Partitioning Methods for Unicast/Multicast Traffic in 3D NoC Architecture

Masoumeh Ebrahimi[1,2], Masoud Daneshtalab[2], Pasi Liljeberg[2], Hannu Tenhunen[1,2]
[1]Turku Centre for Computer Science (TUCS), Joukahaisenkatu 3-5, 20520 Turku, Finland
[2]Department of Information Technology, University of Turku, Finland
{ masebr, masdan, pakrli, hanten}@utu.fi

***Abstract*—as the scale of integration grows, the interconnection problem becomes one of the major design considerations of Multi Processor System on Chip (MPSoC). In recent years, many researchers have conducted studies on 3D IC designs stacking multiple layers on top of each other. In order to decrease the transmission delay of unicast/multicast messages in a network based multicore system, the network is divided into several partitions. In this paper, we first introduce a novel idea of balanced partitioning that allows the network to be partitioned effectively. Then, we propose a set of partitioning approaches each with a different level of efficiency. In addition, we present an advantageous method based on the idea of balanced partitioning to provide a high degree of parallelism with a considerable reduction of packet delay in unicast/multicast traffic. Simulations are provided to evaluate and compare the performance of proposed methods.**

I. INTRODUCTION

As the technology scaling allows dozens or hundreds of processing elements to be integrated on a single chip, the interconnection between processing elements become more and more complicated and inefficient with traditional approaches such as bus architecture. In order to overcome these problems, many researchers have focused on the communication architecture of 2D NoCs which provides more efficient communication through the processing elements. However, since a 2D NoC architecture has limited floor-planning choices, the performance degrades when the number of processing elements increases. Considering these bottlenecks, the technology is moving rapidly towards the concept of 3D NoCs where multiple active silicon layers are vertically stacked. The major advantages of 3D NoCs are the considerable reduction in the average wire length and wire delay, resulting in lower power consumption and better performance [1][2].

The choice of routing protocols can have a large impact on performance, latency and power consumption. The routing protocols in NoCs and MPSoCs can be unicast or multicast. In the unicast communication a message is sent from a source node to a single destination node, while in the multicast communication a message is delivered from one source node to an arbitrary number of destination nodes. The multicast is a special case of unicast while the unicast routing cannot support multicast messages efficiency. Consequently, a fundamental issue in the multicast communication is how to determine efficient message routing. The tree-based [3] and path-based [2] methods are two famous hardware-based schemes employed in the multicast communication. In tree-based routing, a destination set is partitioned at the source node and separate copies are sent on one or more outgoing channels. A message may be replicated at intermediate nodes and forwarded along multiple outgoing channels toward disjoint subsets of destinations. Since no ordering of destinations is required, the startup latency is low. However, the decision concerning routing and replication is made using all header flits in each node. The most severe drawback of this method is the high probability of message blocking at intermediate nodes leading to higher network contention.

In the path-based method, a source node prepares a message for delivery to a set of destinations by first sorting the addresses of destinations in the order in which they are delivered. The multicast message can be represented by m=(s,D), where s is the source node, D = {d1,d2, . . . ,dn} is the set of ordered destination nodes, and n is the number of destination nodes. The message is routed on the path until it reaches a router with address d1, then the address d1 is removed from the header and subsequent flits are delivered both to the local core and next destination (d2). In this way, the message is eventually delivered to all specified destinations in the header. Hence, path-based routing does not suffer from message blocking. Moreover the routing decision is made only by using the first address in the header. However, it requires destinations to be sorted at the source node which imposes higher startup latency. Therefore, the major concerns in path-based methods are reducing the startup latency and the overall path length traveled by the messages. Some researches show that path-based routing is more beneficial in the wormhole switched network while in networks employing store-and-forward and virtual cut-through, tree-based routing is advantageous [3].

In wormhole routing, the message is divided into small flits that travel through the network in a pipelined fashion and therefore eliminate the need to allocate large buffers in the intermediate nodes along the path [4]. The main drawback of the wormhole switching is the possibility of deadlock arising, prohibiting other messages from using the occupied links and buffers and therefore wasting channel bandwidth. For preventing deadlock in path-based routing, the Hamiltonian path [6] was introduced. Different partitioning methods can be defined in the Hamiltonian-based networks in order to reduce the overall paths traveled by messages. On the other hand, methods for preventing deadlock essentially depend on the underlying topology. Different topologies can be considered for each layer of 3D NoCs such as mesh, torus, and ring. In this work we limit our considerations to 3D mesh NoCs in which each layer consists of a 2D mesh.

The remainder of this paper is organized as follows: Section II reviews related work. A brief background on the Hamiltonian Path and network partitioning are provided in Section III. The proposed partitioning methods are discussed in Section IV. The results are given in Section V while we summarize and conclude in the last section.

