Published in IET Computers & Digital Techniques
Received on 6th July 2008
Revised on 11th April 2009
doi: 10.1049/iet-cdt.2008.0086

In Special Issue on Networks on Chip

ISSN 1751-8601

Low-distance path-based multicast routing algorithm for network-on-chips

M. Daneshtalab[1,3] *M. Ebrahimi*[2,3] *S. Mohammadi*[1]
A. Afzali-Kusha[1]
[1]*Nanoelectronics Center of Excellence, School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran*
[2]*Department of Computer Engineering, Islamic Azad University, Science and Research Branch, Tehran, Iran*
[3]*Department of Information Technology, University of Turku, Turku, Finland*
E-mail: masdan@utu.fi

Abstract: In this study, a low-distance path-based multicast routing algorithm for network-on-chips (NoCs) and multiprocessor systems-on-chip is proposed. The algorithm, which is based on the mesh topology, makes use of network partitioning, optimised destination ordering and the odd–even turn model adaptive routing technique for both the multicast and unicast messages. Additionally, the algorithm invokes non-congested paths in routing the messages to prevent creating highly congested areas. This is achieved by considering the congestion condition of the input ports. The efficiency of the proposed multicast routing algorithm is evaluated by comparing its performance with those of previously proposed algorithms under both multicast and mixed (mixture of unicast and multicast) traffic models. The results show that the proposed technique has lower average delays and lower average and peak power consumptions compared to those of the other path-based multicasting algorithm for different message injection rates. The technique has a hardware overhead of less than 8%.

1 Introduction

As is predicted by the Moore's law, over a billion transistors could be integrated on a single chip in the near future. In these chips, over hundreds of functional intellectual property (IP) blocks and a large amount of embedded memory could be placed together to form a multiprocessor systems-on-chip (MPSoC) [1]. The performance of the MPSoC is highly dependent on the underlying communication mechanism and the communication requirements are critical design issues. Since the traditional bus-based communication solutions may not be used for these systems, another communication paradigm called network-on-chip (NoC) may be used to solve the global interconnection problems of these systems [1]. In fact, on-chip networks like computer networks may take the advantage of data packetisation to ensure the fairness of communication [2]. Since on-chip networks should use lighter and faster protocol layers, they do not need to follow all the standard schemes for the communication in computer networks. The communication in NoC (or MPSoC) can be either unicast (one-to-one) or multicast (one-to-many) [3]. In the unicast communication, a message (packet) is sent from a source (IP or memory) node to a single destination node (IP or memory), whereas in the multicast communication a message is transmitted from a source node to an arbitrary set of destination nodes. Thus, the former is a special case of the latter.

The multicast communication is frequently employed in many application of MPSoC such as replication [4], barrier synchronisation [5], cache coherency in distributed shared-memory architectures [6] and clock synchronisation [7]. Although the multicast communication can be implemented by multiple unicast communications, it produces too much unnecessary traffic increasing the latency and congestion in the network [8].

In this work, we present an adaptive multicast wormhole routing algorithm for two-dimensional (2D) mesh NoCs which is inspired by a multicast routing algorithms used in multicomputers [8]. The method, called low distance (LD), takes advantage of the odd–even turn model [9–11]. The rest of the paper is organised as follows. In Section 2, the multicast routing algorithms including the previous techniques and the suggested one are discussed, whereas the switch architecture is described in Section 3. The results are discussed in Section 4 with the summary and conclusion given in the last section.

2 Multicast routing algorithms

In this work, we consider NoCs with 2D mesh topologies which offer many desirable properties including better parallelism and scalability, low cross-section bandwidth and fixed degree of nodes compared with many other topologies for MPSoC interconnection [12]. Besides, meshes are suitable for a variety of applications including matrix computation, image processing and problems whose task graphs can be embedded naturally into the topology [13]. An $m \times n$ 2D mesh consists of N $(=m \times n)$ nodes, where each node has an associated integer coordinate pair (x, y) such that $0 \leq x < n$ and $0 \leq y < m$. Two nodes with coordinates (x_i, y_i) and (x_j, y_j) are connected by a communication channel if and only if $|x_i - x_j| + |y_i - y_j| = 1$.

The multicast communication has been exploited in multicomputers (see, e.g. [4–8]). Multicast routing algorithms can be classified as unicast based (UB) [14, 15], tree based [15] and path based [8].

2.1 UB multicast routing algorithm

UB is a simple multicast routing algorithm where multiple copies of the same message, as a unicast message, are routed independently towards every destination or to a subset of destinations [14]. The drawback of this scheme is that multiple copies of the same message are injected into the network increasing the network traffic. Furthermore, each copy of the message loses considerable startup latency at the source.

2.2 Tree-based multicast routing algorithm

In tree-based multicast routing approach, the destination set is partitioned at the source and separate copies of the message are sent through one or more outgoing channels. Here, a spanning tree is constructed where the source is considered as the root and the messages are sent down the tree [15]. This way, a message might be replicated at some of the intermediate nodes and forwarded along the multiple outgoing channels towards disjoint subsets of the destinations. Since there is no message buffering at routers, if one branch of the tree is blocked, all are blocked [16]. Since the message may not proceed forward, many channels may be in lockstep for extended periods resulting in an increasing network contention [16]. Although such schemes have to be used effectively in networks employing store-and-forward and virtual cut-through routing, tree-based routing incurs high congestion in wormhole networks [8]. A tree-based routing algorithm which supports multicasting in NoCs is called virtual circuit tree multicasting (VCTM) [12]. By using virtual circuit table (VCT) and content addressable memory, and sending separate unicast setup messages (look ahead signals) for each destination, it builds several virtual circuit trees through the destinations before the multicast messages are injected into the network. In this method, cyclic dependencies are avoided by using the dimension-order routing algorithm for both the setup and the multicast messages. The method, however, has some shortcomings. First, its complexity, and hence, hardware overhead strongly depends on the network size. Second, the VCTM is an efficient algorithm mostly for low injection rate network conditions, whereas for high injection rate conditions (or workloads near saturation) the path-based algorithms are more efficient [12]. Third, for updating the VCT, discrete unicast setup messages per destination should be sent by the source node. If the number of destinations grows, the number of unicast setup message will be increased, thereby reducing the performance. Therefore the VCTM scheme is more efficient for some applications where there is a significant reuse of a small percentage of multicasts [12]. In these cases, there are multicasts from the same source intended for the same destination set. An example includes token coherence protocol which uses one-to-all communication and has very few distinct multicast combinations [12]. Finally, as discussed for the case of multicomputers, the tree-based multicasting may cause a message to hold many channels for extended periods, thereby increasing network contention, and hence degrading the performance [13].

An example of a tree-based multicast routing in 5 × 5 2D-mesh is shown in Fig. 1 where the source node (2, 3) is selected as the root and a spanning tree is formed with respect to it. When the flits enter the routers at the branch point [nodes (1, 3) and (3, 3)], they are duplicated and forwarded to multiple output channels. Since there is no message buffering in the routers, if one branch of the tree is blocked, all are blocked. Therefore this scheme might lead to increasing network contention (and eventually deadlock).

