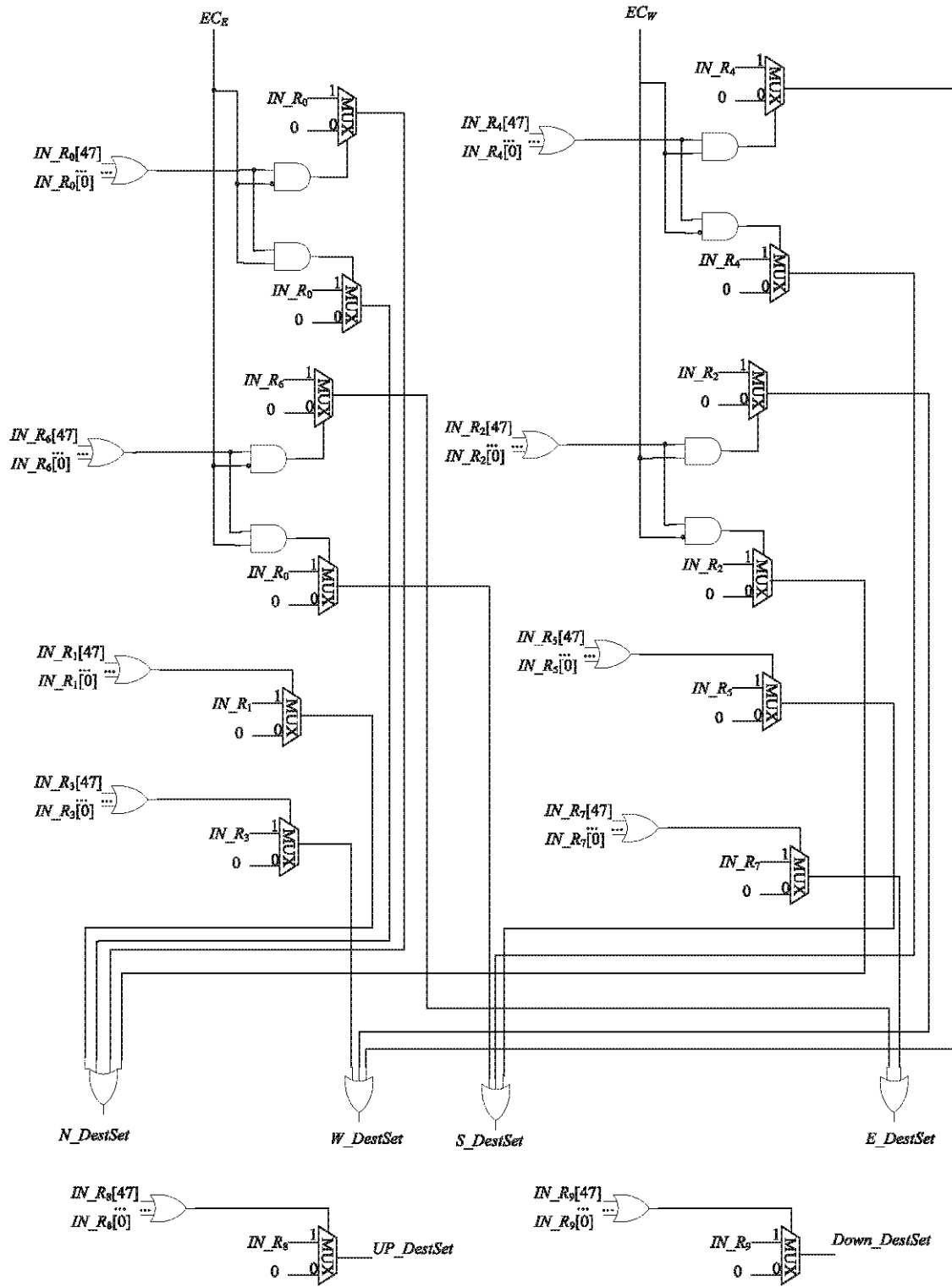
Fig. 14. The logic of MR sub-module for a $4 \times 4 \times 3$ 3-D mesh.

source node makes two copies of the packet, one following VN_0 and the other following VN_1. For destinations on the same X coordinate of the source, either virtual network can be chosen. The intermediate nodes never change the virtual network identification bit and only forward the packet to the same virtual network (Fig. 12).

6. Hardware implementation

To support the proposed multicasting algorithms in hardware, the multicast routing unit (MRU) is designed as in Fig. 13. The MRU is composed of three sub-modules.

Table 1
Simulation Configuration.

| Network size	$4 \times 4 \times 3$
Destination set size	8
Packet length (flits)	8
Flit size (bits)	75
VC depth	8
VC number	2
Distribution of the destinations	Uniform
C_{Hwire} (fF/mm)	212.12
C_{Vwire} (fF/mm)	600
Vias length (μm)	50
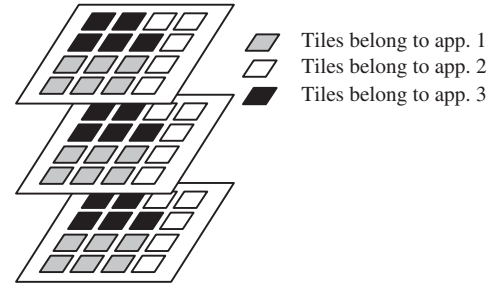
Fig. 16. Two sub-networks in the $4 \times 4 \times 3$ NoC system.