II. RELATED WORK

Due to the fact that the multicast communication is used commonly in various parallel applications and can be used to support several other collective communication operations, there have been several attempts to improve the performance of multicast communication in 2D NoCs. Virtual Circuit Tree Multicasting (VCTM) [7], Recursive Partitioning Multicast (RPM) [8] and Low Distance multicast (LD) [9] are three recent works in the realm of NoC in which the RPM and VCTM are based on the tree-based routing algorithm while LD is based on the path-based method. In VCTM, a set up packet is sent from a source node to all destinations in order to build a virtual circuit tree, then the multicast packet is send down the tree. The RPM, which supposes the network is divided into several partitions, minimizes the packet replication time by defining priority rules between the directions which can be selected to reach those partitions. The LD algorithm optimized destination ordering and utilized adaptive routing for both the unicast and multicast messages through the network.

Some researches have been taken on evaluating the performance metric of 3D NoCs and deciding whether to choose a 2D or 3D NoC. In [1], the authors perform a performance evaluation between 2D and 3D NoCs by demonstrating both mesh and tree-based architectures. The comparisons performed by applying real traffic patterns in a cycle accurate simulation. Another discussion on 2D and 3D architectures had been addressed in [10] which resulted to propose a hybrid model by combination of 3D NoC and bus architecture. This model provides considerable latency reduction in vertical interconnects.

A method closely related to our work is discussed in [11] where the Hamiltonian Path is employed for adaptive multicast routing in 3D mesh based networks.

In this paper, we introduce a novel idea of balanced partitioning in 3D mesh network for unicast/multicast traffic. We also present several partitioning methods each with a different level of efficiency.

III. Preliminaries

A. Hamiltonian Path

The Hamiltonian Path strategy guaranties that the network will be free of deadlocks for unicast and multicast traffic. The Hamiltonian Path visits each node exactly once along the path. As shown in Fig. 1(a), for each node a label is assigned from 1 to N-1 in which N is the number of nodes in the network. Several Hamiltonian Paths can be considered in the mesh topology. In 3D a×b×c mesh, each node is presented by the ordered triple (x,y,z) where x is the X-coordinate, y is the Y-coordinate and z is the Z-coordinate. The following equations show one possibility of assigning the labels which we utilize in this paper:

$$L(x,y,z)=\{(a\times b\times z)+(a\times y)+(x)\} \quad where z: even, y: even$$
$$L(x,y,z)=\{(a\times b\times z)+(a\times y)+(a-x-1)\} \quad where z: even, y: odd$$
$$L(x,y,z)=\{(a\times b\times z)+(a\times(b-y-1))+(a-x-1)\} \quad where z: odd, y: even$$
$$L(x,y,z)=\{(a\times b\times z)+(a\times(b-y-1))+(x)\} \quad where z: odd, y: odd$$

As exhibited in Fig. 1, two directed Hamiltonian Paths (or two subnetworks) are constructed by the labelling [9]. The high channel subnetwork starts at node 0, and the low channel subnetwork ends at node 0. In case the label of the destination node is greater than the label of the source node, the routing always takes place in the high subnetwork; otherwise it takes place in the low subnetwork. The destinations are placed into two groups. One group contains all the destinations that could be reached using the high subnetwork, and the other contains the remaining destinations that could be reached using the low subnetwork. To reduce the path length, the vertical channels that are not part of the Hamiltonian Path (the dashed lines in Fig. 1) could be used in appropriate directions.

B. Network Partitioning

The performance of multicast communication is measured in terms of its latency in delivering a message to all destinations. Multicast latency consists of two parts: startup latency and network latency. The startup latency is the time required to break a message into packets (each with different destinations), prepare packets, and deliver them completely to the network. The network latency is defined as the time between the first flit is injected to the network until the tail flit of all packets has reached corresponding destinations. Partitioning methods deal with reducing network latency by dividing the network into several partitions and therefore reducing the overall path length. Nevertheless, breaking the network into partitions has differing constraints as follows:

- Increasing network partitions leads to additional startup latency probably due to the preparation time of more packets at the source node.
- Breaking the network into unbalanced partitions results to create long paths in the network. Therefore, they increase the latency to reach the last destination which increases network latency for multicast packets. We call this factor “fairness”.

