2012 International Congress on Informatics, Environment, Energy and Applications-IEEA 2012
IPCSIT vol.38 (2012) © (2012) IACSIT Press, Singapore

A Novel Multicast Routing for two-dimensional de Bruijn NoCs

Abolfazl Malek Mohammadi[1] and Reza Sabbaghi-Nadooshan[2]

[1]Member of Scientific Association of Electrical &Electronic Engineering, Islamic Azad University Central Tehran Branch

[2]Islamic Azad University Central Tehran Branch, Tehran, Iran

Abstract. In this paper, we have proposed a novel multicast routing for two-dimensional de Bruijn Networks-on-Chips (NoCs). The proposed routing compared with unicast routing. Based on comparison results, the proposed routing 1) offers better performance under uniform and hot spot traffics pattern and 2) consumes lower energy for packet delivery.

Keywords: NoCs, De Bruijn, Routing, Multicast, Performance, Power.

1. Introduction

Network-on-Chip (NoC) is a promising communication paradigm for multiprocessor Systems-on-Chip (SoCs). This communication paradigm was inspired from interconnection networks and aims to overcome the performance and scalability problems of the shared buses and point-to-point dedicated links in multi-core SoCs [1].

The choice of network topology is an important issue in designing NoCs [2]. Different topologies provide different average inter-node distances, total wiring lengths, and communication flow distributions [3]. These characteristics mainly determine the power consumption and average packet latency [4].

Among the proposed topologies for NoCs, de Bruijn topology has interesting properties as compared to the popular mesh topology. De Bruijn topology has the same number of links as a linear array has; it also offers logarithmic diameter, which make it faster than an equal-sized linear array. In the before our articles [5] we have suggested two dimensional Bruijn for NoCs.

We can improve performance in the two dimensional Bruijn NoCs. We have used 3-d layout or torus as we used in [6][7]. Furthermore, we can improve routing algorithm. In this paper, we use multicast routing for the improvement of performance.

The rest of the paper is organized as follows. Section 2 presents multicast routing algorithm. Section 3 compares the simulation results of multicast routing and unicast routing. Finally in section 4, we conclude the paper.

2. Novel tree-based multicast routing

2.1. The routing introduction

Deterministic routing is based on minimum hop and the distance between source and destination nodes, unlike partially adaptive and fully adaptive is constant and is shown with d_{S-D}. In deterministic routing, if the specific node k is a part of the route, the number of hops between the nodes of source S and destination D equal to the distance between source node and specific node k plus the distance between node k and destination node and vice versa. Above condition will be $d_{S-D} = d_{S-k} + d_{k-D}$.

if a specific node k is next node (N), above condition will be

$$dS\text{-}D = dS\text{-}N + dN\text{-}D \qquad (0)$$

Where d is the number of hops between two nodes, S is source node, D is destination node and N is next node.

In this paper, P(S, D) is included all of nodes between two nodes S, D that are necessary for unicast deterministic routing. Also main route is the route that source node moves to marked destination node.

2.2. The proposed multicast routing

In proposed multicast routing (that is based on minimum distance between nodes of source and destination); a destination is selected randomly (D_0) and marked. Message routes as unicast to deliver it to the marked destination (D_0). At each hop, if condition (0) for N (that is next node in current message) and D_P (that is one of destination nodes except marked destination node in current message) is true, the next node in main route belongs to P(S, D_P). Therefore, message is not duplicated and routing is continued with a message. Otherwise, next node in main route with next node in P(S, D_P) is different and for routing D_P, message should be duplicated. Therefore, a necessary condition to duplicate the message is below condition.

$$dS\text{-}DP \neq dS\text{-}N + dN\text{-}DP \qquad (1)$$

Where d is number of hops between two nodes, S is source node, D_P is one of destination nodes (except the marked destination node in current message) and N is next node in current message.

