www.ietdl.org

Published in IET Computers & Digital Techniques
Received on 14th February 2013
Revised on 23rd July 2013
Accepted on 4th August 2013
doi: 10.1049/iet-cdt.2013.0034

Special Issue on Emerging On-Chip Networks and Architectures

ISSN 1751-8601

Fully adaptive routing algorithms and region-based approaches for two-dimensional and three-dimensional networks-on-chip

Masoumeh Ebrahimi

Laboratory of Embedded Computer and Electronic Systems (ECES), Department of Information Technology, University of Turku, Turku, Finland
E-mail: masebr@utu.fi

Abstract: Network congestion has negative impact on the performance of networks-on-chip (NoC). In traditional congestion-aware techniques, congestion is measured at a router level and delivered to other routers, either local or non-local. One of the contributions of this study is to show that performance can be improved if the congestion level is measured for a group of routers, called cluster, and propagated over the network, rather than considering the congestion level of a single router. The presented approach is discussed in both two-dimensional (2D) and three-dimensional (3D) mesh networks. To collect and propagate the congestion information of different clusters, a distributed approach is presented. The gathered information is utilised at routing units to deliver packets through the less congested regions. To distribute packets over the network without forming deadlock, routing algorithms should be carefully designed. The authors take advantage of fully adaptive routing algorithms, providing the maximum degree of adaptiveness for distributing packets. For 2D NoCs, a conventional fully adaptive routing algorithm, named dynamic XY (DyXY), is utilised. However, for 3D NoCs a fully adaptive routing algorithm is proposed and this method is called 3D-FAR. On top of each fully adaptive routing algorithm, a region-based approach is developed.

1 Introduction

System-on-chip (SoC) design is moving towards the integration of tens or hundreds of intellectual property (IP) blocks on a single chip. As chip integration grows, the on-chip communication becomes a performance bottleneck in high performance multi-processor systems-on-chip (MPSoCs). The regular tile-based network-on-chip (NoC) architecture has been proposed as a solution to meet the performance and design productivity requirements of the complex on-chip communication infrastructure [1]. A NoC provides an infrastructure for better modularity, scalability, fault-tolerant and higher bandwidth compared with traditional approaches [1]. It enables the integration of a large number of IP cores into a chip [1–3]. However, a planar chip fabrication technology is facing new challenges in the deep submicron regime [4, 5]. By the usage of global interconnects in two-dimensional (2D) designs, wire delay and power consumption increase significantly. To overcome these limitations, technology is moving rapidly towards the concept of three-dimensional integrated circuits (3D ICs), where multiple active silicon layers are vertically stacked. Three-dimensional technology overcomes the limited floorplanning choices of 2D designs and allows each layer to be instantiated with a different technology [6]. The major advantages of 3D NoCs are the considerable reduction on the average wire length and wire delay, resulting in lower power consumption and higher performance [4, 7–9].

Congestion occurs frequently in NoCs when the packets' demands exceed the capacity of network resources. Congestion may lead to the increased transmission delay, and thus limiting the performance of NoCs. Efficient routing algorithms can address this issue by routing packets through less congested areas and flattening traffic over the network. Routing algorithms are classified as deterministic and adaptive algorithms. The simplest deterministic routing method is dimension-order routing, which is known as XY or YX algorithm. Implementations of deterministic routing algorithms are simple but they are not able to balance the load across the links in a non-uniform or bursty traffic [3].

Adaptive routing has been used in interconnection networks to improve network performance and to tolerate link or router failures. In adaptive routing algorithms, a packet can traverse from a source to a destination through multiple paths. Specifically, adaptive routing algorithms can be used to avoid congestion by adapting the routing decision to the network status. Adaptive routing algorithms can be either partially adaptive or fully adaptive. In partially adaptive routing algorithms, packets are limited to choose among some shortest paths, whereas in fully adaptive methods, packets are allowed to take any minimal paths available between the source and destination pair [10]. Obviously, fully adaptive routing algorithms can distribute packets more efficiently over the network. We utilise the dynamic XY (DyXY) method [3] for 2D NoCs. One of the contributions of this paper is to propose a fully adaptive routing algorithm for 3D NoCs. This method requires two, two and four virtual channels along X, Y and Z dimensions, respectively, offering the minimum number of virtual channels for such algorithms.

In congestion-aware routing algorithms, the path a packet traverses from a source to a destination is determined by the network condition which can be based on local or global information. In approaches considering local traffic conditions, the routing decision is made only based on the congestion statuses of adjacent neighbours. These methods provide a limited view of the network condition; thereby they are not able to balance the traffic load in a non-uniform or bursty traffic. Globally adaptive routing algorithms mitigate this issue by considering the network status beyond neighbouring routers. In this paper, we consider non-local congestion information in the routing decision via collecting and distributing the information of a group of routers, called cluster. The process of gathering and transmitting traffic information is performed through a distributed propagation mechanism.

In this work, we consider 2D and 3D mesh topologies with wormhole switching technique. In a 2D mesh network, each router has five input/output ports whereas in a 3D mesh network, each router includes seven input/output ports, a natural extension from a 5-port 2D router by adding two ports to make connections to the upper and lower layers [12]. There are some other types of 3D routers such as the hybrid router [13] and MIRA [6]; however, since router efficiency is out of the concept of this paper, we have chosen a simple 7-port router.

The remainder of this paper is organised as follows: In Section 2, the related work is given. In Section 3, the fully adaptive routing algorithm in 3D NoCs is introduced. Sections 4 and 5 discuss the region-based NoC for 2D and 3D mesh networks. The results are reported in Section 6 while the summary and conclusion are given in the last section.

2 Related work

Several methods are presented in the realm of 2D NoCs in order to balance the traffic load over the network. We divide them into three main categories and investigate different methods within each category as follows:

2.1 Routing algorithms based on local congestion

Most of the congestion-aware algorithms consider local traffic condition; each router analyses the congestion conditions of its own and adjacent routers to choose an output channel. Routing decisions based on local congestion information may lead to an unbalanced distribution of traffic load.

DyXY [14] is a fully adaptive routing algorithm using one and two virtual channels along the X and Y dimensions, respectively. This algorithm considers local traffic condition in decision making in which each router compares the congestion condition in the instance input buffers of the neighbouring routers. This type of algorithms is efficient when the traffic is mostly local, that is, cores communicate with those close to them. In the neighbour-on-path (NoP) approach [15], the routing decision is performed based on the congestion information of the routers within two hops of the current router that are located in the minimal path to the destination. In fact, in NoP the locality decision is extended to 2-hop neighbours. NoP suffers from the recursive nature of the routing algorithm, resulting in a high hardware overhead and increased router complexity. Another routing algorithm based on local congestion is presented in [16]. This algorithm allows packets to be routed through more output channels at each router but the routing decision is made based on local congestion information.

2.2 Routing algorithms based on non-local congestion

The DBAR method [17] only considers the congestion value of the routers that are located in axes. These routers should be located in the shortest path between a source and destination. As the network size increases, the congestion information of faraway routers becomes unreliable and may result in wrong decisions.