(1) Destination inclusion (DI) sub-module. For a multicast packet, this sub-module checks the sub-networks that the destination nodes belong to and sets the destination bit vectors of regions as defined in Section 4.1.

(2) Multicast routing calculation (MRC) sub-module. This sub-module determines the output directions of the multicast packet. For regular 3-D NoCs, the MXYZ algorithm in Fig. 6 is implemented in MRC. For irregular 3-D NoC sub-networks, the AL + XYZ algorithm in Fig. 11 is implemented as in Fig. 14.

(3) The extended connectivity bits selection sub-module. For regular 3-D NoCs, the extended bits are set to 1. For irregular 3-D NoC sub-networks, non-overlapped sub-networks are used for simplicity [12]. The extended bits are set by GM after allocating the applications. When virtualization is not assumed, i.e., only a single application is running in the NoC system, this module can be removed.

In order to investigate the overhead of hardware cost incurred by the multicast routing unit, the 7×7 multicast router (Fig. 3) implemented with AL + XYZ is compared with a 7×7 unicast router with no multicast routing unit, in terms of area and power consumption. Both routers are configured with two virtual channels per input port and flit size equal to 75 bits (supporting a $4 \times 4 \times 3$ network). The two designs are synthesized on using Synopses Design Compiler with TSMC 65 nm CMOS library. The power consumption and area of the AL + XYZ router are 50 mW and 153776 μm^2, respectively. As a reference, the power consumption and area of the 7×7 unicast router are 43 mW and 141775 μm^2, respectively. Thus, the area overhead of the AL + XYZ router over the unicast router is 8%.

7. Performance evaluation

In order to evaluate the performance of the proposed MXYZ and AL + XYZ, extensive experiments have been conducted with synthetic and real applications.
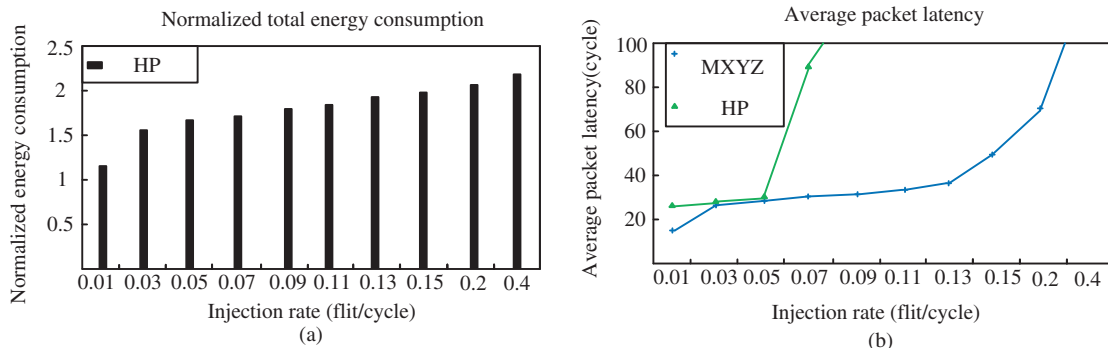
7.1. Experimental setup

Table 1 lists the simulation configuration. By default, each tile is composed of a processor, an L1 I/D cache, an L2 cache bank, and a router. In Section 7.4, another NoC architecture is considered. The cycle accurate NoC simulator Noxim [31] is extended to simulate the 3-D NoC systems. We measure the total communication energy consumption and average packet latency in the experiments. The power/energy model is defined in Section 3 which encloses the router synthesis results reported in Section 6. The leakage power is estimated from ORION2.0 [32]. The multicast packet latency is calculated as the average transmission cycles of all replicated packets.

Two sets of experiments are performed. The first set of experiments is used to compare MXYZ against the path-based hybrid partition algorithm (HP) [11] in regular 3-D NoC systems, in terms of energy consumption and packet latency. The second set of experiments is used to compare AL + XYZ and the multiple unicast routing (multiple UC) scheme in 3-D NoC systems with irregular sub-networks. In MUC, at the source node, multiple unicast packets are generated, each destined to a different destination. The XYZ routing is used as the base routing algorithm. In case of unavailable output channel, MUC selects an alternative output channel. In all the experiments, a metric named multicast to unicast ratio (MUR) is used to measure the ratio of the number of multicasting packets to that of unicasting packets. For each set of experiment, five experiments are performed and the average results are reported.