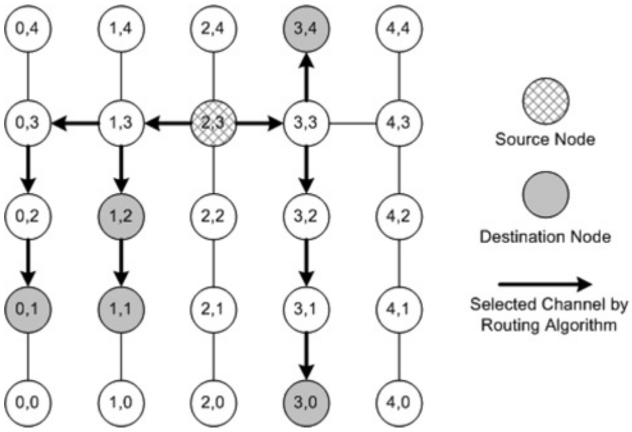


Figure 1 *Example of tree-based multicast routing in 5 × 5 2D mesh*

doi: 10.1049/iet-cdt.2008.0086

2.3 Hamilton path-based multicast routing algorithm

To overcome the disadvantages of the tree-based approaches, one may use path-based multicast wormhole routing algorithms. In this method, a source node prepares a message for delivery to a set of destinations by first sorting the addresses of the destination in the order in which they are to be delivered, and then placing this sorted list in the header of the message. When the header enters a router with the address A, the router checks to see if A is the next address in the header. If so, the address A is removed from the message header and a copy of data flits will be delivered to the local core and the flits are forwarded to the next node on the path. Otherwise, the message is forwarded only to the next node on the path. In this way, the message is eventually delivered to every destination in the header. A number of studies have shown that a path-based exhibit superior performance characteristic over their UB and tree-based counterparts [16–18].

The path-based routing algorithms are Hamilton path algorithm where a undirected Hamilton path of the network is constructed [8]. A Hamilton path visits every node in a graph exactly once [19]. For each node in an $m \times n$ mesh, a label $L(x, y)$ is assigned as

$$L(x, y) = \begin{cases} y \times n + x & \text{if } y \text{ is even} \\ y \times n + n - x - 1 & \text{if } y \text{ is odd} \end{cases}$$

where x and y are the coordinates of the node.

As exhibited in Fig. 2, two directed Hamilton paths (or two subnetworks) are constructed by labelling the nodes [8]. The high channel subnetwork (H_u) starts at (0, 0), whereas the low channel subnetwork (H_l) ends at (0, 0). If the label of the destination node is greater than the label of the source node, the routing always takes place in the H_u subnetwork; otherwise, it takes place in the H_l subnetwork. The destinations are placed into two groups. One group contains all the destinations that could be reached using the H_u subnetwork and the other contains the remaining destinations that could be reached using the H_l subnetwork. To reduce the path length, the vertical channels that are not part of the Hamilton path (the dashed lines in Fig. 2) could be used in appropriate directions. Next, we describe dual path (DP) [8], multi path (MP) [8] and column path (CP) [17] multicast routing algorithms along with the proposed algorithm in this work.

2.3.1 DP and MP multicast routing algorithms: In DP routing algorithm, the destination node set is partitioned into two subsets of D_H and D_L [8]. Every node in D_H has a higher label than that of the source node and every node in D_L has a lower label than that of the source node. D_H and D_L which are then sorted in ascending order and descending order, respectively, with the label of each node is used as the key for the sorting. Thus, multicast messages from the source node will be sent to the destination nodes in D_H using the H_u subnetwork and to the destination nodes in D_L using the H_l subnetwork. Consider the example shown in Fig. 3*a* for a 6 × 6 mesh network where node (2, 3) will send its multicast messages to destinations (2, 0), (4, 0), (0, 1), (2, 1), (4, 1), (0, 4), (5, 4), (3, 5) and (5, 5). Two subsets are organised. The first subset (D_H), which contains all the destinations that could be reached from the source node using H_u subnetwork, includes (0, 4), (5, 4), (5, 5) and (3, 5) in sequence. The second subset (D_L), which has the remaining destinations that all could be reached using the H_l subnetwork, includes (2, 0), (4, 0), (4, 1), (2, 1) and (0, 1). Some of the vertical links that are not part of the Hamilton paths are used properly, for minimising the paths.

To reduce the path lengths, the MP multicast routing algorithm has been proposed in [8]. In this scheme, as most nodes have four output channels in the 2D mesh, up

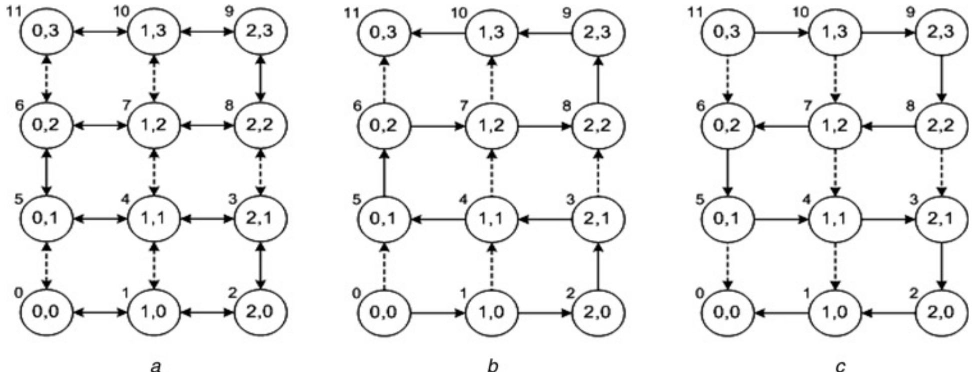


Figure 2 *In a 3 × 4 2D mesh network with the label assignment and the corresponding [17]*

a Full channel
b High channel
c Low channel networks
The solid lines indicate the Hamilton path and dashed lines indicate the links that could be used to reduce the path length in routing