A well-known method, named regional congestion awareness (RCA), is proposed in [18] to utilise non-local congestion information in the routing decision. In the RCA method, in order to provide global congestion information, the locally computed congestion value of a router is combined with those global signals propagated from the downstream routers and the newly aggregated value is transmitted to the upstream routers and so on. The main drawback of RCA is that the same congestion value may be used for the comparison purposes regardless of the destination position. It implies that the routing decision is affected by the congestion values of some routers which are resided outside of the minimal region from the source to the destination. DAR [11] has addressed the shortcoming of RCA by considering only the routers which are located between the source and the destination. However, the mechanism of distributing the information is more complex than RCA. Moreover, the congestion information is less frequently updated, which may result in routing decision based on un-updated data. Different realisations of cluster-based topology for 2D mesh networks are discussed in [19]. However, the emphasis of this paper is on the network and cluster sizes rather than a suitable routing algorithm. The presented method in [20] is another cluster-based approach with static short-cut channels to reduce the packets' latency. The complexity and overhead of the algorithm are the main weakness points of it. In CATRA [21], the passing probability of packets through the intermediate routers was calculated and based on this knowledge, clusters with the shape of trapezoids were formed. CATRA performs well in 2D mesh networks, but because of its specific structure, scalability is limited when moving to 3D networks.

2.3 Routing algorithms based on fuzzy logics and artificial intelligence algorithms

Different attempts were made to exploit fuzzy logic and machine learning approaches into NoCs. The presented fuzzy-based routing algorithm in [22] relies on local congestion information but the routing decision is more accurate and validated than DyXY. The idea behind these algorithms is to avoid rigid boundaries on the congestion values by employing the fuzzy logic mechanism. Fuzzy controllers compensate for ambiguities in a data by giving a level of confidence rather than declaring the data simply true or false.

DuQAR [23] and HARAQ [24] are two approaches based on Q-learning models. In these methods, the network condition is learned at run time and then this knowledge is utilised in the routing decision. In DuQAR, only the shortest paths can be used and each router maintains a Q-table to store the estimated latencies from the source to each destination router while in HARAQ, both minimal and non-minimal paths are utilised and Q-tables maintain the estimated latencies from the source to each destination region. As packets move within the network, Q-tables incorporate more global information.

In addition to routing algorithms which directly affect the performance of the network, there are some other attempts trying to collect the profiling information [25] or model the traffic dynamics in many-core systems [26]. These series of works help to optimise NoCs for more realistic models of network traffic. Some other works focus on deflection routings which naturally leads to a better load balance than wormhole routings [27, 28]. However, in deflection routings flits should be reordered at destinations to form the original packet.

2.4 Routing algorithms in 3D mesh networks

Although there are many congestion-aware routing algorithms presented in 2D NoCs, there are a few presented methods in 3D NoCs. MAR [29] is a partially adaptive routing algorithm in 3D networks based on the Hamiltonian path. It is a simple approach which provides adaptivity without using virtual channels. An extension of turn models from 2D to 3D network is done in 4N-First and 4P-First methods [30]. These algorithms are also partially adaptive routing algorithms and do not require any virtual channels. The planar-adaptive routing algorithm [31] is a well-known method presented in the realm of interconnection networks. This algorithm requires one, three and two virtual channels along the X, Y and Z dimensions, respectively. The adaptivity of this method is limited to a fully adaptive routing algorithm inside a sequence of 2D planes.

In this paper, we present a region-based routing algorithm on a 2D and 3D mesh networks. In these approaches, the network is partitioned into a group of clusters. The clusters are connected to each other via a light weight clustering network to distribute the congestion information. This network is built upon the mesh network where the data packets are propagated. The routing decision relies on the congestion level at the neighbouring clusters rather than local information. To implement these methods, a fully adaptive routing algorithm is needed in 2D and 3D NoCs. For this purpose, we use a traditional method, DyXY, as an underlying fully adaptive routing algorithm for 2D NoCs. However, due to the lack of fully adaptive method in a 3D NoCs, a novel algorithm is presented in this paper. This algorithm requires two, two and four virtual channels along the X, Y and Z dimensions, respectively.

3 Fully adaptive routing algorithms in 2D and 3D NoCs

There are several fully adaptive routing algorithms in 2D NoCs among them DyXY offers the minimum number of virtual channels which we select it in this manuscript. We propose a fully adaptive routing algorithm for 3D NoCs, named 3D-FAR.

3.1 DyXY routing algorithm in 2D NoCs

In the DyXY routing algorithm, if there are multiple shortest paths available between the current and destination router, a packet can always be forwarded to either X or Y dimension. Therefore, this routing algorithm needs a mechanism to guarantee deadlock freedom. For this purpose, the DyXY method utilises one and two virtual channels along the X and Y dimensions, respectively. In this method, the network is partitioned into increasing and decreasing subnetworks, as shown in Fig. 1*a*. The increasing subnetwork covers the $+X$ direction and the first virtual channel in the Y direction, whereas the decreasing subnetwork contains the rest of the channels ($-X$ direction and the second virtual channel in the Y direction). Therefore, if the destination router is to the east of the source, the packet will be routed through the increasing subnetwork. Similarly, if the destination router is to the left of the source, the packet will be routed through the decreasing subnetwork. In the case that the source and destination are in the same column, the packet can be routed using either subnetwork.

3.2 Fully adaptive routing algorithm in 3D NoCs (3D-FAR)

A 3D mesh network can be divided into eight subnetworks as: $((X^+)(Y^+)(Z^+)$, $(X^+)(Y^+)(Z^-)$, $(X^+)(Y^-)(Z^+)$, $(X^+)(Y^-)(Z^-)$,

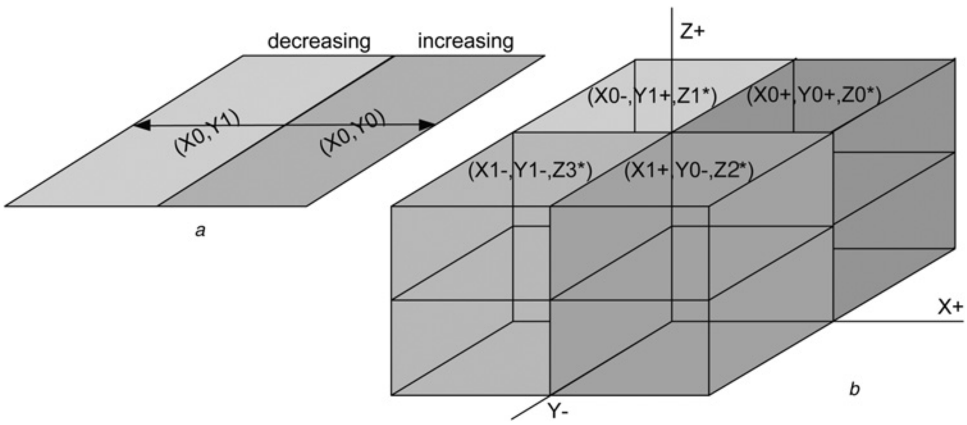

Fig. 1 *Virtual channel allocation*

a 2D NoC
b 3D NoC

Table 1 Assigning virtual channels to subnetworks

Subnetwork 1	Subnetwork 2	X dimension	Y dimension	Z dimension
$(X0^+)(Y0^+)(Z0^*)$	$(X0^-)(Y1^+)(Z1^*)$	Differs in vc direction	Differs in vc number	Differs in vc number
$(X0^+)(Y0^+)(Z0^*)$	$(X1^+)(Y0^-)(Z2^*)$	Differs in vc number	Differs in vc direction	Differs in vc number
$(X0^+)(Y0^+)(Z0^*)$	$(X1^-)(Y1^-)(Z3^*)$	Differs in vc number	Differs in vc number	Differs in vc number
$(X0^-)(Y1^+)(Z1^*)$	$(X1^+)(Y0^-)(Z2^*)$	Differs in vc number	Differs in vc number	Differs in vc number
$(X0^-)(Y1^+)(Z1^*)$	$(X1^-)(Y1^-)(Z3^*)$	Differs in vc number	Differs in vc direction	Differs in vc number
$(X1^+)(Y0^-)(Z2^*)$	$(X1^-)(Y1^-)(Z3^*)$	Differs in vc direction	Differs in vc number	Differs in vc number