In the following example we show how a balanced partitioning can affect the network latency for multicast packets. The network latency is assumed to be measured by the number of hops required for delivering the message from the source node to any destination. Fig. 2 illustrates a 3×3×3 mesh NoC where the source node is 13 in Fig. 2(a), and 4 in Fig. 2(b). We suppose the network is divided into two partitions (dark and bright partitions in Fig. 2) and use the same routing method for each partition. Since the size of each partition is directly related to the source node position, the level of fairness among two partitions is high in Fig. 2(a) while it is low in Fig. 2(b). Now consider the case when a multicast message, is generated for destinations D ={3,8,17,22}. The message should be put into two different packets and route via corresponding partitions. As in Fig. 2(a), one packet requires 4 hops while the other travels 5 hops to reach their destinations. Since two packets route simultaneously, they reach to their destinations in comparable arrival times which can be estimated as 5 hops. However, in Fig. 2(b) one packet takes 1 hop while the other requires 10 hops, hence the network latency is estimated as the maximum which is 10 hops. Both Fig. 2(a), and Fig. 2(b) met two startups due to deliver two packets from the source node. As depicted in these figures, the balanced partitioning allows to achieve high parallelism while considerably reduces the network latency. In general, this method is not fair, because it allows partitions to become unbalanced.

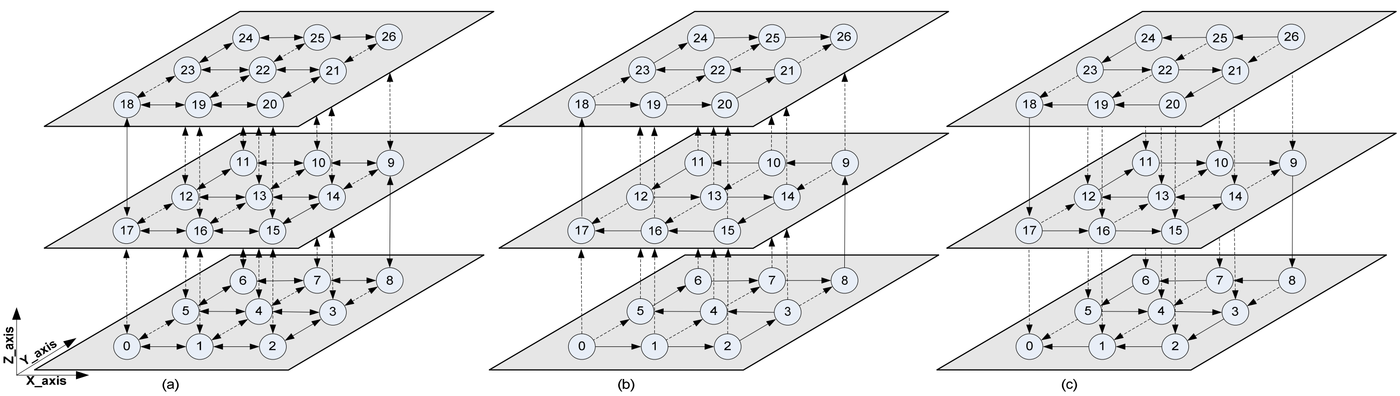
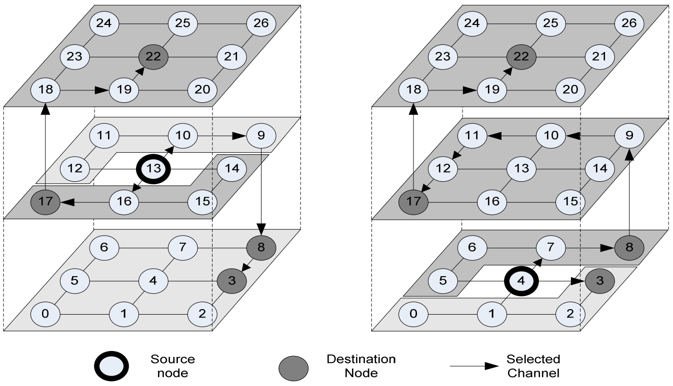


Fig. 1. A 3×3×3 mesh physical network with the label assignment (a) high channel and (b) low channel subnetworks.
The solid lines indicate the Hamiltonian Path and dashed lines indicate the links that could be used to reduce path length in routing.



(a) (b)

Fig. 2. An example of: (a) balanced partitions, (b) unbalanced partitions

The main contributions of this paper are the following: 1) the exploration and evaluation of various partitioning methods with a different level of efficiency. 2) presenting a fairness partitioning method which provides a high degree of parallelism with a considerable reduction in communication latency under unicast/multicast traffic.

IV. Partitioing Methods

In this section, we provide a detailed explanation of proposed methods named Two-Block Partitioning (TBP), Multi-Block Partitioning (MBP), Vertical-Block Partitioning (VBP) and Hybrid Partitioning (HP). The TBP and MBP methods are extensions of known methods called Dual-Path [3] and Multi-Path [3]to the realm of 3D architecture while the VBP and HP methods are two advantageous methods which proposed particularly for 3D networks. In the last part, the deadlock avoidance is discussed.