Condition(1) is not sufficient to duplicate the message because when it is true for a specific destination D_P, for next steps, will remain true and message will duplicate at each hop to routing D_P frequently but for routing each destination node ,only one message is needed. Therefore, condition (2) (that prevents to copy the repeated message) is necessary.

$$dS\text{-}DP = dS\text{-}C + dC\text{-}DP \qquad (2)$$

Where d is number of hops between two nodes, S is source node, D_P is one of destination nodes (except the marked destination node in current message) and C is current node in current message.

Conditions (1) and (2) check whose next node and current node in main route belongs to P(S, D_P). Actually, these two conditions determine last common node between the main route and P(S, D_P) to duplicate the message.

If condition (1) and condition (2) for D_P are true simultaneously, the message should duplicate to route D_P (and D_P is marked). In other words, the message is duplicated if and only if, the current node between two routes is common and the next node is different in same two routes. Above steps for all messages into a network perform.

2.3 Giving an example

As an example, we suppose that node (3, 0) has message for nodes (5, 4), (7, 5) and (1, 1) as Fig. 1.

For unicast routing we have:

(3,0)→(3,1)→(3,2)→(3,4)→(6,4)→(5,4)

(3,0)→(3,1)→(3,2)→(3,5)→(7,5)

(3,0)→(3,1)→(6,1)→(4,1)→(1,1)

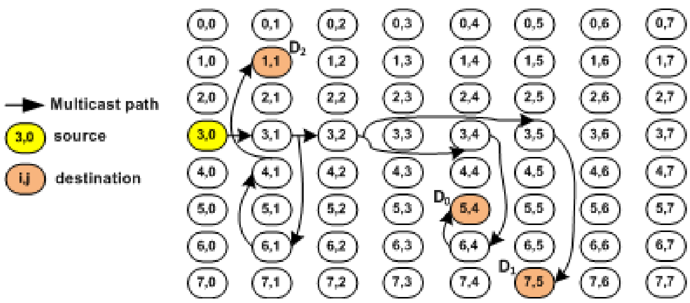

Fig. 1.Tree-based multicast algorithm based on unicast XY routing

For multicasting, one destination is selected randomly (D_0:(5,4)) and message is routed as unicast toward D_0. At each hop, two above conditions (1), (2) for the remaining destinations ((7, 5), (1, 1)) are checked and if both of them are true, message will be duplicated.

In first step, condition (1) is false for both of destinations (7, 5), (1, 1) but condition (2) is true for them. Therefore, the next node (3, 1) in first path is shared with two other paths.

S=(3,0) ,C=(3,0), N=(3,1) ,D={(7,5),(1,1)}

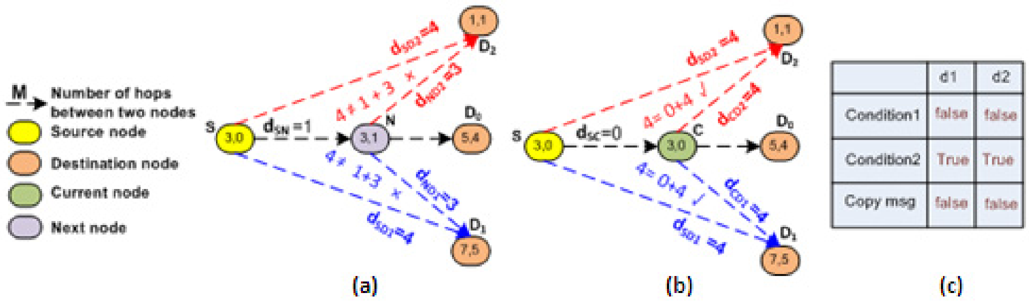

Fig. 2. first hop, checking (a) the dSD i ≠ dSN + dNDi (condition 1) , (b) dSD i = dSC + dCD i (condition 2) (c) Table

In second step, condition (2), (1) are true for destination (1, 1). Thus, message will be duplicated to route (1, 1) and in this new message destination node (1, 1) will be marked. However, for destination (7, 5), according to condition (1) that is false, message will not duplicate.

