Exploring Congestion-Aware Methods for Distributing Traffic in On-Chip Networks

Masoumeh Ebrahimi, Masoud Daneshtalab, Pasi Liljeberg, Juha Plosila, and Hannu Tenhunen

{Masoumeh.Ebrahimi,Masoud.Daneshtalab,Pasi.Liljeberg, Juha.Plosila,Hannu.Tenhunen}@utu.fi

Abstract. The performance of NoC is highly affected by the network congestion condition. Congestion in the network can increase the delay of packets to be routed between sources and destinations, so it should be avoided. The routing decision can be based on local or non-local congestion information. Methods based on local congestion condition are generally simple but they are unable to balance the traffic load efficiently. On the other hand, methods using non-local congestion information are more complex while providing better distribution of traffic over the network. In this paper, we explored several proposed locally and non-locally congestion-aware methods. Then we discussed about their advantages and disadvantages. Finally, we compared the methods with each other regarding the latency metric.

Keywords: Networks-on-Chip, Congestion, Adaptive Routing Algorithms.

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 [1]. In these chips, 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 (MPSoCs) [1]. By increasing the number of processing elements in a single chip, the traditional bus-based architectures in MPSoCs are not useful anymore and new communication infrastructure is needed. Network-on-Chip (NoC) has been addressed as a solution for the communication requirement of MPSoCs [1][3][4][5]. The performance and efficiency of NoC largely depend on the underlying routing technique which decides the direction a packet should be sent [6].

Routing algorithms are used in NoCs in order to determine the path of a packet from a source to a destination. Routing algorithms are classified as deterministic and adaptive algorithms. 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 [7][8]. The simplest deterministic routing method is dimension-order routing which is known as XY or YX algorithm. The dimension-order routing algorithms route packets by crossing dimensions in strictly increasing order, reducing to zero the

P. Pichappan, H. Ahmadi, and E. Ariwa (Eds.): INCT 2011, CCIS 241, pp. 319–327, 2011.

offset in one direction before routing in the next one. Adaptive routing has been used in interconnection networks to improve network performance and to tolerate link or router failure. In adaptive routing algorithms, the path a packet travels from a source to a destination is determined by the network condition. So they can decrease the probability of routing packets through congested or faulty regions.

In this paper, we have investigated different well-known congestion-aware routing methods. The routing selections policies in some of them are based on local congestion information while for the rest of them are based on non-local congestion information. We discussed about the advantages and disadvantages of each method and their effect on routing decision and balancing the traffic load. In order to compare the efficiency of methods in term of latency, we have measured the packets delay in each method using uniform and hotspot traffic.

This paper is organized as follows. In Section II, different congestion-aware routing methods named DyXY, EDXY, NoP, and CAS are explained and discussed. The results are reported in Section III while the summary and conclusion are given in the last section.

2 Congestion-Aware Routing Algorithms

2D-mesh topology is a popular architecture for NoC design due to its simple structure, ease of implementation, and support for reuse [10]. The performance and efficiency of NoCs largely depend on the underlying routing methodology. Adaptive routing algorithms can be decomposed into routing and selection functions [9]. The routing function supplies a set of output channels based on the current and destination nodes. The selection function selects an output channel from the set of channels supplied by the routing function [10]. The selection function can be classified as either congestion-oblivious or congestion-aware schemes [9]. In congestion-oblivious algorithms, such as Zigzag [12] and random [13], routing decisions are independent of the congestion condition of the network. This policy may disrupt the load balance since the network status is not considered.

3 Dynamic XY (DyXY)

An adaptive deadlock free routing algorithm called Dynamic XY (DyXY) has been proposed in [14]. In this algorithm, which is based on the static XY algorithm, a packet is sent either to the X or Y direction depending on the congestion condition. It uses local information which is the current queue length of the corresponding input port in the neighboring routers to decide on the next hop. It is assumed that the collection of these local decisions should lead to a near-optimal path from the source to the destination. The main weakness of DyXY is that the use of the local information in making routing decision could forward the packet in a path which has congestion in the routers farther than the current neighbors. This situation could happen when the routing unit is one unit apart from the destination in X or Y dimension. Such non-optimal routing decisions increase the network latency in NoC.