$(X^-)(Y^+)(Z^+)$, $(X^-)(Y^+)(Z^-)$, $(X^-)(Y^-)(Z^+)$ and $(X^-)(Y^-)(Z^-)$). A simple way to implement a fully adaptive routing algorithm is to assign a separate set of virtual channels to each subnetwork such as ($(X1^+)(Y1^+)(Z1^+)$, $(X2^+)(Y2^+)(Z1^-)$, $(X3^+)(Y1^-)(Z2^+)$, $(X4^+)(Y2^-)(Z2^-)$, $(X1^-)(Y3^+)(Z3^+)$, $(X2^-)(Y4^+)(Z3^-)$, $(X3^-)(Y3^-)(Z4^+)$ and $(X4^-)(Y4^-)(Z4^-)$), where the numbers indicate the virtual channel on each dimension. With this channel assignment, four virtual channels are needed along each dimension and the network is deadlock-free as all subnetworks are disjoint from each other.

However, the number of virtual channels can be reduced to two virtual channels along one of dimensions, this method is called dynamic XYZ (DyXYZ) [8]. In this way, the whole network can be split into two main subnetworks, each having four subnetworks. Dividing the network into two parts can be done along one of dimensions, reducing the number of virtual channels from four to two for that dimension. Since two main subnetworks are separated, the network remains deadlock-free. On the other hand, the subnetworks within each main subnetwork use different sets of virtual channels on the two remaining dimensions. Therefore, the eight subnetworks are disjoint and the network is deadlock-free.

Using 3D-FAR, the number of virtual channels reduces to two, two and four virtual channels along the X, Y and Z dimensions. In the 3D-FAR algorithm, the network is partitioned into four subnetworks (Fig. 1*b*) as ($(X^+)(Y^+)(Z^*)$, $(X^-)(Y^+)(Z^*)$, $(X^+)(Y^-)(Z^*)$ and $(X^-)(Y^-)(Z^*)$), where '+', '−' represent the channels along the positive and negative directions, respectively, and '*' stands for both positive and negative directions (i.e. a bidirectional channel). The deadlock freeness can be proved by dividing the network into virtual networks and guaranteeing that each of them uses a separate set of virtual channels. The virtual channel assignment of 3D-FAR is as follows: ($(X0^+)(Y0^+)(Z0^*)$, $(X0^-)(Y1^+)(Z1^*)$, $(X1^+)(Y0^-)(Z2^*)$ and $(X1^-)(Y1^-)(Z3^*)$) where the numbers indicate the virtual channel numbers assigned to a direction.

Theorem 1: The network is deadlock-free within each subnetwork.

A cycle can be completed if packets are able to take both positive and negative directions along at least two dimensions. For example, to form a cycle in a XY plane, it is necessary to take X^+, X^-, Y^+ and Y^- directions. In the subnetwork 1, (X0)(Y0)(Z0) packets cannot take the negative direction of the X and Y dimensions and only the pair along the Z dimension (Z^-, Z^+) is completed. However, a cycle cannot be formed without taking all directions of two complete pairs, and thus there is no possibility of deadlock in this subnetwork. Similarly in each of the subnetwork 2, subnetwork 3 and subnetwork 4, only the pair along the Z dimension is completed and thereby a cycle cannot be formed within each subnetwork.

Theorem 2: The network is deadlock-free between subnetworks.

To prove that a network is deadlock-free between subnetworks, it is enough to show that different subnetworks are disjoint from each other. By a pairwise comparison among each two subnetworks, it can be easily obtained that either the direction or the virtual channel number differs along each dimension. For example, subnetworks $(X0^+)(Y0^+)(Z0^*)$) and ($(X0^-)(Y1^+)(Z1^*)$ are disjoint along the X dimension, because of using the positive direction in one and negative direction in the other. The two subnetworks are also disjoint along the Y and Z dimensions, since the virtual channel number 0 is employed in the first one and the virtual channel number 1 is used in the other one. This comparison can be done for any other pair of subnetworks (Table 1). Considering the Theorems 1 and 2, we prove that the network is deadlock-free within each subnetwork and on the other hand, different subnetworks are disjoint from each other, therefore the whole network is deadlock-free.

4 Region-based congestion-aware approach in 2D mesh networks (2D-RA)

4.1 Distributing congestion information

Fig. 2*a* shows the region-based NoC structure in a 2D network. The network is divided into several overlapped clusters in which a cluster contains four routers and a cluster. The design consists of two separate mesh networks: data network and lightweight cluster network. The data network connects the routers to each other to propagate packets over the network; whereas in the cluster network, clusters communicate with each other to spread the congestion information. Each cluster performs three simple tasks. First, it collects and aggregates the congestion information from the local routers; second, it distributes the information to the neighbouring clusters; third, the cluster transfers the information to the local routers (this information includes both local cluster information and those received from the neighbouring clusters). In addition to this information, a router receives 1-bit information from the neighbouring routers about the buffer availability at the corresponding input port. Accordingly, each router is aware of the congestion condition of the routers located in the local and neighbouring clusters and the input buffer availability at the neighbouring routers.

In the DyXY method, each router has five input ports (i.e. L, E, W, N and S). By adding two virtual channels along the Y direction, each router will have seven input buffers (i.e. L, E, W, N1, N2, S1 and S2). As shown in Fig. 2*c*, the congestion level (CL) of a router is computed using the congestion status (CS) of input buffers. In each flit event