A. Two-Block Partitioning (TBP)

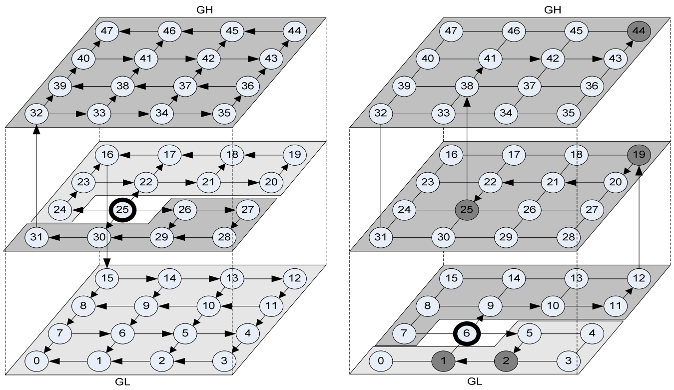
In this scheme, the network is partitioned into high channel and low channel subnetworks. The high channel subnetwork contains all directional channels with nodes labeled in ascending order, and the low channel subnetwork contains all directional channels with nodes labelled in descending order. In this method all destination nodes are split at most into two disjoint groups: a high group and a low group. The high group consists of all destination nodes with the higher labels than the source node and the low group contains all destination nodes with the lower labels.

When considering label assignment described in Section III, all destination nodes located in the same layer as the source node are divided at most into high and low groups while all destinations in higher (lower) layers are put in the high (low) group. In addition, one packet is created for each group and the destinations within each packet should be sorted in the correct order in which they are visited in the path. Therefore, destinations in the high group should be sorted in ascending order and other destinations in descending order. The created packets are routed via high and low channel subnetworks. The TBP algorithm is shown in Fig. 3.

Fig. 4(a) shows an example of the partitioning policy and the portions of each partition that depends on the source node position. As illustrated in Fig. 4(a) if the source node is located at a middle layer, two partitions cover comparable number of nodes but still with a large number of nodes in both partitions. However in Fig. 4(b) one partition contains considerably more nodes than the other. Now, suppose that the multicast message m=(6,{1,2,19,25,44}) is generated by the core. The destinations are split into two groups: GH={19,25,44} and GL={1,2}. the packet created for G_H uses the Hamiltonian Path as follows: {6,9,10,11,12,**19**,20,21,22,**25**,38,41,42,43,**44**} where at least 14 hops are needed to reach the last destination. The packet path for the G_L is:{6,5,**2**,**1**} where 3 hops are required.

Algorithm: Two-Block Partitioning (TBP)
Inputs: a×b×c network; destinations labels; source label;
Outputs: high group (G_H) and low group (G_L);
Begin
For "i: 0 to number of destinations" loop
If (destinationLabel(i) > sourceLabel) then
G_H <= destinationAddress;
--Meanwhile sort G_H in ascending order
Else
G_L <= destinationAddress;
--Meanwhile sort G_L in descending order
End if;
End loop; --Construct a packet for each group
End TBP;

Fig. 3. The TBP method algorithm



(a) (b)

Fig. 4. The TBP method (a) balanced partitions (b) unbalanced partitions

B. Multi-Block Partitioning (MBP)

In this scheme, the network is partitioned into two subnetworks as in the TBP method and then each subnetwork is split into two parts. As a result, the network is partitioned into four subnetworks (high1, high2, low1 and low2) and at most one packet per subnetwork is created and delivered to the network. The MBP algorithm is shown in Fig. 5.

All nodes located in the same layer as the source node are divided into at most four groups while all nodes in higher (lower) layers will be divided into at most high1 and high2 (low1 and low2) groups. Fig. 6 illustrates the partitioning policy where four partitions are formed in the network. The size of the partitions is comparable in Fig. 6(a) while the number of nodes in each partition is unbalanced in Fig. 6(b). Considering the multicast message m=(6,{1,2,19,25,44}). The destinations are split into three groups as GH2 ={19,25,44}, GL1={1} and GL2={2} the taken paths for three generated packets are {6,9,10,11,12,**19**,20,21,22,**25**,38,41,42,43,**44**}, {6,**1**} and {6,5,**2**} hence, the maximum hop is 14.

To sum up, this method attempts to reduce the overall path length by delivering up to four packets from the source node. Nevertheless it does not perform well in path length reduction and still suffers from unfair partitioning.

C. Vertical-Block Partitioning (VBP)

In this method, similar to the TBP method, the network is partitioned into high and low subnetworks; destination nodes are divided into high and low groups and then sorted in each group. In the next step, each subnetwork is vertically partitioned in which the nodes with the same *x* value will be put in the same group. The algorithm is shown in Fig. 7. This scheme has several advantages over the TBP and MBP methods as it achieves a high degree of parallelism; avoids