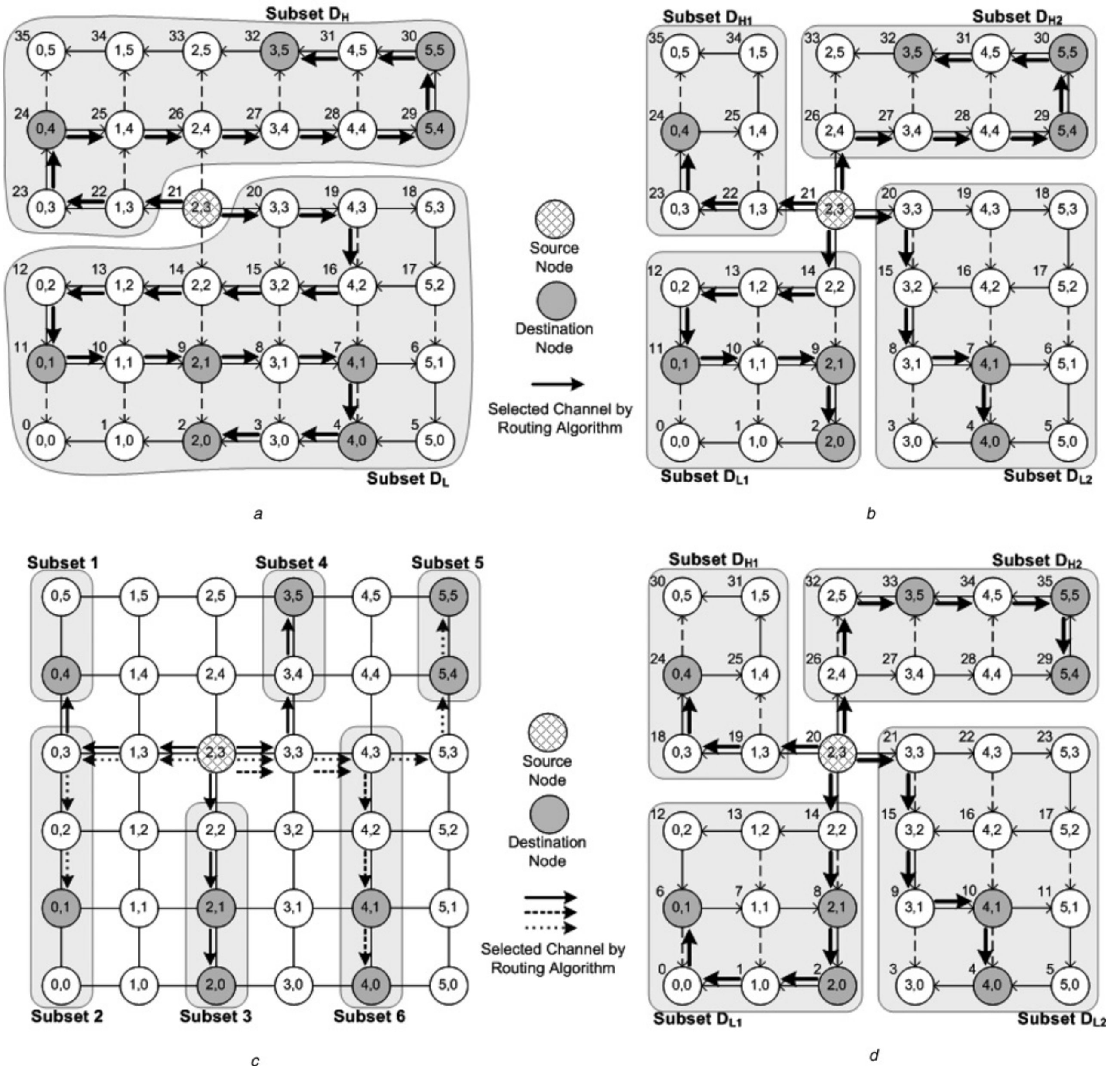


Figure 3 *Examples of*

a DP
b MP
c CP
d LD multicast routing from (2, 3)
The unused links are not indicated

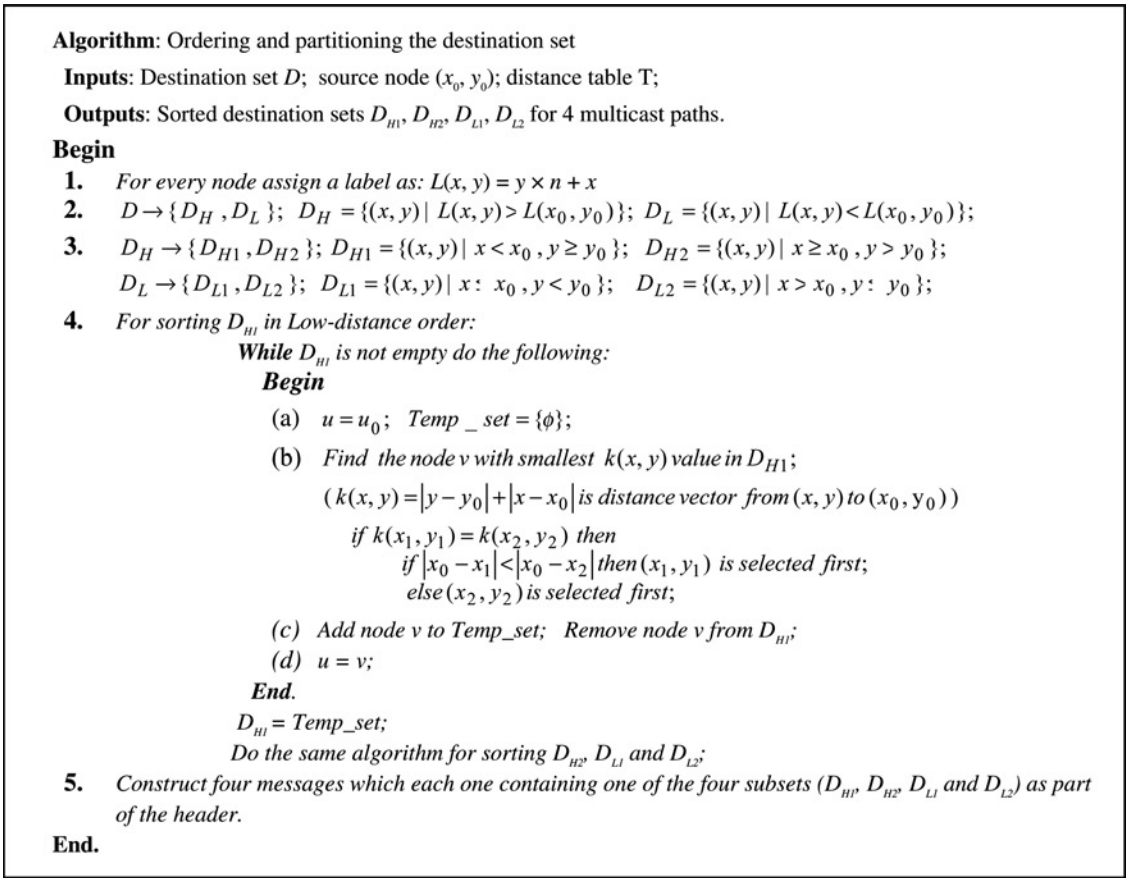
to four independent paths can be used to deliver a message. Thus, the DP destination sets of D_H and D_L are also partitioned. The set D_H is divided into two subsets. One consists of the nodes whose x coordinates are greater than or equal to that of the source and the other subset contains the remaining nodes in D_H. The set D_L is partitioned in a similar way. Hence, all the destinations of the multicast message are grouped into four disjoint subsets such that all the destinations in a subset are in one of the four quadrants when the source is taken as the origin. For the MP example shown in Fig. 3*b*, at source (2, 3) the destination set is first divided into two sets of $D_H = \{(0, 4), (5, 4), (3, 5), (5, 5)\}$ and $D_L = \{(2, 0), (4, 0), (4, 1), (2, 1), (0, 1)\}$. As exhibited in Fig. 3*a*, D_H is divided into two subsets of $D_{H1} = \{(0, 4)\}$ and $D_{H2} = \{(5, 4), (3, 5), (5, 5)\}$. In the same way, D_L is divided into two subsets of $D_{L1} = \{(0, 1), (2, 1), (2, 0)\}$ and $D_{L2} = \{(4, 0), (4, 1)\}$. The DP and MP are both deadlock free and could be used for unicast and multicast routing simultaneously [8].

2.3.2 CP multicast routing: In this method, the destination node set is partitioned into $2k$ subsets, where k

is the number of columns in the mesh. In this method, at most two messages will be copied to each column. If a column of the mesh has one or more destinations in the rows above that of the source, then one copy of the message is sent to service all those destinations. Similarly, if a column has one or more destinations in the rows below that of the source, then another copy of the message is sent to service all those destinations. One copy of the message is sent to a column if all destinations in that column are either below or above the source node. Fig. 3c shows an example where a multicast message is sent to destinations (2, 0), (4, 0), (0, 1), (2, 1), (4, 1), (0, 4), (5, 4), (3, 5) and (5, 5) from source node (2, 3) using the CP routing algorithm. Six copies of the message are used to achieve the desired multicast operation. The routing algorithm used by this scheme is based on the row–column routing algorithm which is deadlock free and livelock free. However, since the CP routing algorithm, similar to the UB routing method, produces too many messages, it suffers from high network latencies for later copies of the messages because of the excessive number of start-up delays before them. In addition, because many multicast messages would be sent through the columns by each source node, the performance of the network is degraded.