S=(3,0) ,C=(3,1), N=(3,2) , D={(7,5),(1,1)}

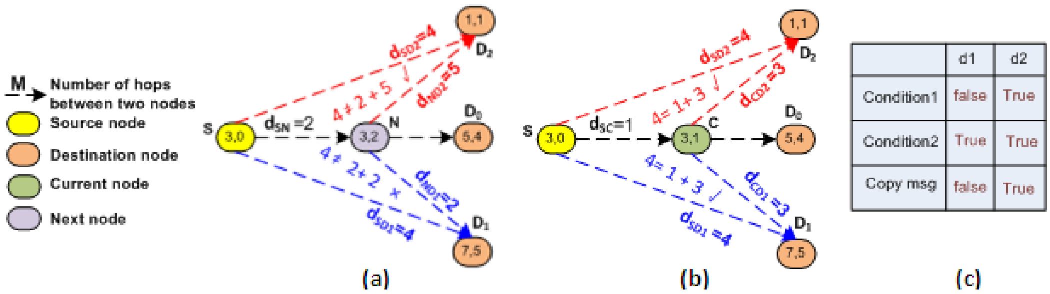

Fig. 3. second hop,checking (a) the dSD i ≠ dSN + dNDi (condition 1) , (b) dSD i = dSC + dCD i (condition 2) (c) Table

In third step, condition (1), (2) are true for destination (7, 5) and message will be duplicated to route it and (7, 5) will be marked in this new message. Even so condition (2) for destination (1, 1) is false. Therefore, message will not copy.

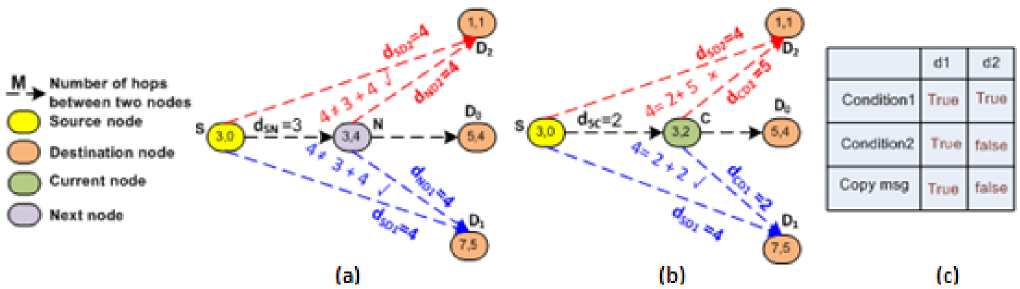

Fig. 4. third hop, checking (a) the dSD i ≠ dSN + dNDi (condition 1) , (b) dSD i = dSC + dCD i (condition 2) (c)Table

After this step (when the number of destinations equal to number of messages), at all next steps, condition (2) for all destinations will be false and no message will be duplicated. Also, routing will be as unicast. Pseudo code is as Fig. 5.

```
Define:
Distance (source, current node, next node, destination):
 Calculates the number of hops between nodes of source-destination (d_SD), current -destination (d_CD), source-current  (d_SC) and next node-destination(d_ND).
Next node: it is after current node
des: set of destinations
Input:  multicast set : <source,{D_0,D_1,D_2,…..,D_n}>
Created messages=0
While (des ≠ Ø)
select a destination from des randomly and mark it.
  Created messages++
    While (created messages > 0)
            If (current node == any one des)
                    des=des-{marked destination in current message}
            Created messages - -
                    End if
                    If (created messages==0)
                    select a destination from des randomly and mark it
              Created messages++
                    End If
              For all created messages
                    For all destinations (D_p) ∈ des-{marked destination in current message}
           //according to current and destination nodes in message, next node is available.
              distance (source (S), current node (C), next node (N), D_p)
                        If ((d_S-Dp <> d_S-N + d_N-Dp) && (d_S-Dp =  d_S-C + d_C-Dp))          //d_S-N=d_S-C+1
                        copy a message for routing D_p
                        Mark D_p
                    Created messages++;
                        Route as unicast from current node to D_p(marked destination)
                        End If
              End For
           End For
    End While
End While
```