7.2. Efficiency of MXYZ in regular 3-D NoC systems

The efficiency of MXYZ is evaluated using random benchmarks. The whole network is used for one application. Fig. 15(a) shows the normalized energy consumption of HP over MXYZ, i.e., the energy consumption of HP divided by that of MXYZ. Fig. 15(a) shows that, when the injection rate is higher (e.g., above 0.09), the energy consumption of HP is about $1.7\times$ to $2.1\times$ over that of MXYZ. Fig. 15(b)
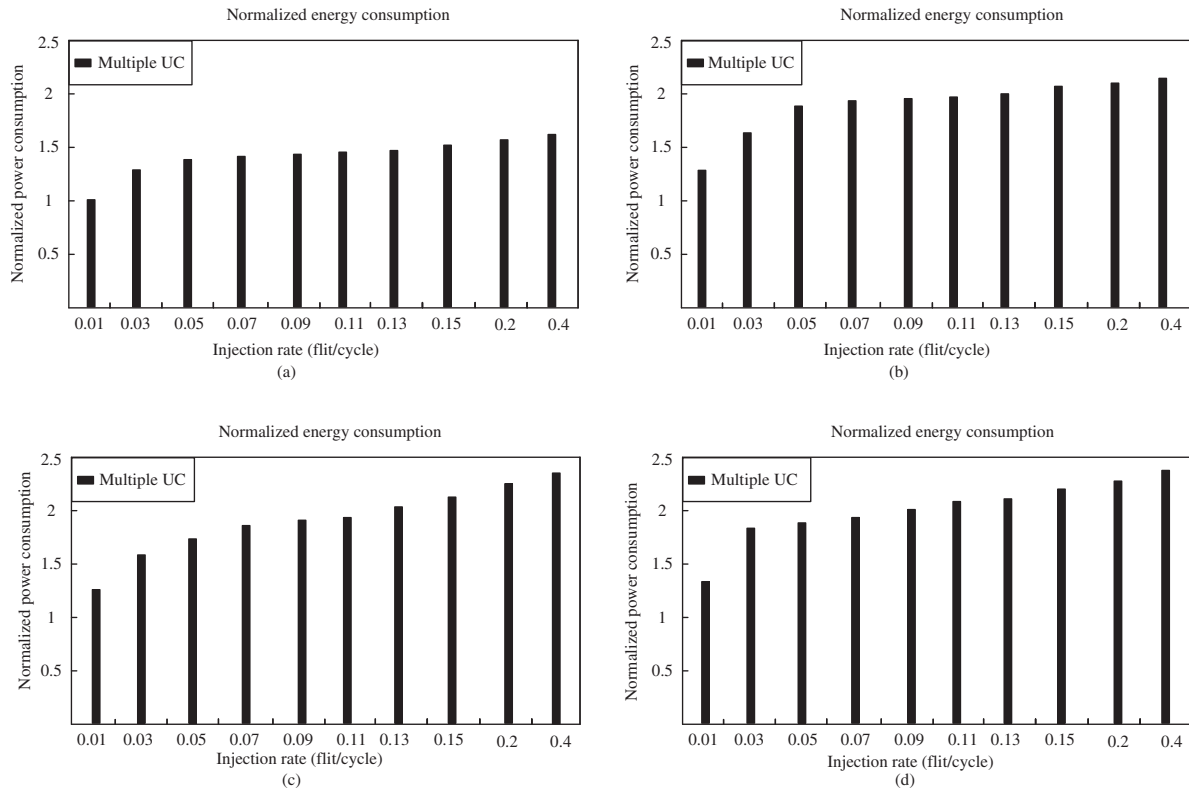

Fig. 15. Comparison of HP and MXYZ in terms of (a) total energy consumption (including both dynamic and leakage power) and (b) latency when MUR = 0.3.

Fig. 17. Comparison of MUC and AL + XYZ in terms of total energy consumption with MUR = (a) 0.05, (b) 0.2 (c) 0.25 and (d) = 0.3.
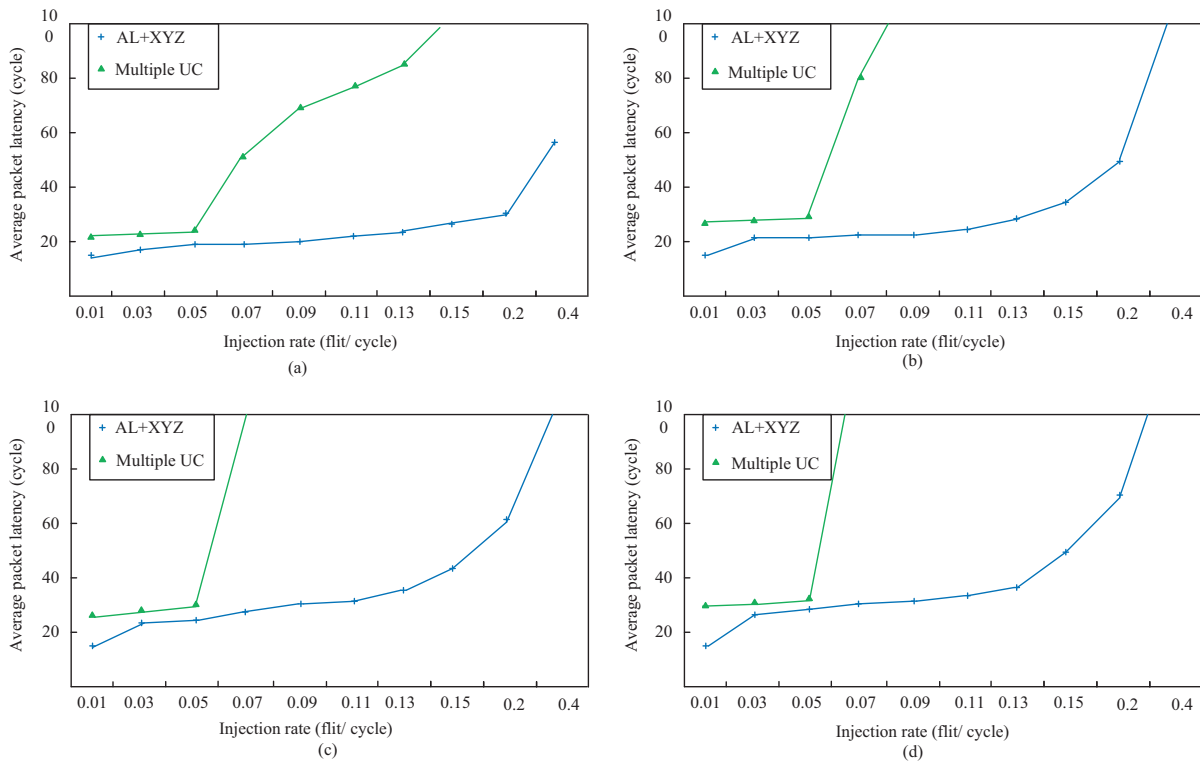