2.3.3 Proposed LD path-based multicast routing:

In this work, we propose a path-based multicast routing algorithm. Three features have been incorporated in this scheme. First, it utilises a network partitioning similar to MP multicast routing technique where up to four destination groups could be formed. Second, the ordering of the destinations in the path must be optimised to shorten the distance of the multicast path. This is achieved at the cost of a small hardware overhead. This improves the performance of the algorithm compared with those of previous path-based multicast routing algorithms. For this propose, a sorting algorithm shown in Fig. 4 is proposed. In this algorithm, for each node a label obtained from $L(x, y) = y \times n + x$ is assigned. Similar to the MP multicast algorithm, the destination node set is partitioned into four subsets of D_{H1}, D_{H2}, D_{L1} and D_{L2}. The subsets are then sorted in the low-distance order with the distance vector of each node used as its key for the sorting. The distance vector of each node is computed as $k = |y - y_0| + |x - x_0|$. To sort the destinations to the low-distance order, first the node (v) that has the lowest distance vector to the source node (u_0) is placed in the Temp_set and is removed from the subset. Then, the selected node will be considered as the source node. While the original subset is not empty, this sequence will be repeated; otherwise, the Temp_set that contains the sorted destination subset is placed in the original subset. If there are two nodes with an equal distance vector compared to the source node, the one with the smaller x dimension relative to that of the source node



```
Algorithm: Ordering and partitioning the destination set
  Inputs: Destination set $D$;  source node $(x_0, y_0)$; distance table T;
  Outputs: Sorted destination sets $D_{H1}$, $D_{H2}$, $D_{L1}$, $D_{L2}$ for 4 multicast paths.
Begin
  1.  For every node assign a label as: $L(x, y) = y \times n + x$
  2.  $D \rightarrow \{D_H, D_L\}$;  $D_H = \{(x, y) \mid L(x, y) > L(x_0, y_0)\}$;  $D_L = \{(x, y) \mid L(x, y) < L(x_0, y_0)\}$;
  3.  $D_H \rightarrow \{D_{H1}, D_{H2}\}$;  $D_{H1} = \{(x, y) \mid x < x_0, y \geq y_0\}$;  $D_{H2} = \{(x, y) \mid x \geq x_0, y > y_0\}$;
      $D_L \rightarrow \{D_{L1}, D_{L2}\}$;  $D_{L1} = \{(x, y) \mid x :\ x_0, y < y_0\}$;  $D_{L2} = \{(x, y) \mid x > x_0, y :\ y_0\}$;
  4.  For sorting $D_{H1}$ in Low-distance order:
              While $D_{H1}$ is not empty do the following:
                Begin
                  (a)  $u = u_0$;  $Temp\_set = \{\phi\}$;
                  (b)  Find the node v with smallest $k(x, y)$ value in $D_{H1}$;
                       ( $k(x, y) = |y - y_0| + |x - x_0|$ is distance vector from $(x, y)$ to $(x_0, y_0)$ )
                         if $k(x_1, y_1) = k(x_2, y_2)$ then
                             if $|x_0 - x_1| < |x_0 - x_2|$ then $(x_1, y_1)$ is selected first;
                             else $(x_2, y_2)$ is selected first;
                  (c)  Add node v to Temp_set;  Remove node v from $D_{H1}$;
                  (d)  $u = v$;
                End.
              $D_{H1}$ = Temp_set;
              Do the same algorithm for sorting $D_{H2}$, $D_{L1}$ and $D_{L2}$;
  5.  Construct four messages which each one containing one of the four subsets ($D_{H1}$, $D_{H2}$, $D_{L1}$ and $D_{L2}$) as part
      of the header.
End.
```

Figure 4 *Message header construction for LD multicast routing* will be selected first. Subsequently, the original subset will be placed in the message header. Third, for routing the messages to the destinations, the algorithm utilises the odd–even turn model [9, 11]. The odd–even turn model prohibits the east to north and east to south (north to west and south to west) turns at any switches located in an even (odd) column. This makes the technique as an adaptive deadlock free algorithm which uses the minimum path. Since it is deadlock free, there is no need for implementing virtual channels in the router to prevent the deadlock problem [20]. Adding virtual channels is costly since the complexity and latency of the controller increase with the number of virtual channels because of increased buffering and arbitration requirements [21]. In a few cases, for routing a multicast message from one destination to the next destination via a minimal path requires a forbidden turn. To prevent a possible deadlock in these cases, the message is first absorbed by the first destination and then a copy of the message will be retransmitted to the next destination address through the consumption channels discussed in Section 3.3. This way the deadlock may be prevented [22]. Fig. 3*d* shows an example of the paths used for the message when the proposed multicast routing algorithm is used.

3 Proposed switch structure

3.1 Topology and switching method

As mentioned before, we make use of an $n \times n$ network of interconnected tiles with a mesh topology. Each tile is composed of a processing element (PE) and a router connected to its four adjacent routers in addition to the PE of the tile through some channels [23]. Two unidirectional point-to-point links form the channel. To minimise the delay and the required resources, we have used the wormhole method for the switching. In this method, a message is divided into smaller segments called FLow control digIT (FLITs), which are routed successively until they reach their destination [12].

3.2 Message format

The multicast message format is shown in Fig. 5. It includes one or several header flits and a parametric number of payload flits. The number of flits depends on the number of destinations and the flit width in the network. Each flit is n bit wide where the nth bit is the end of message sign and the $(n-1)$th bit is the begin of message (BOM) sign. In the header, the third field, which is represented by T, is used to describe the type of the message. There are two types of message which are unicast ($T=0$) and multicast ($T=1$). The address the source node (SA), the pointer counter (P) and the destination node address(es) (DA) are placed in the last fields of the header, respectively, and the content of the message is located in the rest of the flits (payload). The pointer in each header flit points to the address of the next destination in the current header flit and the message identifier (MID) is used for the message ordering.

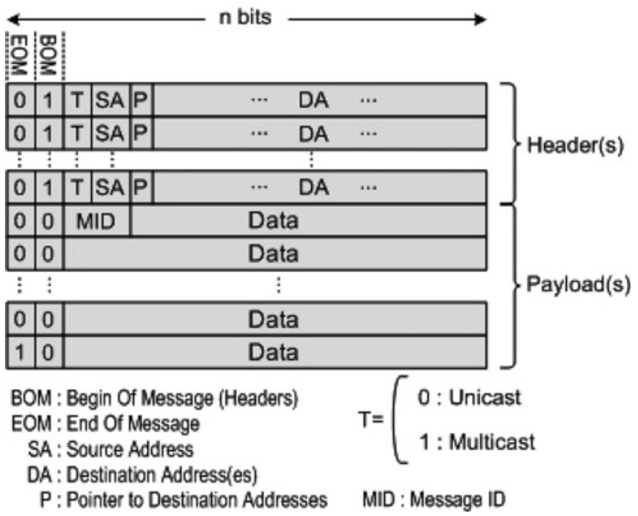


Figure 5 *Multicast message format for the proposed technique*

3.3 Switch structure

Now, we describe the implementation details of the proposed router which is shown in Fig. 6. Each input port has a controller for handshaking and an input buffer for the temporary storage of flits. The wormhole switching method implemented in the controller unit is based on on/off flow control mechanism [24, 25]. After receiving the message header, first the routing unit determines which output should be used for routing this message and then the arbiter requests for a grant to inject the message to a proper output port using the crossbar switch. The router has a crossbar which establishes the connection path from an input port to an output port. Since a crossbar can only serve as a single output port at a time, it uses an arbiter for the arbitration among simultaneous input requests to access the same output port. When a new message reaches the input port, it waits until the previously arrived messages leave the port. Then, the new message header is delivered to the routing unit where it is routed to the appropriate output port. The congestion flag (CF) [20] of the buffer becomes active when the number of empty cells of the buffer is less than a threshold value. In this case, for warning about the full status, the signal CF is activated indicating that most buffer cells are full. Each input port has a CF through which it informs its adjacent router about its congestion condition. Therefore the router which uses that input port for forwarding a message to the next router should consider this router as a congested one (congestion area or hotspot) and should not send messages to this router until the congestion is over.