Fig. 5. Multicast Pseudo code

3. Simulation results:

To evaluate the performance of suggested routing, we develop a discrete event simulator operating at the flit level using xmulator [8]. We set the networks link width to 128 bytes. Each link has the same bandwidth and one flit transmission is allowed on a link. The power is calculated based on a NoC with 65 nm technology whose routers operate at 2.5 GHz. We set the width of the IP cores to 1 mm, and the length of each wire is set based on the number of cores it passes. The number of virtual channels is two and maximum of simulation events are 15000000. The simulation results are obtained for 8×8 de Bruijn NoCs with XY routing algorithm, using the routing algorithms described in the previous section. The message length is assumed to be 32 and 64 flits and one virtual channel per physical channel is used. Messages are generated according to a Poisson distribution with rate λ. The traffic pattern can be *Uniform* and *Hotspot* [9].

With introducing this simulator and checking the advantages of this simulator, the delay and power figures are the following. The x axis of these figures indicates the generation rate and y axis indicates power and delay in our simulations.

Fig. 6 compares the average message latency for different traffic patterns with different message lengths of 32 and 64. As can be seen, the multicast routing has smaller average message latency with respect to the unicast routing algorithm for the full range of network load under various traffic patterns (especially in uniform traffic). For hotspot traffic load a hotspot rate of 16% is assumed (i.e. each node sends 16% of messages to the hotspot node (node (7, 7)) and the rest of messages to other nodes uniformly). Fig. 7 demonstrates power consumption of the multicast routing and unicast routing with various traffic patterns. It is again the multicast routing that shows a better behavior before reaching to the saturation point. In Fig. 8(a), the average message latency is plotted as a function of message generation rate at each node for the multicast routing and unicast routing for different message lengths of 32 and 64.

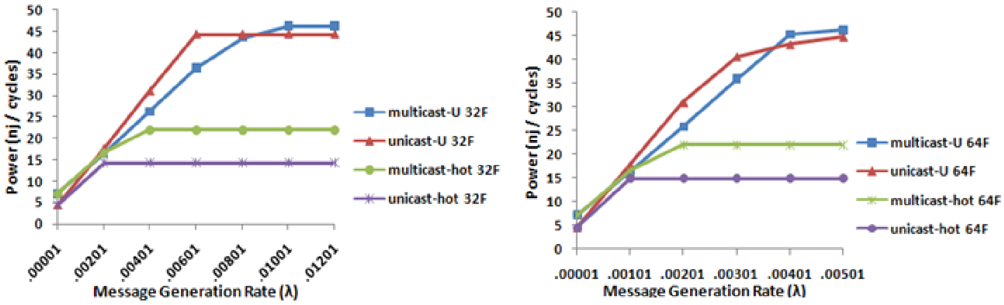

Fig. 6 compares the latency for different traffic patterns with different message lengths of 32 and 64.

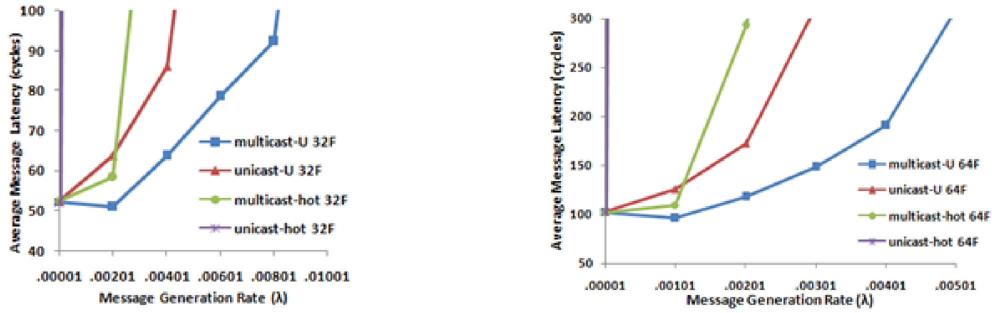

Fig. 7 demonstrates power consumption for various traffic patterns and message lengths of 32 and 64.