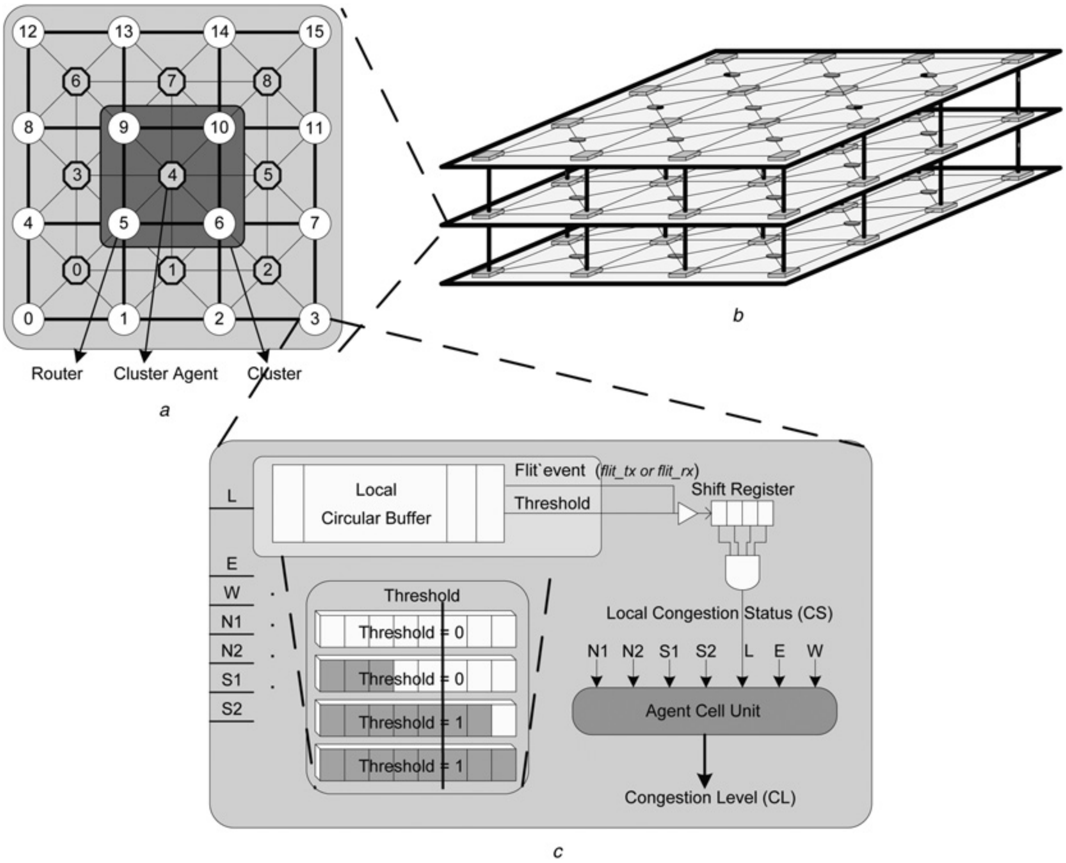

Fig. 2 *Region-based NoC and computing the congestion status of an input buffer and the congestion level of a router*

a 2D region-based NoC
b 3D region-based NoC
c Congestion measurement mechanism

(i.e. flit_tx or flit_rx), if the number of occupied cells of an input buffer is larger (smaller) than a threshold value, the threshold signal is assigned to one, otherwise zero. A history-based scheme is used to capture the threshold signal of an input buffer. For this purpose a 4-bit shift register is adopted to store the threshold signal whenever a new flit enters or leaves the buffer. The CS signal is asserted if all bits of the shift register are one. Finally, the CL value of a router is computed by summing up the CS signals received from the input buffers. In fact, the CL value of each router indicates its load level; for example, if the east and local input buffers of a router are congested (CS(local) = 1 and CS(east) = 1), then the CL value of the router will be two. By considering seven input buffers, the maximum value of CL can be accommodated in 3 bits.

To explain the propagation strategy, let us consider the central cluster (i.e. cluster 4) in Fig. 2*a*. The cluster 4 receives the 3-bit CL value from each of the four local routers (i.e. routers 5, 6, 9 and 10). The obtained 12-bit information should be transferred to the local routers and neighbouring clusters, but it does not require transferring the whole information to all of them. For example, a local router knows about its CL value and only needs to be informed about the CL values of the other three local routers. In another example, when transferring the information from the cluster 4 to the cluster 5, only the CL value of routers 5 and 9 are delivered to the cluster 5 since this cluster receives the CL value of routers 6 and 10 by its local connections. A similar perspective is applied to transfer the information from the cluster 4 to clusters 7, 3 and 1.

4.2 Congestion-aware selection method

By distributing the congestion information over the network, the routing decision can be assisted by the local and non-local congestion information received from the neighbouring clusters. For example, the router 0 in Fig. 2*a* not only knows the CL value of the routers within its cluster (i.e. routers 1, 4 and 5) but also have the information about the routers in the clusters C1 and C3 (i.e. routers 2, 6, 8 and 9).

The routing decision in the congestion-aware selection method is made based on the relative position of the source and destination routers. The following rules are valid for both 2D and 3D NoCs:

Rule 1: If the source and destination are located in the same row or column, the packet has no adaptivity and it is delivered through that direction. For instance, in Fig. 3*a*, when the packet is currently at the router 0 and the destination is located at the routers 1, 2 or 3, the packet has to be routed along the X direction while the packet is sent to the Y direction when the destination is located at the routers 4, 8 or 12.

Rule 2: If the distance along both directions is one, the input buffer availabilities at the neighbouring routers are compared together and the packet is sent to the less congested direction. For example, in Fig. 3*b*, when the destination is at the router 5, the buffer availability at the west input buffer of the router 1

doi: 10.1049/iet-cdt.2013.0034

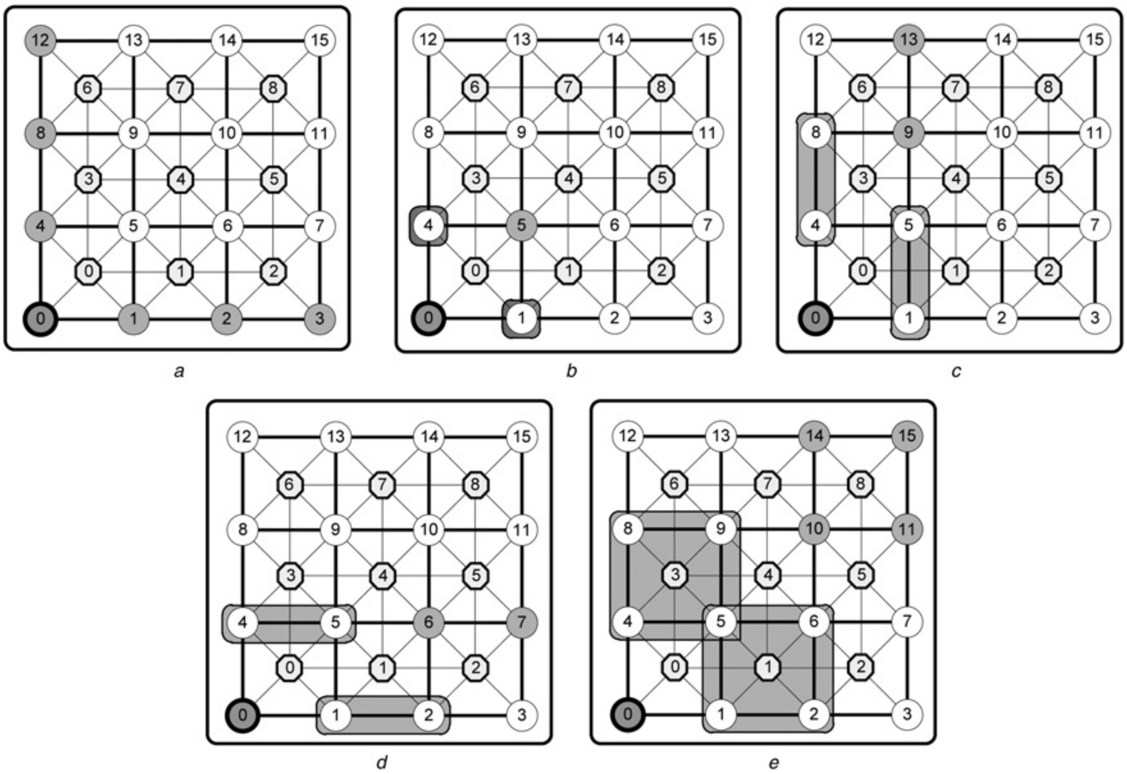

Fig. 3 *The Congestion-aware selection method in 2D mesh network*

a Rule 1 is applied (single choice)
b Rule 2 is applied (buffer availability)
c Rule 3 is applied (buffer availability, pair)
d Rule 3 is applied (buffer availability, pair)
e Rule 4 is applied (buffer availability, cluster)

is compared with the south (virtual channel1) input buffer of the router 4.