In the path-based multicast mechanism, when multiple delivery channels are occupied by one message along the multicast path, cyclic dependencies on the delivery channels

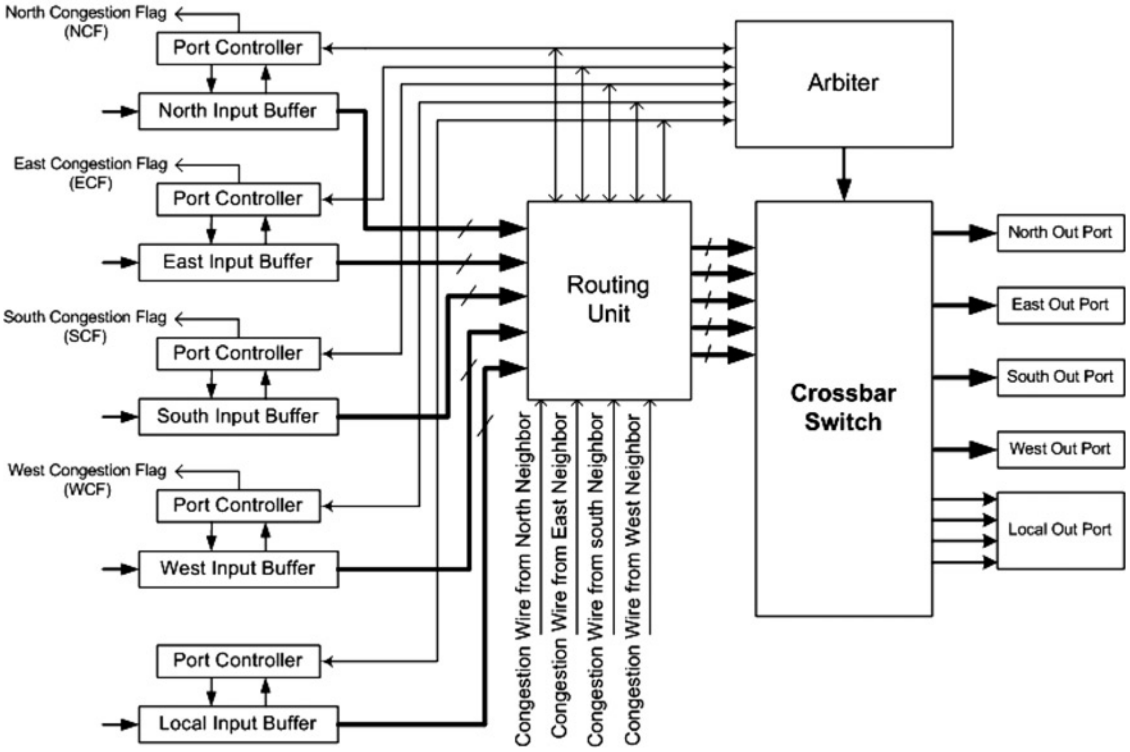


Figure 6 *Proposed router structure*

may occur [17, 24, 26, 27]. As illustrated in Fig. 7, the multicast message A destined to nodes 2 and 3 is generated by node 1. Simultaneously, node 4 generates the message B destined to the same set of destinations. As a result, because of the delivery channel contention, this cyclic wait creates a deadlock. To prevent deadlocks in delivery (consumption) channels, the upper bound of the number of delivery channels required to avoid such deadlocks is equal to $2nv$, where n is the network dimension and v is the number of virtual channels per input port [17, 27]. As a result, at least two delivery channels are necessary and sufficient for DP, MP and CP algorithms and four delivery channels are enough to support deadlock-free multicasting mechanism under the LD model in 2D meshes when the base routing is either XY, odd–even, or the other turn model routing algorithms [22, 27].

The router employs a routing unit which decodes the header of the message coming from an input port. If the header belongs to a unicast message ($T = 0$), the minimal path adaptive routing algorithms based on the odd–even turn model is used to determine the output port to which the message should be sent. In the odd–even adaptive routing algorithm there could be more than one minimal output direction to route the message. In this case, the address decoder will choose the direction in which the corresponding downstream router has not raised its CF. For instance, if a message with a given source and destination could be routed to both output ports of p1 (CF = 0) and p2 (CF = 1), then it will be routed to p1. If p1 and p2 happen to have both their CFs raised, the message will be routed to p1. On the other hand, if the header type is a multicast message ($T = 1$), the routing unit fetches the destination address specified by the pointer in the header. If the destination address is the current node, the routing unit will request the local output port. Otherwise, the routing unit fetches the next destination address from the header and runs the odd–even procedure

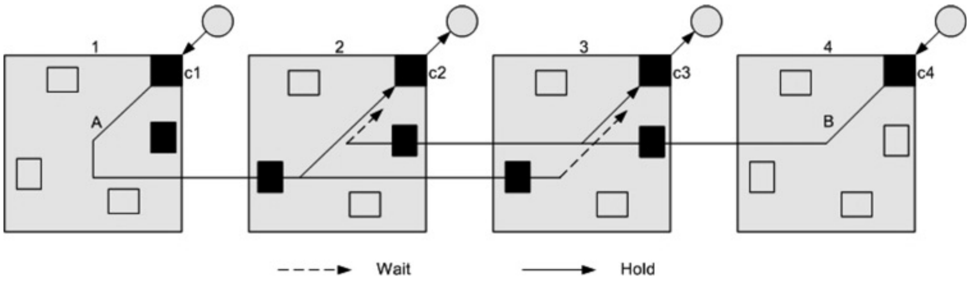


Figure 7 *Deadlock because of the delivery channel contention [22]*

doi: 10.1049/iet-cdt.2008.0086

to determine the output port(s) corresponding to the next destination address. Also, after fetching, the routing unit increases the pointer value of the header, and if it is overflowed, it means that the multicast message has been sent to all the destination addresses in this header flit, the routing unit will remove the corresponding header flit from the message.

It should be noted that as a result of exploiting the adaptive odd–even routing algorithm, the messages of the same data may traverse different paths reaching at the destination out-of-order. Hence, a technique may be needed to reorder the messages at the destination. In the proposed technique in this work, the messages that reach the destination node have the information about the message source node (SA) and the message order (MID). Using the SA and MID, the destination core may store each message in its proper location in the core memory such that the original source order can be achieved with negligible overhead. Note that the data in the memory might not be processed by the core unless all parts of the data are received. This is also true for deterministic multicast routing algorithms. Also, the use of the source address enables the destination to concurrently handle different data coming from different sources.

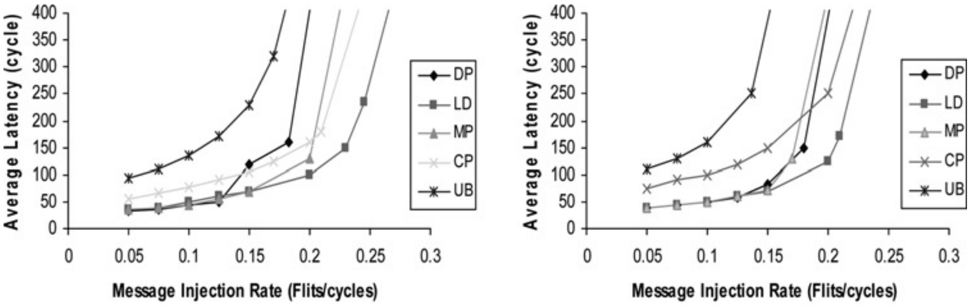
4 Results and discussion

To assess the efficiency of the LD path-based multicast routing algorithm, three other multicast routing algorithms were also implemented. These algorithms included DP, MP and CP. We have developed a flit level event-driven wormhole NoC simulator implemented in C++ based on standard template libraries, running under Fedora Linux OS. The simulator calculates the average delay and the power consumption for the message transmission. This simulator can be used for the wormhole switching in NoCs with the 2D mesh topology. The simulator inputs include the array size, the switch operating frequency, the routing algorithm, the link width length and the traffic type. The simulator can generate different traffic profiles. To calculate the power consumption, we have used Orion library functions [28]. For all switches, the data width and the frequency were set to 32 bits and 1 GHz, respectively, which led to a bandwidth of 32 Gb/s. Each input channel had a buffer [first in first out (FIFO)] size of eight flits with the congestion threshold set at 75% of the total buffer capacity. The message size was assumed to be 16 flits. In addition, we also assumed that the 2D mesh topology was regular and the delays on wires would not exceed the clock period. For the long communication channel cases that the delays on wires exceed the clock period, the channels should be pipelined by inserting some repeaters such as FF-repeater or SR-repeater [25]. This requires additional buffer resources and will be discussed in our future work.