Fig. 18. Comparison of MUC and AL + XYZ in terms of latency with MUR = (a) 0.05, (b) 0.2 (c) 0.25 and (d) = 0.3.

shows the average packet latency results of HP and MXYZ. As the injection rate increases, the packet latency of HP increases more rapidly than that of MXYZ. The reason is that, the Hamiltonian paths generated by HP result in non-minimal paths to destination nodes. Thus, both the energy consumption and packet latency of the multicast traffic are increased.
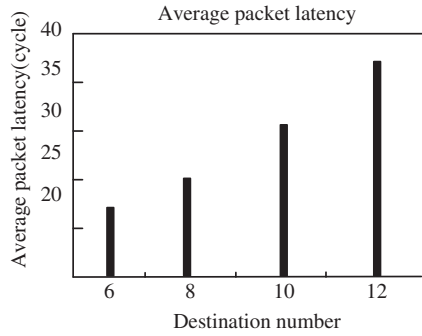
Fig. 19. Average latency of AL + XYZ when desitnation number equals to 6, 8, 10, and 12. MUR = 0.3, injection rate = 0.05.

7.3. Efficiency of AL + XYZ in 3-D NoC systems with irregular sub-networks under random benchmarks

In this set of simulations, the whole network is used for three applications. Fig. 16 shows the three sub-networks visualized in the $4 \times 4 \times 3$ NoC system.

Fig. 17 shows the energy comparison of AL + XYZ against multiple UC under different MUR values. Fig. 17(a) shows the normalized energy consumption of multiple UC over AL + XYZ when the MUR is low (MUR = 0.05) which means that the multicast communication occupies a small portion in the entire traffic. From Fig. 17(a), we can see that, when the injection rate is low (e.g., 0.01), the energy consumption of multiple UC is the same as that of AL + XYZ. When the injection rate increases, the energy consumption of multiple UC is about $1.3 \times$ to $1.4 \times$ over that of AL + XYZ.

When the multicast traffic increases, e.g., MUR is above 0.2 (Fig. 17(b) to (d)), AL + XYZ saves more energy compared to multiple UC. Fig. 17(d) shows the energy consumption when the multicasting communication has a large ratio over the entire traffic. When the injection rate increases (e.g., above 0.03), the energy consumption of multiple UC is higher than that of AL + XYZ by a

factor of 1.7–2.2. This can be attributed to the fact that multiple UC generates more packets than AL + XYZ, which results in higher energy consumption. The higher the MUR, the higher is the energy consumption overhead of multiple UC.

Fig. 18 shows the latency comparison of AL + XYZ against multiple UC under different MUR. Fig. 18(a) shows the average packet latency when MUR is low (MUR = 0.05). We can see that, when the injection rate is low (below 0.07), the packet latency of multiple UC is relatively low. When the injection rate is above 0.07, the latency of multiple UC increases rapidly. The increase in the latency of AL + XYZ is much slower until the injection rate is 0.4.

When the multicast traffic increases, e.g., MUR is above 0.2 (Fig. 18(b)–(d)), AL + XYZ has much lower packet latency compared to multiple UC. Fig. 18(d) shows the packet latency when MUR is high (MUR = 0.3). Multiple UC saturates at small injection rate (below 0.07) has much higher latency compared to that of AL + XYZ. The reason is that, multiple UC generates multiple unicast packets to support multicasting. Thus, the number of extra packets contributes to severe congestion and earlier saturation point. AL + XYZ, on the other hand, minimizes the number of multicast packets, which greatly helps alleviating the congestion situation.