Algorithm: Multi-Block Partitioning (MBP)
Inputs: a×b×c network; destinations labels; source labels;
The X value of source node (X_s) and destinations (X_d)
Outputs: high groups (G_{H1},G_{H2});low groups (G_{L1},G_{L2});
Begin
For "i: 0 to number of destinations" loop
If (destinationLabel(i) > sourceLabel) then
If (($X_d(i) \le X_s$) and (b , c are both odd or even)) or
(($X_d(i) < X_s$) and not(b , c are both odd or even) then
G_{H1} <= destinationAddress;
Else
G_{H2} <= destinationAddress;
End if; --Meanwhile sort G_{H1}, G_{H2} in ascending order
Else
If (($X_d(i) < X_s$) and (b , c are both odd or even)) or
(($X_d(i) \le X_s$) and not(b , c are both odd or even) then
G_{L1} <= destinationAddress;
Else
G_{L2} <= destinationAddress;
End if --Meanwhile sort G_{L1}, G_{L2} in descending order
End if;
End loop; --Construct a packet for each group
End MBP;

Fig. 5. The MBP method algorithm

Algorithm: Vertical-Block Partitioning (VBP)
Inputs: a×b×c network; destinations labels; source label;
The X value of destinations (X_d)
Outputs:high groups(G_{H1},G_{H2},..G_{Ha}),low groups(G_{L1},G_{L2},.., G_{La})
Begin
For "i: 0 to number of destinations" loop
If (destinationLabel(i) > sourceLabel) then
Case "$X_d(i)$" is
When 1 => G_{H1} <= destinationAddress;
When 2 => G_{H2} <= destinationAddress;
…
When a => G_{Ha} <= destinationAddress;
End Case;--Meanwhile sort G_{H1},.. G_{Ha} in ascending order
Else
Case "$X_d(i)$" is
When 1 => G_{L1} <= destinationAddress;
When 2 => G_{L2} <= destinationAddress;
…
When a => G_{La} <= destinationAddress;
End Case;--Meanwhile sort G_{L1},..G_{La}in descending order
End if;
End loop; --Construct a packet for each group
End VBP;

Fig. 7. The VBP method algorithm

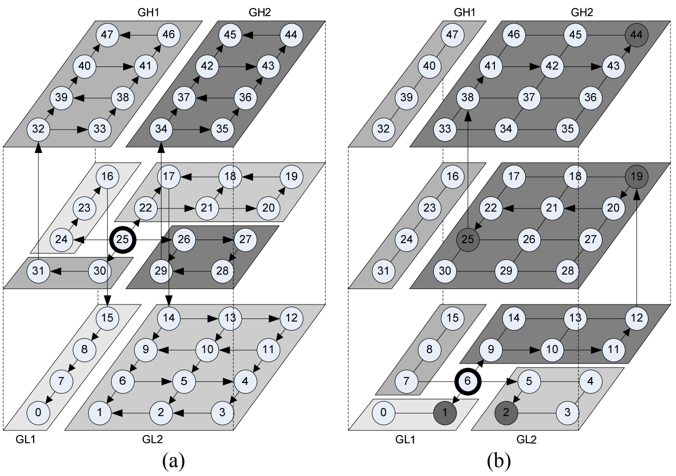


Fig. 6. The MBP method (a) balanced partitions (b) unbalanced partitions

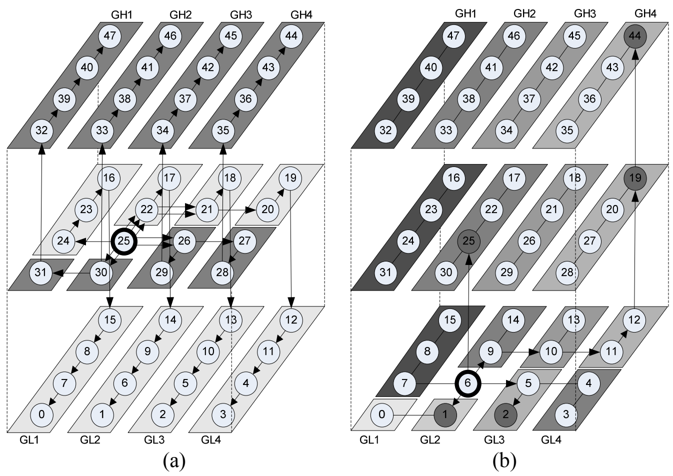


Fig. 8. The VBP method (a) balanced partitions (b)unbalanced partitions

the creation of long paths and reduces the network latency. However the VBP method increases the startup latency due to using up to 2×a packets in a×b×c network.

As shown in Fig. 8, this scheme does not guarantee the fairness among partitions as it is fair when the source node located at 25 while four groups will cover more nodes than the other groups when the source node is at 6. Moreover, the time required to prepare and deliver at most 2×4 packets is considered as the startup latency. For multicast message m=(6,{1,2,19,25,44}), four groups are formed: G_{H2}={25},G_{H4}={19,44},G_{L2}={1} and G_{L3}={2}. One packet is generated for each group and packet paths are {6,**25**},{6,9,10,11,12,**19**,**44**},{6,**1**} and {6,5,**2**} in which the maximum hop is 6.