For the performance metric, we use the multicast latency defined as the number of cycles between the initiation of the multicast message operation and the time when the tail of the multicast message reaches all the destinations. The CP has the most complicated procedure to prepare the multicast messages, whereas the DP has the easiest procedure [17]. The preparation mechanism consists of partitioning the destination set into appropriate subsets and creating multiple copies of the message. For computing the preparation time, we have run several sets of multicast destinations by our simulator. Under these test sets, the average preparation time to complete multicast messages in the DP, MP, LD and CP algorithms were 35, 46, 46 and 82 cycles, respectively. Because the DP algorithm generates only two multicast messages, it is the best among the other algorithms and the CP is the worst in terms of the startup latency.

4.1 Multicast traffic profile

The first sets of simulations were performed for a random traffic profile. In these simulations, the PEs generate five flit data messages and inject them into the network using the time intervals which are obtained based on the exponential distribution. Two mesh sizes of 8 × 8 and 16 × 16 have been considered. In the multicast traffic profile, each PE sends a message to a set of destinations. A uniform distribution was used to construct the destination set of each multicast message [8]. The number of destinations were set to 10 and 25. In Figs. 8 and 9, the average communication delay as a function of the average



a *b*

Figure 8 *Performance under different loads in 8 × 8 2D mesh with*

a 10 destinations
b 25 destinations

doi: 10.1049/iet-cdt.2008.0086

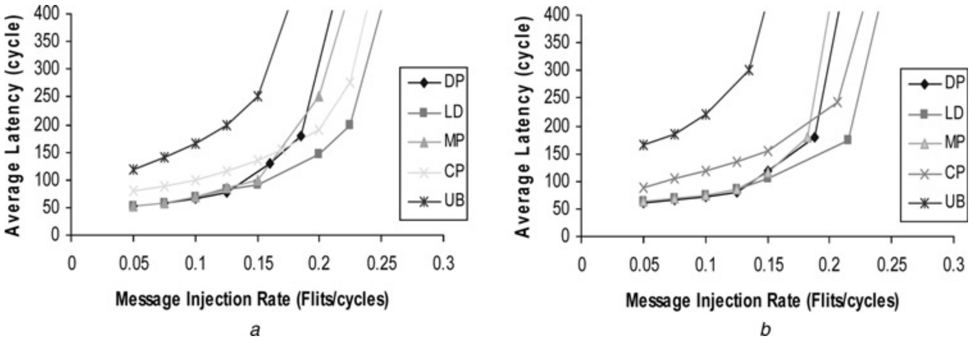


Figure 9 *Performance under different loads in 16 × 16 2D mesh with*

a 10 destinations
b 25 destinations

flit injection rate is shown. As the results show, the LD multicast routing algorithm leads to the lowest latency among all the three multicast routing algorithms even at high traffic loads or with a large number of destinations (25 destinations).

4.2 Unicast and multicast (mixed) traffic profiles

In these simulations, we employed a mixture of unicast and multicast traffic where 80% of the injected messages are unicast messages and the remaining 20% are multicast messages. This pattern may represent the traffic in a distributed shared-memory multiprocessor, where updates and invalidation produce multicast messages and cache misses are served by unicast messages [15, 17]. For this set, the simulation parameters were similar to the previous simulations in terms of the number of destinations and array sizes. The unicast messages are also routed using the odd–even turn model. Uniform [29, 30] and hotspot [11, 29] were the two different traffic profiles considered for the unicast traffic generation. In the uniform traffic profile, each PE sends a message to any other PE with an equal probability. Therefore the destinations are determined randomly using a uniform distribution. 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 Figs. 10 and 11, the average communication latencies against the message injection rate for different algorithms under the uniform traffic model for unicast traffic profile are shown. As these figures reveal, for this traffic profile, the proposed routing algorithm outperforms the other three algorithms. Under the hotspot traffic model with the hotspot percentage of h, a newly generated message is directed to each hotspot node with an additional h per cent probability. In our simulations, we assumed a single hotspot node. The hotspot node is chosen to be node (4, 4) in the 8 × 8 2D mesh and node (8, 8) in the 16 × 16 2D mesh. Figs. 12 and 13 show the average latencies of the algorithms for the two mesh topologies when $h = 10\%$. As the figures show, the proposed routing algorithm considerably outperforms the other algorithms for different numbers of destinations and mesh sizes under various message injection rates.

4.3 Power dissipation

The power dissipation of DP, MP, CP, UB, and the proposed routing algorithms were calculated and compared under the multicast traffic model. The results for the average and maximum powers under this traffic are shown in Figs. 14 and 15, respectively. As the results presented in Table 1 for ten destinations reveal, the average power dissipation of the network with the proposed algorithm is 25, 3.5, 33 and 63% less than those of the DP, MP, CP

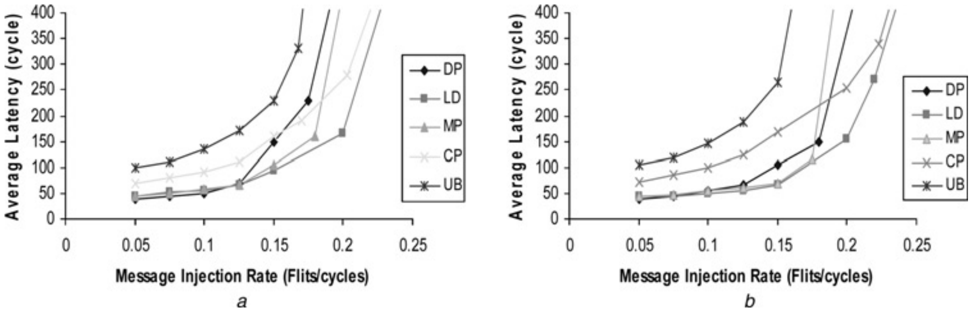


Figure 10 *Performance under different loads in 8 × 8 2D mesh with*

a 10 destinations
b 25 destinations under mixed traffic (20% multicast and 80% unicast). Unicast traffic is based on the uniform traffic model

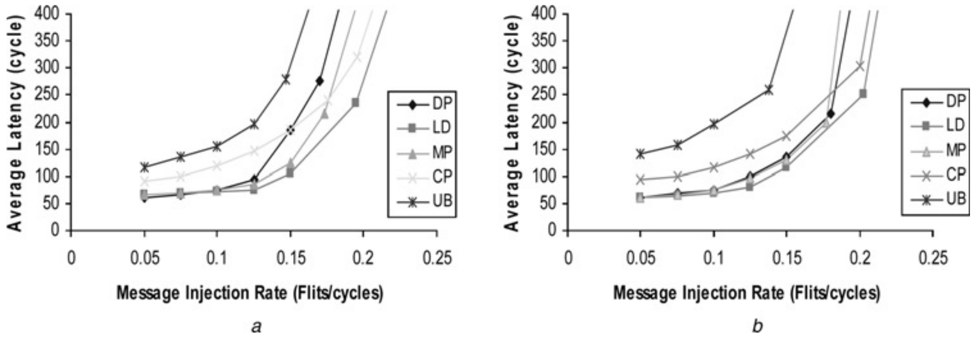


Figure 11 *Performance under different loads in 16 × 16 2D-mesh with*

a 10 destinations
b 25 destinations under mixed traffic (20% multicast and 80% unicast). Unicast traffic is based on the uniform traffic model