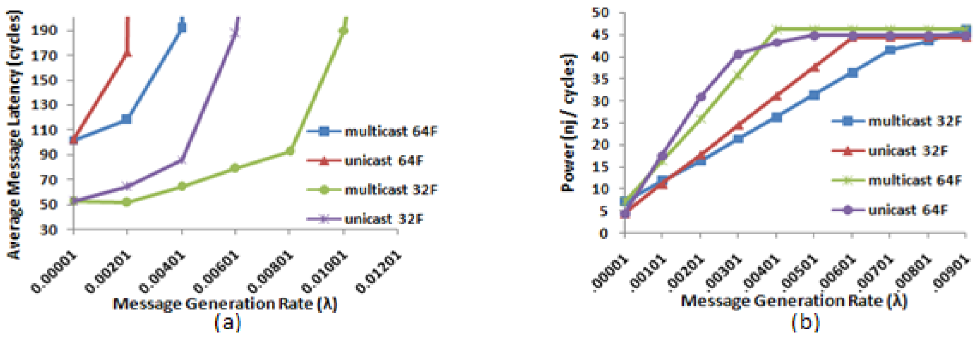

Fig. 8 compares the performance and total network power in different message lengths of 32 and 64.

As can be seen in the Fig. 8(a), the multicast routing has smaller average message latency with respect to the unicast routing. Fig. 8(b) compares the total network power in different message lengths of 32 and 64. The obtained result of xmulator indicates the multicast routing relative to unicast routing goes to the saturation later and can send more packages.

4. Conclusion

This paper proposes multicast routing for two-dimensional de Bruijn NoCs. Simulation experiments were conducted to assess the network latency and power consumption of the proposed routing. Results showed that the proposed routing improved the performance and power consumption of the NoC in comparison with unicast routing under traffic loads in hot spot and uniform traffics with various message lengths.

In the future work, we will try to merge the fault tolerant with multicast routing algorithm. Furthermore, we will use multicast for other digraphs networks such as shuffle-exchange networks [10].

5. References

[1] Benini L, Micheli GD. Networks on Chip: A New Paradigm for Systems on Chip Design. Design Automation and Test in Europe 2002; 418–419.

[2] Pande PP, Grecu C, Jones M, Ivanov A, Saleh R. Performance Evaluation and Design Trade-Offs for Network-on-Chip Interconnect Architectures. IEEE Transactions on Computers 2005; 54:1025-1040.

[3] Salminen E, Kulmala A, Hämäläinen TD. Survey of Network-on-Chip Proposals. OCP-IP white paper 2008.

[4] Flores A, Aragon JL, Acacio ME. An energy consumption characterization of on-chip interconnection networks

for tiled CMP architectures. The Journal of Supercomputing 2008; 45:341-364.

[5] Sabbaghi-Nadooshan R, Modarressi M, Sarbazi-Azad H. The 2d DBM: An Attractive Alternative to the Mesh Topology for Network-on-Chip. IEEE International conference on Computer Design 2008;486-490.

[6] Sabbaghi-Nadooshan R, Modarressi M, Sarbazi-Azad H. The 2D digraph-based NoCs: attractive alternatives to the 2D mesh NoCs. Journal of supercomputing 2012;49:1-21.

[7] Sabbaghi-Nadooshan R, Sarbazi-Azad H. The Kautz mesh: A New Topology for SoCs. International SoC Design Conference 2008; 300-303.

[8] Nayebi A, Meraji S, Shamaei A, Sarbazi-Azad H. "Xmulator: A listener-Based Integrated Simulation Platform for Interconnection Networks," Proc. of Asian Int. Con. on Mod. Sim., 2007, pp. 128-132.

[9] Duato J, Yalamanchili S, Ni LM. Interconnection Networks. Morgan Kaufman; 2003.

[10] Sabbaghi-Nadooshan R, Modarressi M, Sarbazi-Azad H. 2D SEM: A novel high-performance and low-power mesh-bases topology for networks-on-chip. International Journal of Parallel ,Emergent and Distributed Systems 2010;25:331-344.