Rule 3: If the distance along a dimension is one and along the other dimension is greater than one, first the buffer availabilities along both directions are compared together. In the case that the values are the same, the CL value of one and two hop neighbours (pair) is taken into consideration. For example, when the destination is located at the router 9 or 13 in Fig. 3*c*, first the buffer availabilities at routers 1 and 4 are compared together. If they are the same, the congestion value of pair (1, 5) is compared with pair (4, 8). The congestion of these pairs is called pair-congestion. Similarly, Rule3 is used in Fig. 3*d*.

Rule 4: If the distances along both directions are equal or greater than two hops, after checking the buffer availabilities at the input buffers of the neighbouring routers, the congestion levels of the neighbouring clusters are compared together. The congestion on these clusters is called cluster-congestion. For instance, in Fig. 3*e*, first the input buffers of the routers 1 and 4 are compared together, and then the congestion values of clusters 1 and 3 are checked.

5 Region-based congestion-aware approach in 3D mesh networks (3D-RA)

5.1 Distributing congestion information

The congestion propagation mechanism in each layer is the same as in the 2D mesh network. Each router delivers the congestion level of its direct and indirect neighbouring routers plus its own congestion value (i.e. ten routers in total). Similar to the 2D mesh network, the buffer availability at the current router is also transferred to upper and lower layers.

Using the propagation mechanism in a 2D mesh network, the router 5 in Fig. 2 has the congestion level of all routers in its local clusters (i.e. clusters 0, 1, 3 and 4) and neighbouring clusters (i.e. routers 2, 6 and 7). Among them, the router 5 transfers the congestion information of clusters 0, 1, 3 and 4 to the upper and lower layers. In addition, it delivers the buffer availability of the up and down input buffers into the upper and lower layers, respectively.

5.2 Congestion-aware selection method

An example of the congestion-aware selection method is shown in Fig. 4. In this example, a packet is sent from the source router 0 to every other router in the network. We investigate the similar rules as in the 2D network in the 3D network which can be explained as follows. In Fig. 4*a*, according to Rule 1, the east, north, or up direction are the possible choices to forward the packet when the destination is located along the X, Y or Z dimension. Using Rule 2 (Fig. 4*b*), when the destination is not located in the axes and the distances along the X and Y dimensions are less than or equal to one hop, the buffer availabilities are checked. Figs. 4*c* and *d* cover the Rule 3 in which the buffer availabilities are compared together before the pair-congestion values. Finally, if the destination is located two hops away along the X and Y dimensions, the cluster-congestion are evaluated after comparing the buffer availabilities.

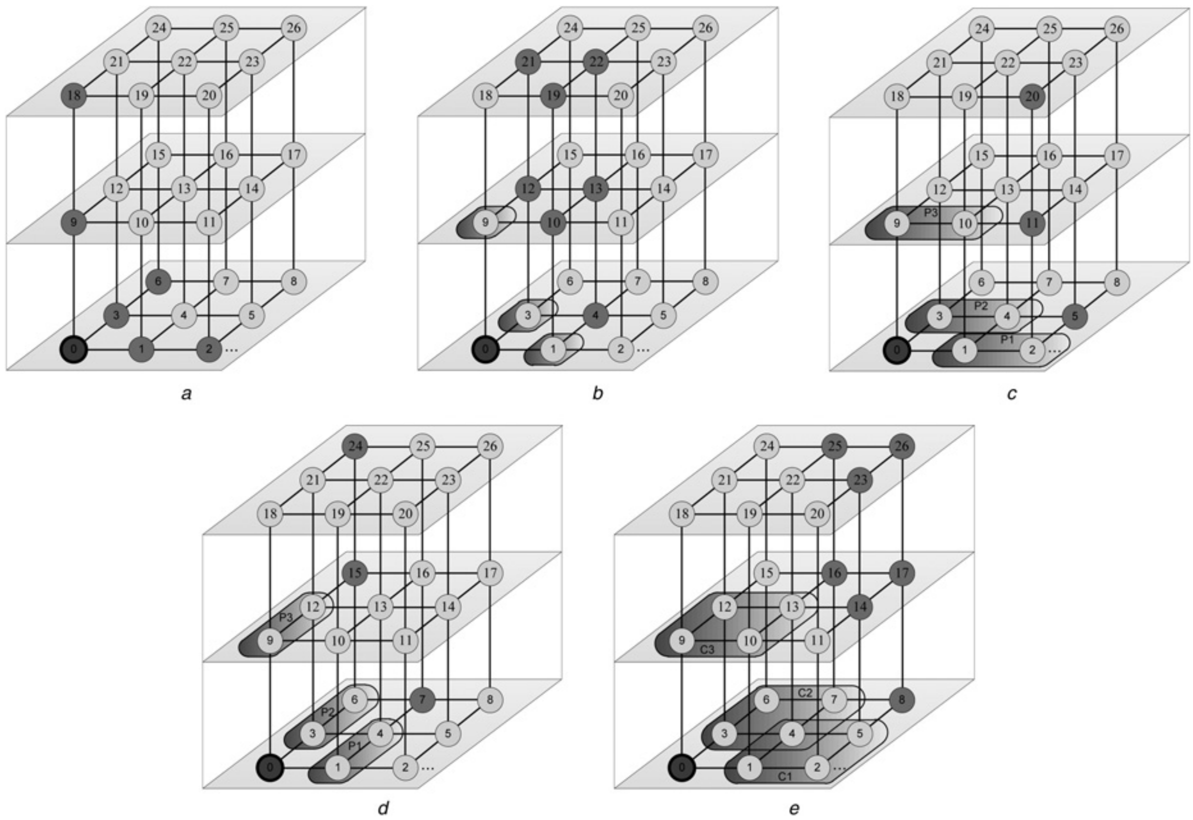

Fig. 4 *The Congestion-aware selection method in 3D mesh network*

a Rule 1 is applied (single choice)
b Rule 2 is applied (buffer availability)
c Rule 3 is applied (buffer availability, pair)
d Rule 3 is applied (buffer availability, pair)
e Rule 4 is applied (buffer availability, cluster)

6 Results and discussion

2D and 3D NoC simulators are implemented with VHDL to model all major components. Simulations are carried out to determine the latency characteristics of each network. As a performance metric, we use latency defined as the number of cycles between the initiation of a message issued by a processing element (PE) and the time when the message is completely delivered to the destination PE. For all routers, the data width and the frequency is set to 32 bits and 1 GHz, respectively, which leads to a bandwidth of 32 Gb/s. The packet size is randomly selected between 3 and 8 flits. Each input virtual channel has a buffer (first-input first-output) with the size of 6 flits. The congestion threshold value is set to 4, meaning that the congestion condition is considered when 4 out of 6 buffer slots are occupied.