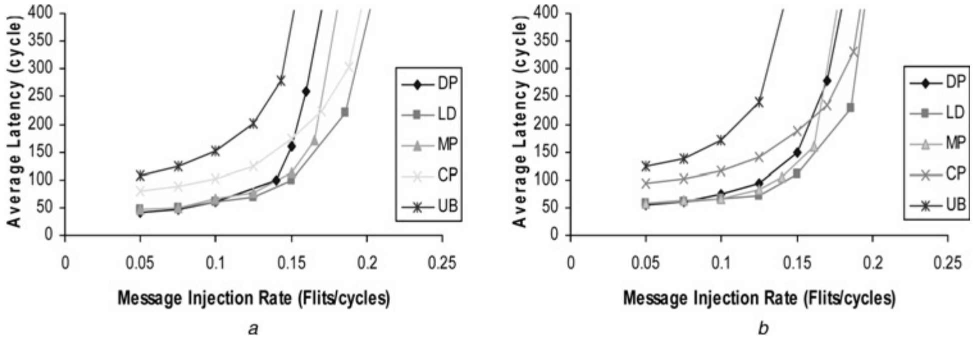


Figure 12 *Performance under different loads in 8 × 8 2D mesh with*

a 10 destinations
b 25 destinations under mixed traffic (20% multicast and 80% unicast). Unicast traffic is based on the hotspot traffic model with a single hotspot node (4, 4). The hotspot percentage is 10%

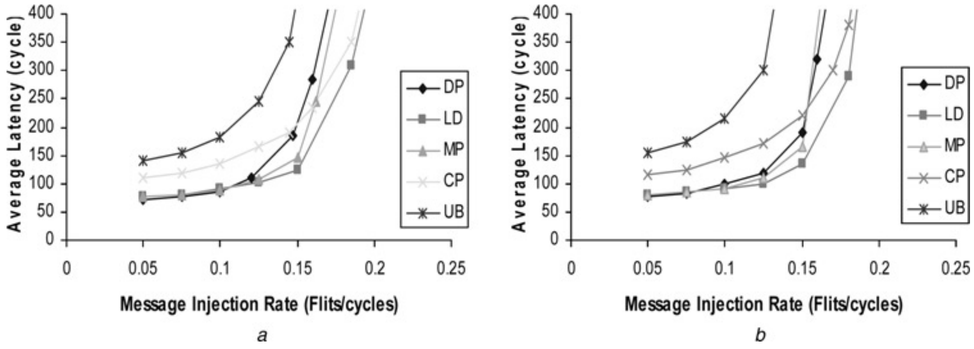


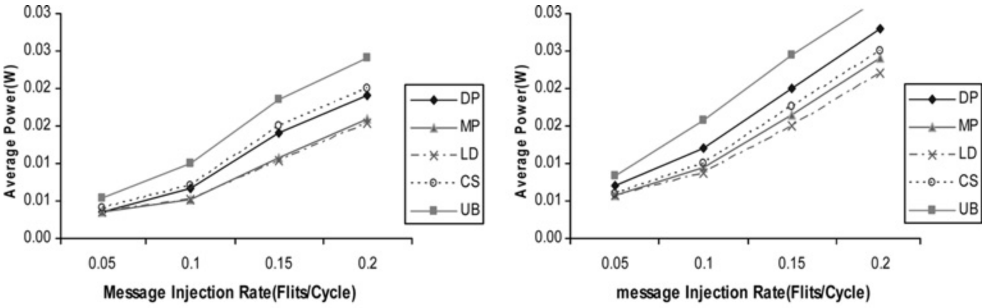
Figure 13 *Performance under different loads in 16 × 16 2D mesh with*

a 10 destinations
b 25 destinations under mixed traffic (20% multicast and 80% unicast). Unicast traffic is based on the hotspot traffic model with a single hotspot node (8, 8). The hotspot percentage is 10%

and UB algorithms under the multicast traffic model, respectively. Also, the results of Table 2 for ten destinations indicate that the peak power of the proposed algorithm is 27, 8, 44 and 70% less than those of the DP, MP, CP and UB algorithms, respectively, under the multicast traffic model. Similar power savings are obtained for 25 destinations. The power reduction for the proposed algorithm is achieved by smoothly distributing the power consumption over the network using the adaptive routing scheme which reduces the number of the hotspots, and hence, lowering both the average power and the peak power.

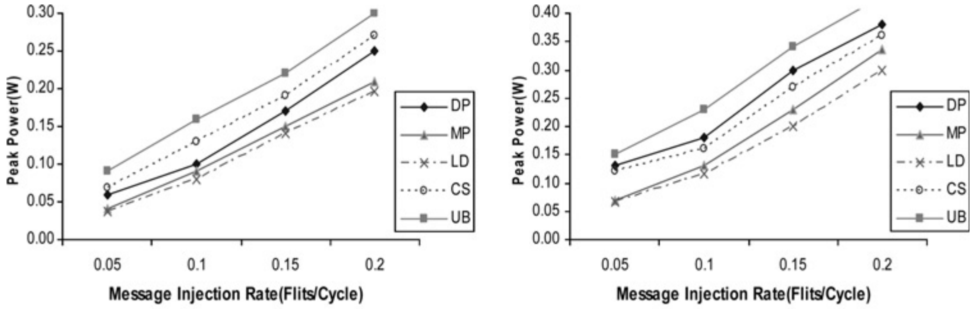
4.4 Hardware overhead

To evaluate the area overhead of the proposed algorithm, we designed the switches based on the multicast routing



a *b*

Figure 14 *Average power dissipation of the proposed, the DP, the MP and the CP algorithms in 16 × 16 2D mesh with*
a 10 destinations and
b 25 destinations under multicast traffic



a *b*

Figure 15 *Maximum power dissipation of the proposed, the DP, the MP and the CP algorithms in 16 × 16 2D mesh with*
a 10 destinations and
b 25 destinations under multicast traffic

Table 1 Comparative average power dissipation of the proposed algorithm with other algorithms in 16 × 16 2D mesh

Average power dissipation	DP	MP	CP	UB
with ten destinations (%)	−25	−3.5	−33	−63
with 25 destinations (%)	−32	−8.5	−13	−51

Table 2 Comparative maximum power dissipation of the proposed algorithm with other algorithms in 16 × 16 2D mesh

Peak power dissipation	DP	MP	CP	UB
With ten destinations (%)	−27	−8	−44	−70
with 25 destinations (%)	−43	−12	−33	−64

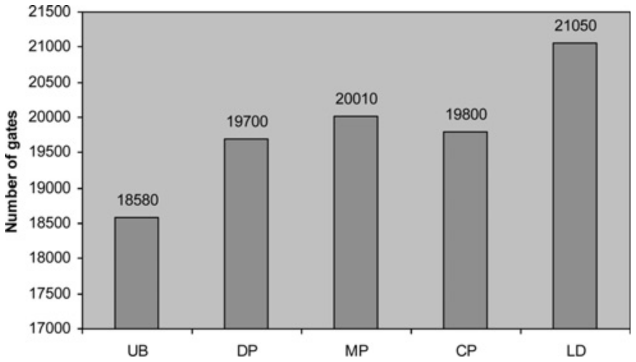


Figure 16 *Area cost of switches for implementing different multicast routing algorithms*

schemes including the additional hardware required for each scheme as described in Section 3.3. The switches were described in VHDL for a 16 × 16 2D mesh NoC environment, and synthesised with the Leonardo-Spectrum ASIC using a 0.25 μm standard cell library. In addition, the destination sorting algorithms were included in the hardware overhead. For all the switches, the data width was set to 32 bits (flit size) and each input channel had a buffer size of eight flits. As discussed in Section 3.3, for the DP, MP and CP switches, we used two delivery channels, and for the LD switch we used four delivery channels. In order to achieve better performance/power efficiency, the FIFOs were implemented using registers. Fig. 16 shows the area cost of the switches. Although the same switch structure was used for the CP, MP and DP multicasting schemes,

doi: 10.1049/iet-cdt.2008.0086

different number of registers were employed in implementing their sorting mechanisms leading to different areas for the algorithms. Comparing the area cost of the proposed switch with those of the UB, DP, MP and CP switches indicates an additional overhead of 11, 6.4, 5 and 6%, respectively.