D. Hybrid Partitioning (HP)

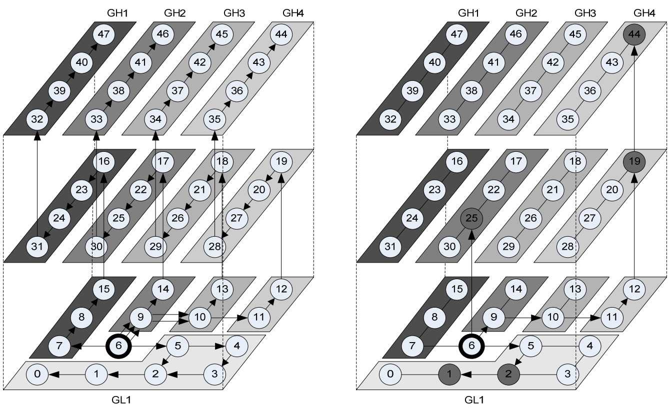
All the aforementioned methods (The TBP, MBP, and VBP), perform the partitioning in a highly sequential manner and do not consider the communication latency regarding unfair partitions and different message arrival times at the destinations. As already mentioned, different methods have different levels of efficiency depending on the source node position. The key idea of hybrid partitioning is based on a mixed method that combines the best features of the two previously mentioned approaches, the TBP and VBP methods. The whole algorithm, which is shown in Fig. 9, can be expressed as follows:

Step 1: Partitions formed after employing the TBP method can be inter-layer or multi-layer partitions. The inter-layer partition covers less than or equal to *a×b* nodes in a×b×c network while the multi-layer partition covers more than one layer. Suppose that the source node is located at node 6. After employing the TBP method as shown in Fig. 4(b), high subnetwork covers more than two layers (41 nodes) and thus it is a multi-layer partition. The low subnetwork is an inter-layer partition due to covering only 6 nodes. According to the source node position, the network can be divided into one inter-layer and one multi-layer partition or two multi-layer partitions.

Step 2: As the VBP method avoids long paths by breaking the network into vertical partitions, it is an efficient method for multi-layer partitions. In order to find a suitable method for inter-layer partitions, the number of nodes in the network is divided by value *a* in a×b×c mesh network. The obtained value determines an average number of nodes per vertical group. If the number of nodes in an inter-layer partition is equal or less than this value, the default partitioning method, the TBP method, is selected, otherwise partitioning is continued by applying the VBP method. As illustrated in Fig. 10(a), the source node divides the network into one inter-layer and one multi-layer partition using the TBP method. For the multi-layer partition, the VBP method is applied while the TBP method is used for the inter-layer partition. The TBP method is adopted as the inter-layer partition because of consisting less nodes than the evaluated value (48/4=12). For multicast message m=(6,{1,2,19,25,44}), three packets are formed and their paths are {6,9,10,11,12,**19**,**44**},{6,**25**} and {6,5,**2**,**1**} with 6 hops as the maximum latency. As a result, the number of nodes is comparable among partitions and the startup latency is less than the VBP method. However, considering the node 25 as the source node position and applying the TBP method, two multi-layer partitions are formed; therefore the VBP method is used for both partitions as illustrated in Fig. 8(a). In brief, this scheme has a similar performance in avoiding long paths and increasing parallelism as the VBP method. Moreover, this method achieves higher performance by decreasing the startup latency.

Algorithm: Hybrid Partitioning (HP)
Inputs: a×b×c network; destinations labels; source label;
Outputs: highgroups(G_{H1},G_{H2},.., G_{Ha}),lowgroups(G_{L1},G_{L2},.., G_{La})
Begin
n <= number of nodes in the network;
If (sourceLabel ≤ (n/a)) then
G_L : applying Two-Block Partitioning method;
elsif ((n - sourceLabel) ≤ (n/a)) then
G_H : applying Two-Block Partitioning method;
Else
G_L : applying Vertical-Block Partitioning method;
G_H : applying Vertical-Block Partitioning method;
End if; --Construct a packet for each group
End HP:

Fig. 9. The HP method algorithm



(a) (b)

Fig. 10. The HP method (a) balanced partitions by employing VBP for the high and TBP for the low subnetwork (b) an example of muticast message

E. Deadlock Avoidance

Deadlock is a situation in which the network resources continuously wait for each other to be released. Our proposed partitioning methods utilizing the routing algorithm based on the Hamiltonian Path guaranties that the network will be free of deadlocks because: 1) the intersection of all formed partitions is always empty 2) all paths are followed in ascending or descending order and therefore no cyclic dependency will be formed.