6.1 Analysis of 2D mesh networks

To evaluate the efficiency of the proposed 2D-RA, three other on-chip networks are also implemented as DyXY [14], NoP [15] and DBAR [17]. All algorithms are implemented using one and two virtual channels along the X and Y dimensions. Two synthetic traffic profiles including uniform

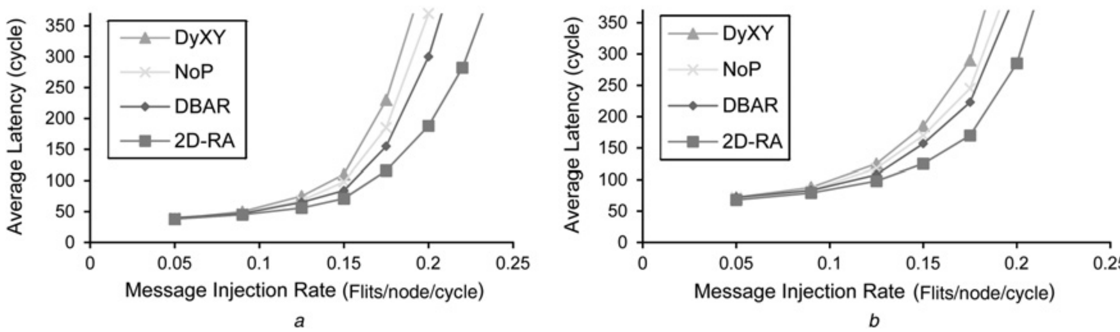

Fig. 5 *Performance analysis under uniform traffic model*

a 8 × 8 2D mesh
b 14 × 14 2D mesh

IET Comput. Digit. Tech., 2013, Vol. 7, Iss. 6, pp. 264–273
doi: 10.1049/iet-cdt.2013.0034

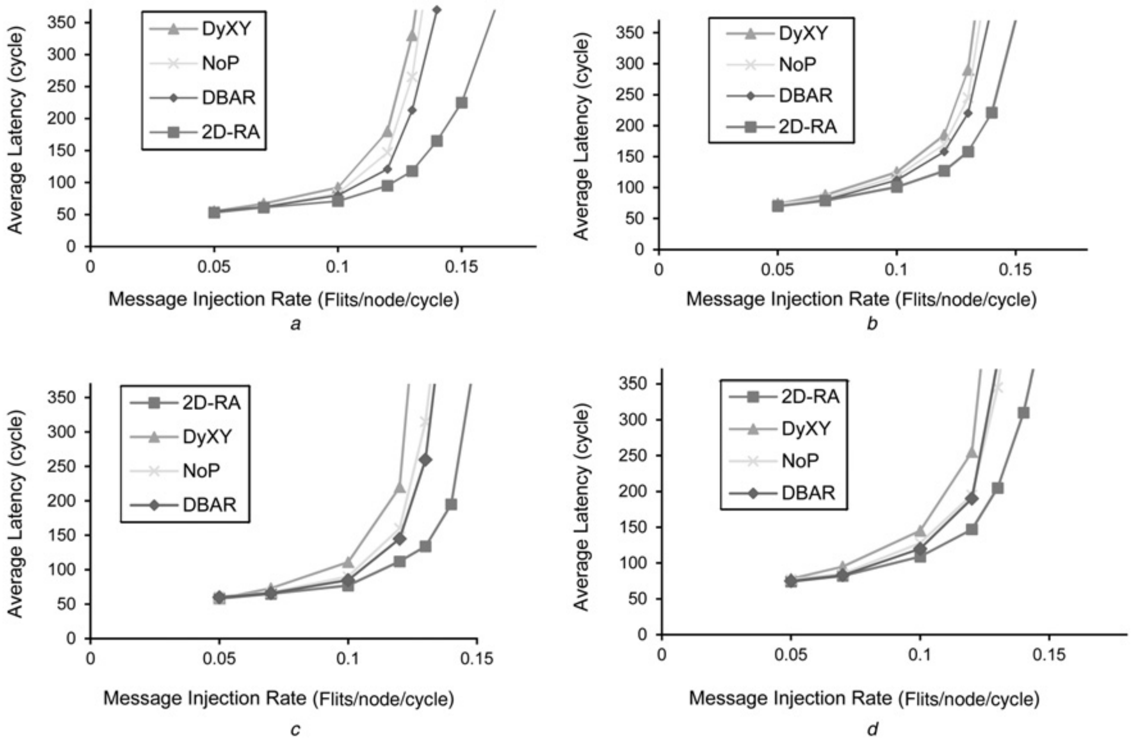

Fig. 6 *Performance analysis under uniform hotspot traffic model*
a 8 × 8 2D mesh with single hotspot router at (4, 4)
b 14 × 14 2D mesh with single hotspot router at (7, 7)
c 8 × 8 2D mesh with fat hotspot routers at (3,3) (4,3) (3,4) and (4,4)

random and hotspot, along with SPLASH-2 and PARSEC application traces are used.

6.1.1 Performance analysis under uniform traffic profile: In the uniform traffic profile, each PE generates data packets and sends them to another PE using a uniform distribution. The mesh sizes are considered to be 8 × 8 and 14 × 14. In Fig. 5, the average communication delay as a function of the average packet injection rate is plotted for both mesh sizes. As observed from the results, 2D-RA leads to the lowest latency. This was expected due to the distribution of traffic over less congested areas. Using 2D-RA, each router can observe the congestion information of not only the neighbouring routers, but also the routers residing beyond the neighbouring routers.

Table 2 System configuration parameters

Processor configuration	
Instruction set architecture	SPARC
Number of processors	36
Issue width	1
Cache configuration	
L1 cache	Private, split instruction and data cache, each cache is 16 KB. 4-way associative, 64-bit line, 3-cycle access time
L2 cache	Shared, distributed in 3 layers, unified 48 MB (48 banks, each 1 MB). 64-bit line, 6-cycle access time
Cache coherence protocol	MESI
Cache hierarchy	SNUCA
Memory configuration	
Size	4 GB DRAM
Access latency	260 cycles
Requests per processor	16 outstanding

6.1.2 Performance analysis under hotspot traffic profile: Under the hotspot traffic pattern, one or more routers are chosen as hotspots receiving an extra portion of the traffic in addition to the regular uniform traffic. In simulations, given a hotspot percentage of H, a newly generated message is directed to each hotspot router with an additional H percent probability. First, we simulate the hotspot traffic with a single hotspot router at (4, 4) and (7, 7) in the 8 × 8 and 14 × 14 2D mesh networks, respectively. Figs. 6*a* and *b* show the latency curves in 8 × 8 and 14 × 14 2D mesh networks when $H = 10\%$. As observed from the figures, 2D-RA achieves better performance compared to the other schemes. The performance improvement in 8 × 8 mesh network is more distinguishable than 14 × 14 network. The reason might be that the size of cluster regions is small for bigger network sizes. Second, we measure the packet latencies when there are four hotspot routers in the network. These hotspot routers are considered to be the nodes at (3,

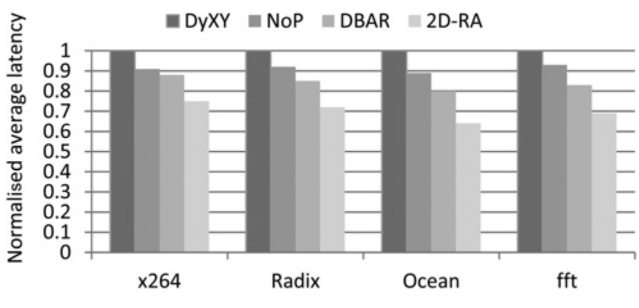

Fig. 7 *Performance under different application benchmarks*

Table 3 Hardware implementation details

Networks	Area, mm^2	Power, W
DyXY	6.670	2.34
NoP	6.843	2.61
DBAR	6.731	2.31
2D-RA	6.862	2.47

Table 4 Power dissipation

Network platforms	Power (W) dynamic and static
XYZ	1.94
DyXYZ	1.61
3D-RA	1.74

3), (4, 3), (3, 4) and (4, 4) in an 8×8 mesh network and the nodes (7, 7), (8, 7), (7, 8) and (8, 8) in a 14×14 mesh network. Figs. 6*c* and *d* show the latency curves in 8×8 and 14×14 2D mesh networks with four hotspot routers and $H = 10\%$.