Fig. 19 shows the average latency comparison of AL + XYZ under different destination numbers. The MUR is set to be 0.3 and injection rate is 0.05. The destination number is set to be 6, 8, 10, and 12. From Fig. 19, we can see that, when the destination number increases, the latency increases. When the destination number is 20, the latency is about $1.7 \times$ over that when the destination number is 12. The reason is that, more destinations will cause more communication traffic in the network, which tends to increase the possibility of congestion.

7.4. Efficiency of AL + XYZ in 3-D NoC systems with irregular sub-networks under real benchmarks

Two NoC architectures shown in Figs. 20(a) and 21(a) are configured in the simulator [33]. In the configuration of Fig. 20, each tile is composed of a processor, an L1 I/D cache, an L2 cache bank, and a router. In the configuration of Fig. 21, an L2 cache is placed in
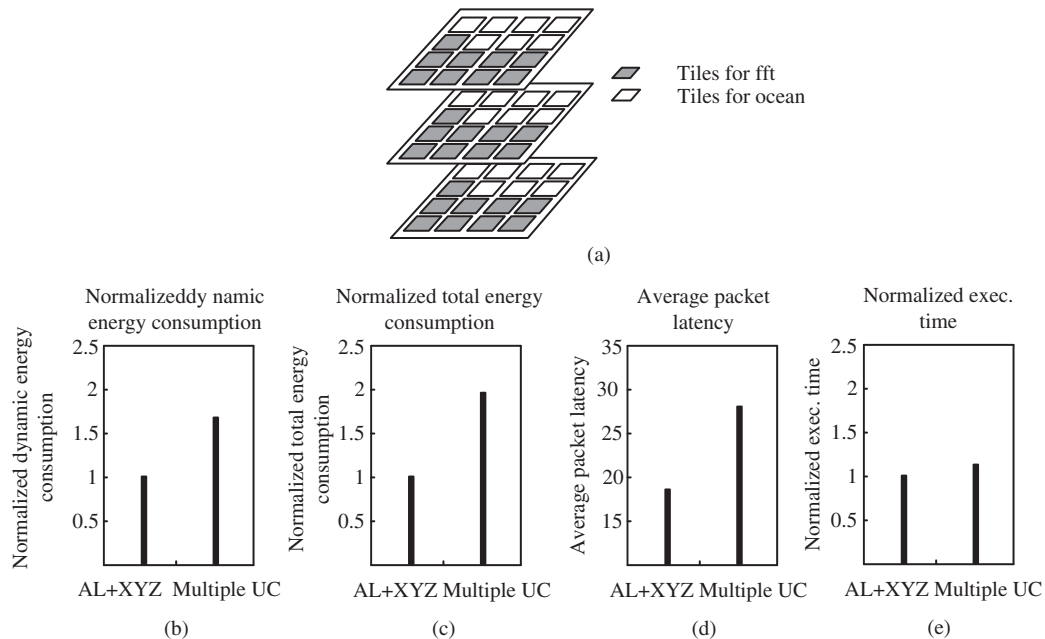

Fig. 20. (a) Two sub-networks in the $4 \times 4 \times 3$ NoC system. Each tile includes a processor, an L1 I/D cache, an L2 banck, and a router. Normalized dynamic energy consumption (b), total energy consumption (including both dynamic and leakage power) (c), latency (d), and normalized execution time (e) of AL + XYZ and Multiple UC.
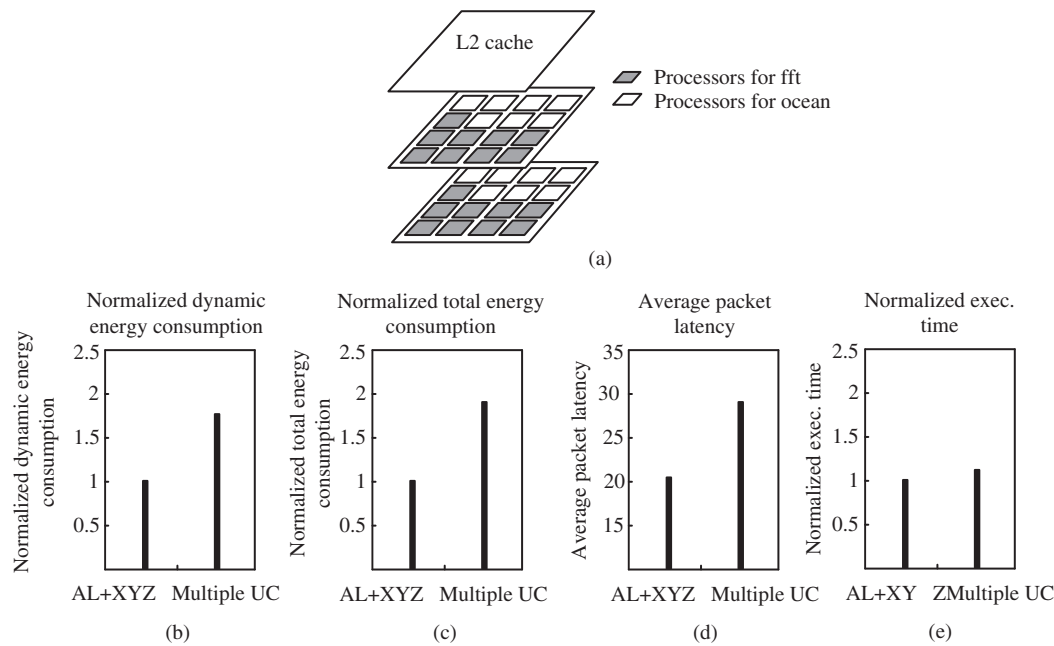
Fig. 21. (a) Two sub-networks in a different NoC architecture. An L2 cache is placed in a separate layer. The other tiles are composed of processors, L1 caches, and routers. Normalized dynamic energy consumption (b), total energy consumption (including both dynamic and leakage power) (c), latency (d), and normalized execution time (e) of AL + XYZ and Multiple UC.