V. RESULTS AND DISCUSSION

To assess the efficiency of the proposed partitioning models, we have developed a synthesizable NoC simulator implemented in VHDL and is based on wormhole switching in 3D mesh configuration. The simulator inputs include the array size, the routing algorithm, the link width, buffer size, and the traffic type. For all routers, the data width was set to 32 bits, and each input channel has a buffer (FIFO) size of 12 flits. The message size was assumed to be 6 flits. For the performance metric, we use the multicast latency defined as the number of cycles between the generation of a multicast message until the tail of the multicast message reaches all the destinations.

A. Multicast Traffic Profile

The first set of simulations was performed for a random traffic profile. The array size was considered to be 4×4×4. In the multicast traffic profile, each core sends a message to a set of destinations. A uniform distribution is used to construct the destination set of each multicast message [6]. The number of destinations has been set to 8 and 16. The average communication delay as a function of the average flit injection rate has been shown in Fig. 11. As observed from the results, the VBP and HP methods meet lower delay than the TBP and MBP approaches. However, the HP method outperforms the VBP approach due to the lower startup latency.

B. Unicast and Multicast (Mixed) Traffic Profile

In this set of simulation, we have employed a mixture of unicast and multicast traffic, where 80% of injected messages are unicast messages and the remaining 20% are multicast messages. This pattern may be representative of the traffic in a distributed shared-memory multiprocessor where updates and invalidations produce multicast messages and cache misses are served by unicast messages [12]. Hotspot traffic model profile [13] has been taken into account for unicast traffic generation. Under the hotspot traffic pattern, one or more nodes are chosen as hotspots receiving an extra portion of the traffic in addition to the regular uniform traffic. In the hotspot traffic model, given a hotspot percentage of h, a newly generated message is directed to each hotspot node with an additional h percent probability. We simulate hotspot traffic with a single hotspot node. The hotspot node is chosen to be node (2,2,2) in the 4×4×4 mesh network. Fig. 12 shows the performance with h = 10%. As the figure shows, the HP method outperforms the all other partitioning methods.

C. Hardware Overhead

To evaluate the area overhead of the proposed algorithms, the routers were synthesized with Synopsys D.C. using the TSMC 0.09µm standard cell library. In order to achieve better performance/power efficiency, the FIFOs were implemented using registers. All the schemes used the same routing unit implementation which is based on the Hamiltonian model, but their packet preparation mechanisms uses different number of registers. Comparing the area cost of the proposed methods with the TBP based scheme, indicates that the hardware overhead of implementing the MBP, VBP and HP methods are less than 0.5% and that can be considered negligible.

D. Power Dissipation

The power dissipation of the TBP, MBP, VBP and HP methods were calculated and compared under the multicast traffic model with 16 destinations using Synopsys PrimePower. The typical clock of 1 GHz is applied in 4×4×4 3D-mesh network. The results for the average power under multicast traffic are shown in Fig. 13. The average power values are computed near the saturation point, 0.17 (messages/cycle), under multicast traffic. The average power consumption of the HP scheme is 12.7%, 8.4%, and 4.2% less than that of TBP, MBP, and VBP schemes, respectively. In fact, this is achieved by smoothly balancing the traffic over the network using efficient balancing scheme which reduces the number of the hotspots and, hence, lowering the average power.

VI. Summary and Conclusion

This paper presented a novel idea of balanced partitioning which leads the network to be partitioned effectively. We first presented two partitioning methods in the realm of 3D-mesh NoCs, named Two-Block Partitioning (TBP) and Multi-Block Partitioning (MBP), to study the effect of balanced/unbalanced partitions. Afterward, the Vertical-Block partitioning (VBP) which partitions the network vertically was proposed. In order to minimize the startup latency, the Hybrid Partitioning (HP) is presented by the combination of the TBP and VBP methods. Experimental results show that the presented idea in the HP method reduces the transmission delay and provides a high degree of parallelism compared to the three mentioned methods.

Acknowledgment

The authors wish to acknowledge Nokia Foundation for the partial financial support during the course of this research.

References

[1] B.S. Stanley, P. Pande, "Networks-on-Chip in a Three-Dimensional Environment: A Performance Evaluation", IEEE transactions on computers, vol. 58, no. 1, pp.32–45, January 2009.

[2] H. Matsutani, M. Koibuchi, "Tighly-Coupled Multi-Layer Topologies for 3-D NoCs", International Conference on Parallel Processing ICCP, pp.75, September 2007.

[3] LIN X., NI L.M.: "Multicast communication in multicomputer networks", IEEE Trans. Parallel Distrib. Syst., 1993, 4, pp. 1105–1117

[4] Z. Liu, J.Duato, "Adaptive Unicast and Multicast in 3D Mesh Networks", Proc. of the Twenty-Seventh Hawaii International Conference, v.1, pp.173-182, 1994.

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

[6] X. Lin, and L. M. Ni, "Multicast Communication in Multicomputer Networks", IEEE Transactions on Parallel and Distributed Systems, pp. 1105-1117, 1993.