6.1.3 Performance analysis under application traffic profile: To know the real impact of the proposed method, we used traces from some application benchmark suites selected from SPLASH-2 [32] and PARSEC [33]. Traces are generated from SPLASH and PARSEC using the GEMS simulator [34]. We used the x264 application of PARSEC and the Radix, Ocean and fast Fourier transform applications from SPALSH-2 in our simulation. Table 2 summarises our full system configuration.

Fig. 7 shows the average packet latency across four benchmark traces, normalised to DyXY. Although 2D-RA provides lower latency than other schemes, it shows the greatest performance gain on Ocean with 36% reduction in latency. The average performance gain of 2D-RA across all benchmarks is up to 30% against DyXY, 23% against NoP and 16% against DBAR.

6.1.4 Physical analysis: To assess the area overhead and power consumption of the proposed on-chip network, the whole platform of each network is synthesised by Synopsys Design Compiler. Each network includes network interfaces, routers, clusters (for 2D-RA) and communication channels. For synthesis, we use the UMC 90 nm technology at the operating frequency of 1 GHz and supply voltage of 1 V. We perform place-and-route, using Cadence Encounter, to have precise power and area estimations. The power dissipation of each scheme is calculated under the x264 benchmark using Synopsys PrimePower in a 6×6 2D mesh. The layout area and power consumption of each platform are shown in Table 3.

The area overhead of the 2D-RA is slightly higher than the other approaches. For example, the 2D-RA platform imposes only 2% hardware overhead and 5% higher power consumption compared to the simplest method DyXY.

6.2 Analysis of 3D mesh network

In this section, we compare 3D-RA with XYZ and DyXYZ in terms of latency and power consumption. For consistency, four, four, and two virtual channels are used for all three methods. Extra virtual channels are utilized to improve the performance so that packets can adaptively select between virtual channels based on the congestion values, whenever possible.

6.2.1 Performance analysis under uniform traffic profile: In the uniform traffic profile, each PE generates data packets and sends them to another PE using a uniform distribution. In Fig. 8*a*, the average communication delay as a function of the average packet injection rate is plotted for all schemes. As observed from the results, the XYZ routing algorithm leads to considerably lower latency than the DyXYZ and 3D-RA methods. The reason is that dimension-order routings are best suited on uniform traffic which results in evenly distributing traffic over the network. Among 3D-RA and DyXYZ, 3D-RA performs better.

6.2.2 Performance analysis under hotspot traffic profile: We simulate the hotspot traffic with four hotspot routers at positions (2, 1) and (3, 1) in layer 2 and the same positions in layer 3 in the $4 \times 4 \times 4$ mesh network. The performance of each network with $H = 10\%$ is illustrated in Fig. 8*b*. As observed from the figure, 3D-RA leads to the best performance since it considers the congestion condition of the neighbouring regions in the routing decision. In addition, DyXYZ performs better than simple XYZ routing algorithm. In fact, the improvement is achieved by smoothly balancing the traffic over the network which reduces the number of the hotspots and, hence, improving the performance than the XYZ routing algorithm. It is worth mentioning that DyXYZ can adaptively selects between output directions based on the congestion condition in the input buffer of the neighbouring routers.

6.2.3 Physical analysis: To assess the area overhead and power consumption of the presented communication architectures, the whole platforms, including network

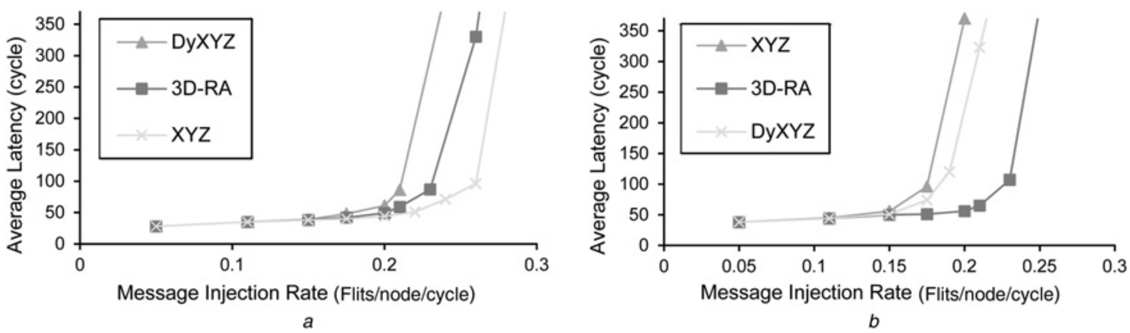

Fig. 8 *Performance in $4 \times 4 \times 4$ mesh network*

a Under uniform model
b Under hotspot traffic model with $H = 10\%$

IET Comput. Digit. Tech., 2013, Vol. 7, Iss. 6, pp. 264–273
doi: 10.1049/iet-cdt.2013.0034

interfaces and routers, are synthesized using Synopsys Design Compiler with UMC 90 nm technology, while the backend is performed with the Cadence Encounter tool. Since all the algorithms (i.e. the XYZ, DyXYZ and 3D-RA routing algorithms) use the same number of virtual channels, the total area cost of each layer for all platforms is almost similar and about 18 mm^2. The power dissipation of the 3D-RA, XYZ and DyXYZ methods were calculated and compared under the hotspot traffic model. The results for the average power under hotspot traffic are shown in Table 4. The average power values are computed near the saturation point.

7 Summary and conclusion

Congestion can be avoided by balancing traffic load over the network. This is possible by knowing about the traffic conditions in different regions of the network and routing packets through the less-congested parts. Traditional methods focused on measuring the traffic at individual routers than a region. One of the contributions of this work is to show that the performance can be improved if the routing decisions are made based on regional traffic condition.

We discussed the region-based approach in both 2D and 3D mesh networks and utilised fully adaptive routing algorithms in both networks. Therefore, pockets can be routed using all the shortest paths without any routing restriction. We use a conventional algorithm named DyXY for 2D NoCs, while we presented a fully adaptive routing algorithm for 3D NoCs. This algorithm requires the minimum number of virtual channels to provide fully adaptiveness which is two, two and four virtual channels along the X, Y and Z dimensions.