a separate layer, while the other tiles are composed of processors, L1 caches, and routers. For real benchmarks, the communication traces of two SPLASH-2 benchmarks [34], fft and ocean are extracted from the same simulator [33]. Based on the traffic analysis of the traces, two synthetic traffics are generated with the same injection rate, size and distribution of destinations, and MUR as those of the original trace files. These two applications are mapped to the sub-networks as in Fig. 20(a).

Fig. 20 shows the energy consumption and packet latency results of AL + XYZ. The energy consumption of multiple unicast is normalized over the energy consumption of AL + XYZ. In Fig. 20(b), the dynamic energy consumption of multiple unicast is about $1.7\times$ over that of AL + XYZ. The reason is that, AL + XYZ could effectively reduce the total number of packets transmitted in the network, which results in significant reduction of dynamic energy consumption. In Fig. 20(c), we consider the total energy consumption, including both the dynamic and leakage power. We can see that, multiple UC consumes $2\times$ energy over that of AL + XYZ. The difference in Fig. 20(b) and (c) is caused by leakage power, which will be increased in case of congestion. AL + XYZ helps to reduce the possibility of congestion as it reduces the number of packets. The congestion is more severe in multiple UC, which has longer average packet latency. As shown in Fig. 20(d), the packet latency of multiple UC is about $1.5\times$ over that of AL + XYZ. In Fig. 20(e), the average execution time of the two applications under multiple unicast is about $1.15\times$ over that of AL + XYZ. As AL + XYZ reduces the number of packets and packet latency, the execution time is reduced consequently.

The efficiency of AL + XYZ in a different architecture as in Fig. 21(a) is also evaluated with the same two benchmark application traces [33,34]. Fig. 21(b) shows that, multiple UC consumes about $1.7\times$ dynamic energy over AL + XYZ. Fig. 21(c) shows that, multiple UC consumes about $1.9\times$ total energy over AL + XYZ. Fig. 21(d) shows the packet latency of multiple UC is about $1.4\times$ over that of AL + XYZ. In Fig. 21(e), the average execution time of the two applications under multiple UC is about $1.13\times$ over that of AL + XYZ.

8. Conclusion

In this paper, two multicast routing algorithms MXYZ and AL + XYZ have been proposed to support 3-D NoC systems. MXYZ is a dimension order multicast routing algorithm designed for regular 3-D NoC topologies. Based on the same design principle, AL + XYZ is proposed for irregular 3-D NoC topologies. In essence, if an output channel identified by a multicast routing algorithm designed for regular NoC systems (e.g., MXYZ) is not available, an alternative output channel is selected. The simulation results have confirmed that in regular 3-D NoCs, MXYZ outperforms a path-based multicast routing scheme HP in terms of packet latency and energy consumption. For 3-D NoC systems with irregular sub-networks, AL + XYZ achieves significant improvement in packet latency and energy consumption than multiple unicast routing under both random and real benchmarks.

References

[1] S. Borkar, Thousand core chips: a technology perspective, in: Proc. Design Automation Conf, 2007, pp. 746–749.
[2] V.F. Pavlidis, E.G. Friedman, 3-D topologies for networks-on-chip, IEEE Transactions on Very Large Scale Integration Systems 15 (2007) 1081–1090.
[3] W.R. Davis, J. Wilson, S. Mick, J. Xu, H. Hua, C. Mineo, A.M. Sule, M. Steer, P.D. Franzon, Demystifying 3D ICs: the pros and cons of going vertical, in: Proc. IEEE Design and Test of Computers, vol. 22, 2005, pp. 498–511.
[4] J. Duato, S. Yalamanchili, L. Ni, Interconnection Networks an Engineering Approach, Morgan Kaufmann, 2002.
[5] L. Wang, Y. Jin, H. Kim, E.J. Kim, Recursive partitioning multicast: a bandwidth-efficient routing for on-chip, in: Proc. ACM/IEEE Int'l Symp. Networks-on-Chip, 2009, pp. 64–73.
[6] N.E. Jerger, L.S. Peh, M. Lipasti, Virtual circuit tree multicasting: a case for on-chip hardware multicast support, in: Proc. Int'l Symp. Computer, Architecture, 2008, pp. 229–240.
[7] H.W. Cain, K.M. Lepak, B.A. Schwartz, M.H. Lipasti, Precise and Accurate Processor Simulation, Workshop on Computer Architecture Evaluation Using Commercial Workloads, 2002, pp. 13–22.