[7] N. E. Jerger, L. S. Peh, and M. H. Lipasti, " Virtual Circuit Tree Multicasting: A Case for On-Chip Hardware Multicast Support", In Proceedings of the International Symposium on Computer Architecture (ISCA), Beijing China, June 2008.

[8] L. Wang, H. Kim and E. J. Kim, "Recursive Partitioning Multicast: A Bandwidth-Efficient routing for Networks-On-Chip", International Symposium on Networks-on-Chip (NOCS), San Diego, CA, May 2009.

[9] M. Daneshtalab, M. Ebrahimi, S. Mohammadi, A. Afzali-Kusha, "Low distance path-based multicast algorithm in NOCs", IET (IEE) - Computers and Digital Techniques, Special issue on NoC, Vol. 3, Issue 5, pp. 430-442, Sep 2009.

[10] F. Li, C. Nicopoulos, T. Richardson, Y. Xie, V. Narayanan, and M. Kandemir, "Design and Management of 3D Chip Multiprocessors Using Network-in-Memory", ISCA-33, 2006.

[11] Z. Liu, J.Duato, "Adaptive Unicast and Multicast in 3D Mesh Networks", Proc. of the Twenty-Seventh Hawaii International Conference, v.1, pp.173-182, 1994.

[12] R. V. Boppana, S. Chalasani, C.S Raghavendra, "Resource deadlock and performance of wormhole multicast routing algorithms," IEEE Transactions on Parallel and Distributed Systems, pp. 535-549, 1998.

[13] G. Chiu, "The Odd-Even Turn Model for Adaptive Routing," IEEE Tran. On Parallel and Distributed System, pp 729-738, July 2000.

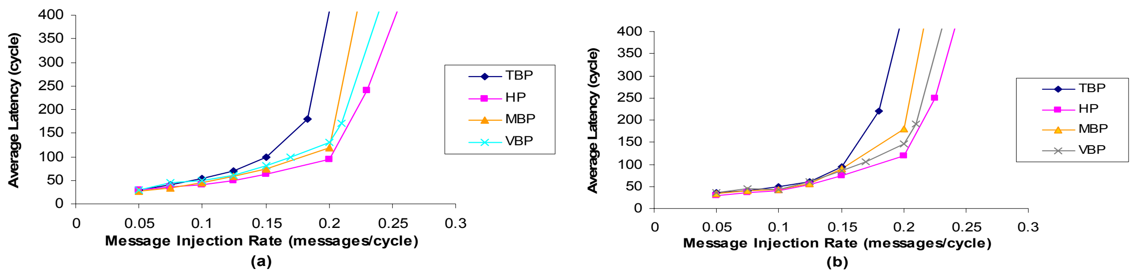


Fig. 11. Performance with different loads in 4×4×4 3D-mesh with (a) 8 destinations, (b) 16 destinations.

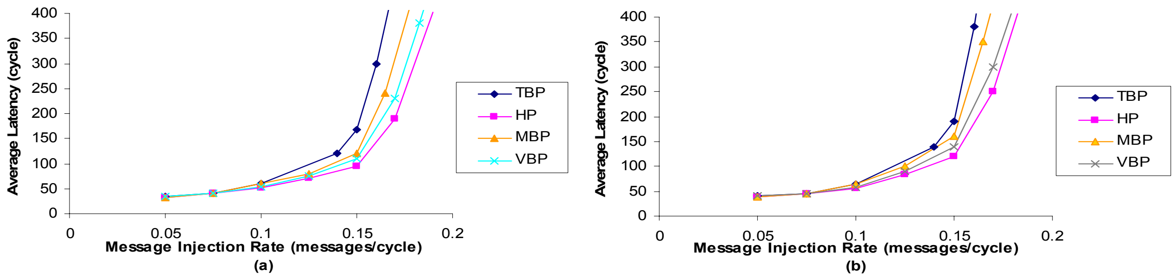


Fig. 12. Performance with different loads in 4×4×4 3D-mesh with (a) 8 destinations, (b) 16 destinations under mixed traffic (20% multicast and 80% unicast). Unicast traffic is based on the hotspot traffic model with a single hotspot node (2,2,2), and h=10%.

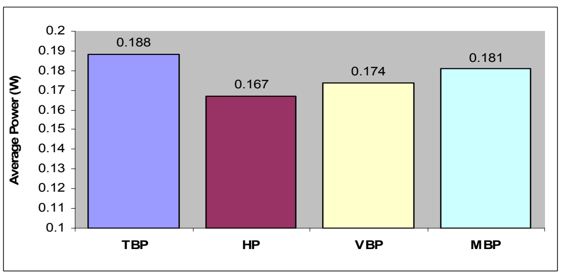


Fig. 13. Average power dissipation results in 4×4×4 3D-mesh under multicast traffic profile.