5 Summary and conclusion

In this work, a path-based multicast routing algorithm for 2D mesh NoCs was proposed. To enhance the efficiency, the technique used network partitioning, optimised destination ordering, and odd–even turn model adaptive algorithm for routing both the multicast and unicast messages through the network. Additionally, the adaptive routing algorithm used the congestion condition of the input ports to route the messages through non-congested paths. A C++ simulator was used to evaluate the latency and power consumption of the proposed multicast routing algorithm. The simulations were performed under the multicast and mixed (mixture of unicast and multicast) traffic models for different flit injection rates. The proposed algorithm was compared with four different multicast routing algorithms including the DP, MP, CP and UB algorithms. The comparison showed that the proposed technique had the lowest average communication delay under the multicast and mixed traffic models. It also reduced both the average and the maximum power dissipations of the network. The additional area overhead of the algorithm compared to these techniques was small.

6 Acknowledgment

M. Daneshtalab and A. Afzali-Kusha acknowledge the financial support by the Research Council of the University of Tehran. Also, M. Ebrahimi acknowledges the financial support by Iran Telecommunication Research Center.

7 References

[1] CESARIO W.O., LYONNARD L., NICOLESCU G., *ET AL.*: 'Multiprocessor SoC platforms: a component-based design approach'. Proc. Int. Conf. IEEE Design and Test of Computers, Paris, France, 2002, pp. 52–63

[2] RIJPKEMA E., GOOSSENS K.G.W., RADULESCU A., *ET AL.*: 'Trade offs in the design of a router with both guaranteed and best-effort services for networks on chip'. Proc. Int. Conf. Design Automation and Test Conf., Europe, Munich, Germany, 2003, pp. 350–355

[3] LU Z., YIN B., JANTSCH A.: 'Connection-oriented multicasting in wormhole-switched networks on chip'. Proc. Int. Conf. ISVLSI, Germany, 2006, pp. 205–210

[4] MCKINLEY P.K., XU H., KALNS E., NI L.M.: 'CompaSS: efficient communication services for scalable architectures'. Proc. Int. Conf. Supercomputing, Minnesota, USA, 1992, pp. 478–487

[5] XU H., MCKINLEY P.K., KALNS E., NI L.M.: 'Efficient implementation of barrier synchronization in wormhole-routed hypercube multicomputers', *J. Parallel Distrib. Comput.*, 1992, **16**, pp. 172–184

[6] LI K., SCHAEFER R.: 'A hypercube shared virtual memory'. Proc. Int. Conf. ICPP, USA, 1989, pp. 125–132

[7] AZEVEDO M., BLOUGH D.: 'Fault-tolerant clock synchronization of large multicomputers via multistep interactive convergence'. Proc. Int. Conf. ICDCS, Hong Kong, 1996, pp. 249–257

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

[9] HU J., MARCULESCU R.: 'DyAD-smart routing for networks-on-chip'. Proc. Int. Conf. DAC, California, USA, 2004, pp. 260–263

[10] GLASS C.J., NI L.M.: 'The turn model for adaptive routing'. Proc. Int. Conf. Computer Architecture, USA, 1992, pp. 278–287

[11] CHIU G.: 'The odd–even turn model for adaptive routing', *IEEE Trans. Parallel Distrib. Syst.*, 2000, **11**, pp. 729–738

[12] JERGER N.E., PEH L.S., LIPASTI M.H.: 'Virtual circuit tree multicasting: a case for on-chip hardware multicast support'. Proc. Int. Conf. Computer Architecture, China, 2008, pp. 229–240

[13] DUATO J., YALAMANCHILI S., NI L.: 'Interconnection networks: an engineering approach' (Morgan Kaufmann Publishers, 2003)

[14] MCKINELY P., XU H., ESFAHANIAN A.H., NI L.: 'Unicast-based multicast communication in wormhole-routed networks', *IEEE Trans. Parallel Distrib. Syst.*, 1994, **5**, pp. 1252–1265

[15] MALUMBRES M.P., DUATO J., TORRELLAS J.: 'An efficient implementation of tree-based multicast routing for distributed shared-memory multiprocessors'. Proc. Int. Conf. SPDP, New Orlean, USA, 1996, pp. 186–190

[16] AL-DUBAI A., ROMDHANI I.: 'A performance study of path based multicast communication algorithms'. Proc. Int. Conf. PARELEC, Bialystok, Poland, 2006, pp. 245–250

[17] BOPPANA R.V., CHALASANI S., RAGHAVENDRA C.S.: 'Resource deadlock and performance of wormhole multicast routing algorithms', *IEEE Trans. Parallel Distrib. Syst.*, 1998, **9**, pp. 535–549

[18] PANDA D., SINGAL S., KESAVAN R.: 'Multi destination message passing in wormhole k-ary n-cube networks with base

routing conformed paths', *IEEE Trans. Parallel Distrib. Syst.*, 1999, **10**, pp. 76–96

[19] HARARY F.: 'Graph theory' (Readings, Addson-Wesley, MA, 1972)

[20] DANESHTALAB M., PEDRAM A., NEISHABURI M.H., MOHAMMADI S., AFZALI-KUSHA A.: 'Distributing congestions in nocs through a dynamic routing algorithm based on input and output selections'. Proc. Int. Conf. VLSID, India, 2007, pp. 546–550

[21] NI L.M., MCKINLEY P.K.: 'A survey of wormhole routing techniques in direct networks', *IEEE Trans. Comput.*, 1993, **26**, pp. 62–76

[22] MOHAPATRA P., VARAVITHYA V.: 'A hardware multicast routing algorithm for two-dimensional meshes'. Proc. Int. Conf. SPDP, New Orleans, 1996, pp. 198–205

[23] LIANG J., SWAMINATHAN S., TESSIER R.: 'aSOC: a scalable, single-chip communication architectures'. Proc. Int. Conf. PACT, Oregon, USA, 2000, pp. 37–46

[24] DALLY W.J., TOWLES B.: 'Principles and practices of interconnection networks' (Morgan Kaufmann Publishers, San Mateo, CA, 2004)

[25] CONCER N., PETRACCA M., CARLONI L.: 'Distributed flit-buffer flow control for networks-on-chip'. Proc. Int. Conf. CODES + ISSS, Georgia, USA, 2008, pp. 215–220

[26] CARARA E.A., MORAES F.G.: 'Deadlock-free multicast routing algorithm for wormhole-switched mesh networks-on-chip'. Proc. Int. Conf. ISVLSI, France, 2008, pp. 341–346

[27] PANDA D.K., SINGAL S., KESAVAN R.: 'Multidestination message passing in wormhole *k*-ary *n*-cube networks with base routing conformed paths', *IEEE Trans. Parallel Distrib. Syst.*, 1999, **10**, pp. 76–96

[28] WANG H., ZHU X., PEH L.S., MALIK S.: 'Orion: a power-performance simulator for interconnection network'. Proc. Int. Conf. Hot Interconnection, Stanford, USA, 2002, pp. 294–305

[29] BOPPANA R.V., CHALASANI S.: 'A comparison of adaptive wormhole routing algorithms'. Proc. Int. Conf. Computer Architecture, USA, 1993, pp. 351–360

[30] FLUGHAM M.L., SNYDER L.: 'Performance of chaos and oblivious routers under non-uniform traffic'. Technical Report UW-CSE-93-06-01, University of Washington, July 1993

doi: 10.1049/iet-cdt.2008.0086