8 References

1 Wu, D., Al-Hashimi, B.M., Schmitz, M.T.: 'Improving routing efficiency for network-on-chip through contention-aware input selection'. Proc. Asia and South Pacific Conf. Design Automation, 2006, pp. 36–41

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

3 Jantsch, A., Tenhunen, H.: 'Networks on chip' (Springer, 2003)

4 Ebrahimi, M., Daneshtalab, M., Liljeberg, P., Plosila, J., Tenhunen, H.: 'Cluster-based Topologies for 3D Networks-on-Chip Using Advanced Interlayer Bus Architecture'. *Journal of Computer and System Sciences*, 2012, **79**, (4), pp. 475–491

5 Banerjee, K., Souri, S.J., Kapur, P., Saraswat, K.C.: '3-D ICs: a novel chip design for improving deep-submicrometer interconnect performance and systems-on-chip integration', *Proc. IEEE*, 2001, **89**, (5), pp. 602–633

6 Murali, S., Seiculescu, C., Benini, L., De Micheli, G.: 'Synthesis of networks on chips for 3D systems on chips'. Proc. Design Automation Conf. (ASP-DAC), 2009, pp. 242–247

7 Matsutani, H., Koibuchi, M., Amano, H.: 'Tightly-coupled multi-layer topologies for 3-D NoCs'. Proc. 41st Int. Conf. Parallel Processing, 2007

8 Ebrahimi, M., Chang, X., Daneshtalab, M., Plosila, J., Liljeberg, P., Tenhunen, H.: 'DyXYZ: Fully Adaptive Routing Algorithm for 3D NoCs'. Proc. 21th IEEE Euromicro Conf. Parallel, Distributed and Network-Based Computing (PDP), 2013, pp. 499–503

9 Pavlidis, V.F., Friedman, E.G.: '3-D topologies for networks-on-chip', *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, 2007, **15**, (10), pp. 1081–1090

10 Dally, W., Towles, B.: 'Principles and practices of interconnection networks' (Morgan Kaufmann Publishers, 2003)

11 Ramanujam, R.S., Lin, B.: 'Destination-based adaptive routing on 2D mesh networks'. Proc. ACM/IEEE Symp. Architectures for Networking and Communications Sys. (ANCS), 2010, pp. 1–12

12 Ebrahimi, M., Daneshtalab, M., Plosila, J.: 'Fault-Tolerant Routing Algorithm for 3D NoC Using Hamiltonian Path Strategy'. Proc. 16th ACM/IEEE Design, Automation, and Test in Europe (DATE), 2013, pp. 1601–1604

13 Weldezion, A.Y., Grange, M., Pamunuwa, D., *et al.*: 'Scalability of network-on-chip communication architecture for 3-D meshes'. Proc. Third ACM/IEEE Int. Symp. Networks-on-Chip (NOCS), 2009, pp. 114–123

14 Li, M., Zeng, Q.-A., Jone, W.-B.: 'DyXY – a proximity congestion-aware deadlock-free dynamic routing method for network on chip'. Proc. 43rd ACM/IEEE Design Automation Conf., 2006, pp. 849–852

15 Ascia, G., Catania, V., Palesi, M., Patti, D.: 'Implementation and analysis of a new selection strategy for adaptive routing in networks-on-chip', *IEEE Trans. Comput.*, 2008, **57**, (6), pp. 809–820

16 Wang, C., Hu, W.-H., Bagherzadeh, N.: 'Congestion-aware network-on-chip router architecture'. 2010 15th CSI Int. Symp. Computer Architecture and Digital Systems (CADS), 2010, pp. 137–144

17 Ma, S., Enright Jerger, N., Wang, Z.: 'DBAR: an efficient routing algorithm to support multiple concurrent applications in networks-on-chip'. Proc. 38th Annual Int. Symp. Computer Architecture, 2011, pp. 413–424

18 Gratz, P., Grot, B., Keckler, S.W.: 'Regional congestion awareness for load balance in networks-on-chip'. Proc. IEEE 14th Int. Symp. High Performance Computer Architecture, HPCA, 2008, pp. 203–214

19 Winter, M., Prusseit, S., Fettweis, G.P.: 'Hierarchical Routing Architectures in Clustered 2D-Mesh Networks-on-Chip'. Proc. of SoC Design Conf. (ISOCC), 2010, pp. 388–391

20 Qian, Z., Bogdan, P., Wei, G., Tsui, C.-Y., Marculescu, R.: 'A traffic-aware adaptive routing algorithm on a highly reconfigurable network-on-chip architecture'. Proc. Eighth IEEE/ACM/IFIP Int. Conf. Hardware/Software Codesign and System Synthesis, 2012, pp. 161–170

21 Ebrahimi, M., Daneshtalab, M., Liljeberg, P., Plosila, J., Tenhunen, H.: 'CATRA- congestion aware trapezoid-based routing algorithm for on-chip networks'. Proc. Design, Automation Test in Europe Conference Exhibition (DATE), 2012, pp. 320–325

22 Dehyadegari, M., Daneshtalab, M., Ebrahimi, M., Plosila, J., Mohammadi, S.: 'An adaptive fuzzy logic-based routing algorithm for networks-on-chip'. Proc. NASA/ESA Conf. Adaptive Hardware and Systems (AHS), 2011, pp. 208–214

23 Farahnakian, F., Ebrahimi, M., Daneshtalab, M., Liljeberg, P., Plosila, J.: 'Adaptive reinforcement learning method for networks-on-chip'. Proc. SAMOSXIII, 2012, pp. 236–243

24 Ebrahimi, M., Daneshtalab, M., Farahnakian, F., *et al.*: 'HARAQ: congestion-aware learning model for highly adaptive routing algorithm in on-chip networks'. Proc. Int. Symp. Networks-on-Chip, 2012, pp. 19–26

25 Bahrebar, P., Florea, R.-M., Heirman, W., Denis, L., Munteanu, A., Stroobandt, D.: 'Making communication a first-class citizen in multicore partitioning'. 2013 21st Euromicro Int. Conf. Parallel, Distributed and Network-Based Processing (PDP), Proc., pp. 287–293

26 Bogdan, P., Marculescu, R.: 'Non-stationary traffic analysis and its implications on multicore platform design', *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 2011, **30**, (4), pp. 508–519

27 Jose, J., Nayak, B., Kumar, K., Mutyam, M.: 'DeBAR: deflection based adaptive router with minimal buffering'. Proc. Conf. Design, Automation and Test in Europe, 2013, pp. 1583–1588

28 Lu, Z., Zhong, M., Jantsch, A.: 'Evaluation of on-chip networks using deflection routing'. Proc. 16th ACM Great Lakes Symposium on VLSI, 2006, pp. 296–301

29 Ebrahimi, M., Daneshtalab, M., Liljeberg, P., Plosila, J., Flich, J., Tenhunen, H.: 'Path-based partitioning methods for 3D networks-on-chip with minimal adaptive routing', *IEEE Trans. Comput.*, 2012

30 Pasricha, S., Zou, Y.: 'A low overhead fault tolerant routing scheme for 3D networks-on-chip'. 2011 12th Int. Symp. Quality Electronic Design (ISQED), 2011, pp. 1–8

31 Chien, A.A., Kim, J.H.: 'Planar-adaptive routing: low-cost adaptive networks for multiprocessors'. Proc. 19th Annual Int. Symp. Computer Architecture, 1992, pp. 268–277

32 Woo, S.C., Ohara, M., Torrie, E., Singh, J.P., Gupta, A.: 'The SPLASH-2 programs: characterization and methodological considerations'. Proc. 22nd Annual Int. Symp. Computer Architecture, 1995, pp. 24–36

33 Bienia, C., Kumar, S., Singh, J.P., Li, K.: 'The PARSEC benchmark suite: characterization and architectural implications'. Proc. 17th Int. Conf. Parallel Architectures and Compilation Techniques, 2008, pp. 72–81

34 Martin, M.M.K., Sorin, D.J., Beckmann, B.M., *et al.*: 'Multifacet's general execution-driven multiprocessor simulator (GEMS) toolset', *SIGARCH Comput. Architect. News*, 2005, **33**, (4), pp. 92–99