[8] M.M.K. Martin, D.J. Sorin, B.M. Beckmann, M.R. Marty, M. Xu, A.R. Alameldeen, K.E. Moore, M.D. Hill, D.A. Wood, Multifacet's general execution-driven multiprocessor simulator (GEMS) toolset, ACM SIGARCH Computer Architecture News 33 (2005) 92–99.

[9] N. Eisley, L.S. Peh, L. Shang, In-network cache coherence, in: Proc. IEEE/ACM Int'l Symp. Microarchitecture, 2006, pp. 321–332.

[10] S. Rodrigo, J. Flich, J. Duato, Efficient unicast and multicast support for CMPs, in: Proc. IEEE/ACM Int'l Symp. Microarchitecture, 2008, pp. 364–375.

[11] M. Ebrahimi, M. Daneshtalab, P. Liljeberg, H. Tenhunen, Partitioning methods for unicast/multicast traffic in 3D NoC architecture, in: Proc. IEEE Int'l Symp. Design and Diagnostics of Electronic Circuits and Systems, 2010, pp. 127–132.

[12] F. Triviño, J.L. Sánchez, F.J. Alfaro, J. Flich, Virtualizing network-on-chip resources in chip-multiprocessors, Microprocessors and Microsystems 35 (2011) 230–245.

[13] S. Murali, Designing Reliable and Efficient Networks on Chips, Springer Verlag, 2009.

[14] M.B. Taylor, J. Kim, J. Miller, D. Wentzlaff, F. Ghodrat, B. Greenwald, H. Hoffman, P. Johnson, J.W. Lee, W. Lee, The Raw microprocessor: a computational fabric for software circuits and general-purpose programs, IEEE Micro (2002) 25–35.

[15] X. Wang, M. Palesi, M. Yang, Y. Jiang, M. Huang, P. Liu, Power-aware run-time incremental mapping for 3-D networks-on-chip, in: Proc. IFIP Int'l Conf. Network and, Parallel Computing, 2011, pp. 232–247.

[16] C.L. Chou, R. Marculescu, User-aware dynamic task allocation in networks-on-chip, in: Proc. Conf Design, automation and Test in, Europe, 2008, pp. 1232–1237.

[17] X. Wang, M. Yang, Y. Jiang, P. Liu, On an efficient NoC multicasting scheme in support of multiple applications running on irregular sub-networks, Microprocessors and Microsystems 35 (2011) 119–129.

[18] X. Wang, M. Yang, Y. Jiang, P. Liu, Efficient multicasting scheme for irregular mesh-based NoCs, in: Proc. IEEE Int'l SoC Conf., 2010, pp. 384–387.

[19] X. Wang, M. Palesi, M. Yang, Y. Jiang, M. Huang, P. Liu, Low latency and energy efficient multicasting schemes for 3D NoC-based SoCs, in: Proc IFIP/IEEE Int'l Conf. VLSI-SoC, 2011, pp. 337–342.

[20] Z. Lu, B. Yin, A. Jantsch, Connection-oriented multicasting in wormhole-switched networks on chip, in: Proc. Emerging VLSI Technologies and Architectures, 2006, pp. 205–211.

[21] I.V. Senin, L. Mhamdi, K. Goossens, Efficient multicast support in buffered crossbars using networks on chip, in: Proc. IEEE Global Telecommunications Conference, 2009, pp. 1–7.

[22] P. Abad, V. Puente, J.A. Gregorio, MRR: enabling fully adaptive multicast routing for CMP interconnection networks, in: Proc. Int'l Conf. High-Performance Computer Architecture, 2009, pp. 355–366.

[23] F.A. Samman, T. Hollstein, M. Glesner, Planar adaptive router microarchitecture for tree-based multicast network-on-chip, in: Proc. Int'l Workshop on Network on Chip Architectures, 2008, pp. 6–13.

[24] C. Xiang, Design and Verification of Network on Chip Components (in Chinese), Dept. Information Science and Electronic Engineering Zhejiang University, Hangzhou, 2008.

[25] A. Mejia, M. Palesi, J. Flich, S. Kumar, P. López, R. Holsmark, J. Duato, Region-based routing: a mechanism to support efficient routing algorithms in NoCs, IEEE Transactions on Very Large Scale Integration (VLSI) Systems 17 (2009) 356–369.

[26] M. Ebrahimi, M. Daneshtalab, P. Liljeberg, J. Plosila, H. Tenhunen, Exploring partitioning methods for 3D Networks-on-Chip utilizing adaptive routing model, in: Proc. IEEE/ACM Int'l Symp. Networks on Chip, 2011, pp. 73–80.

[27] V. Varavithya, P. Mohapatra, Asynchronous tree-based multicasting in wormhole-switched MINs, IEEE Transactions on Parallel and Distributed Systems 10 (1999) 1159–1178.

[28] H. Matsutani, M. Koibuchi, H. Amano, Tightly-coupled multi-layer topologies for 3-D NoCs, in: Proc. Int'l Conf. Parallel Processing, 2007, pp. 75–85.

[29] J. Srinivasan, S.V. Adve, P. Bose, J.A. Rivers, The impact of technology scaling on lifetime reliability, in: Proc. Int'l Conf. Dependable Systems and Networks 2004, pp. 177–187.

[30] C.L. Chou, U.Y. Ogras, R. Marculescu, Energy-and performance-aware incremental mapping for networks on chip with multiple voltage levels, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 27 (2008) 1866–1879.

[31] F. Fazzino, M. Palesi, D. Patti, Noxim. Available From: <http://sourceforge.net/projects/noxim>.

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

[33] J. Xue, A. Garg, B. Ciftcioglu, J. Hu, S. Wang, I. Savidis, M. Jain, R. Berman, P. Liu, M. Huang, H. Wu, E.G. Friedman, G. Wicks, and D. Moore, An intra-chip free-space optical interconnect, in: Proc. Int'l Symp. Computer, Architecture, 2010, pp. 94–105.

[34] J.P. Singh, W.D. Weber, A. Gupta, SPLASH: stanford parallel applications for shared-memory, ACM SIGARCH Computer Architecture News (1992) 5–44.

Xiaohang Wang received the B. Eng. and Ph.D degree in communication and electronic engineering from Zhejiang University, in 2006 and 2011. He is currently an assistant professor at Guangzhou Institute of Advanced Technology, Chinese Academy of Science. His research interests include NoC system simulation, routing algorithm, and parallel programming for NoC based systems.

Mei Yang received her Ph. D. in Computer Science from the University of Texas at Dallas in Aug. 2003. She has been an associate professor in the Department of Electrical and Computer Engineering, University of Nevada, Las Vegas since 2010. Her research interests include computer architectures, networking, and embedded systems.
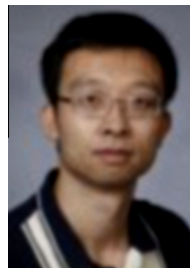

Yingtao Jiang received his Ph. D. in Computer Science from the University of Texas at Dallas in Aug. 2001. He joined the Department of Electrical and Computer Engineering, University of Nevada, Las Vegas in Aug. 2001. He has been an associate professor since Aug. 2007. His research interests include algorithms, computer architectures, VLSI, networking, nano technologies, etc.

Maurizio Palesi received the M.S. and Ph.D. degrees in computer engineering from the University of Catania, Italy, in 1999 and 2003, respectively. Since November 2010 he is Assistant Professor at Kore University, Italy. Dr. Palesi serves on the Editorial Board of VLSI Design journal as an Associate Editor since May 2007. He has served as a Guest Editor for the VLSI Design Journal, the International Journal of High Performance Systems Architecture, Elsevier MICPRO Journal and ACM Transactions on Embedded Computing Systems. He is in the Technical Program Committee of several IEEE/ACM International Conferences including DATE, RTAS, CODES + ISSS, ESTIMedia, NOCS, SOCC, VLSI, ISC, and SITIS. He has been co-organizer of the four editions of the International Workshop on Network-on-Chip Architectures (from 2008 to 2011). Dr. Palesi is member of the European Network of Excellence on High Performance and Embedded Architecture and Compilation (HiPEAC).

Peng Liu received the B.Eng. and M.Eng. degrees in optical engineering from Zhejiang University, in 1992, and 1996, respectively, and the Ph.D. degree in communication and electronic engineering from Zhejiang University, China, in 1999. He has been an associate professor with the information science and electronic engineering Department, Zhejiang University, Hangzhou, China, since 2002. His research focuses embedded processor microarchitecture, multiprocessor system-on-chip architectures, on-chip interconnection networks, parallel programming, and VLSI design.
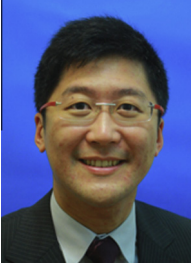
Terrence Mak is an Assistant Professor at the Department of Computer Science and Engineering, the Chinese University of Hong Kong. He also serves as a Director of the Intelligent Chips and Systems Research Centre, which is jointly supported by Guangzhou Institute of Advanced Technology and Chinese Academy of Sciences, in Nansha, Guangzhou. He did his Ph.D in EEE department at Imperial College London and became a lecturer in the School of Electrical, Electronic and Computer Engineering at Newcastle University. He was also an affiliated lecturer in the Institute of Neuroscience at the same University. Supported by the Royal Society, he holds a visiting scientist position at MIT. Previously, he worked with Prof. Ivan Sutherland at Sun Microsystems Labs at Menlo Park. He was awarded the Croucher Foundation Scholarship, the US Navel Research Excellence in Neuroengineering Award, the Best Paper Award at DATE'2011 and the One-Thousand-Youth-Talents Award in China. He has published over 60 research papers in peer-reviewed journals and international conferences.

Nader Bagherzadeh is a professor at Electrical Engineering and Computer Science, University of California, Irvine. His current research area is involved with the design of next generation System-on-Chip (SoC) based on the notion of Network-on-Chip (NoC) architecture for connecting 100's of cores on the same die. He has worked on low power routers, wired and wireless on-chip communication, mapping and scheduling algorithms, as well as the fault-tolerance aspects of the NoC architectures.