Fig. 1 shows an example of DyXY method where the routing decision based on local congestion information leads to deliver a packet through congested area. In this example the nodes 0 and 15 are the source and destination of the packet, respectively. In the DyXY method, the source node 0 compares the occupied slots of the west buffer at node 1 and that of south buffer at node 4. Since the node 1 is less congested, the packet is sent to this node. When the packet arrives at node 1, it has to be delivered through nodes 2 or 5. According to the congestion condition shown in Fig. 1, the node 2 is less congested and thus the packet is delivered to node 2. At node 2, the packet has to pass through the most congested area (i.e. nodes 6, 7, 10, and 11) in the network to reach the destination node. As a result, the congested path is selected since the decision is made based on the local information; the packet could pass through less congested area (i.e. nodes 8, 9, 12, and 13) if non-local information is considered.

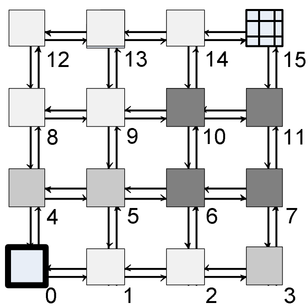

Fig. 1. An example of the DyXY method

4 Enhanced Dynamic XY (EDXY)

In EDXY [15], a wire is propagated along each row and column to carry the congestion information of the corresponding input buffers of the nodes. This information is propagated to the nodes in the adjacent row or column. In this way, each node in the network can be informed about the congestion condition of the nodes along the adjacent rows or columns.

In this method, every router first looks at the destination address of the packet. If the destination node is not located in the adjacent row or column, the packets are routed similar to DyXY method. However, if the destination address is just one hop apart from the router in either the X or Y direction, not only the queue length of the buffer in neighboring routers are considered, but also the congestion wire (based on the position of the destination) is used for routing.

An example is shown in Fig. 2 where a packet is delivered from the source node 0 to destination 15 and it is already at node 2. Based on DyXY method, since the node 3 is less congested than the node 6, the node 3 is selected as the next hop. However, by this decision the packet has to pass nodes 7 and 11 which are highly congested. In contrast, in EDXY method, in a similar situation (i.e. when a packet is located one hop away from the destination row or column and the neighboring nodes are not highly congested), the congestion conditions of the third and fourth columns are compared to each other; since the third column is not highly congested, the packet is sent through it and thus avoiding packets to be routed via highly congested nodes (i.e. nodes 7 and 11).

In a similar example as Fig. 1, when the nodes 6, 7, 10, and 11 are congested, at node 1, the EDXY method also sends packet to the X-direction and thus packets have to be routed through congested region due to the lack of global congestion information.

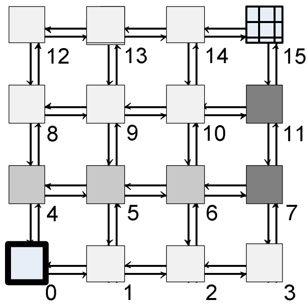

Fig. 2. An example of the EDXY method

5 Neighbor-on-Path (NoP)

In [16] the locality decision is extended to 2-hop neighbors. An example of the NoP method is shown in Fig. 3 where a packet is sent from source node 0 to destination 15. At source node 0, the packet can be sent either to node 1 or node 4. Based on NoP, the congestion value in the X direction is computed by considering the free buffer slots at the west input buffer of node 2 and south input buffer of node 5 (i.e. these nodes are located in the routing path to the destination). Similarly, the congestion value in the Y direction is measured by using the number of free buffer slots at the south input ports of node 8 and west input port of node 5. By comparing the obtained values in two directions, a packet is sent to node 1 or node 4. One of the shortcomings of this method is that the number of free buffer slots at the south and west input ports of node 5 is largely affected by the contention at north and east output ports. In other word, the congestion information of the corresponding input ports of node 5 is included in the congestion value of both X and Y directions. Since the congestion values at X and Y directions are compared with each other, the congestion status of node 5 cannot affect the routing decision. Moreover, NoP method suffers from the recursive nature of the routing algorithm, resulting in increased hardware overhead and router complexity. This method cannot be extended to look at the congestion of 3-hop neighbors due to the nonlinearly increased hardware overhead.

Following the example of Fig. 3, the packet is sent to the node 1 since the congestion status of node 2 is less than node 8. At node 1, the packet is sent to node 2 as the node 3 is less congested than node 9. As a result the packet has to pass through the highly congested area.

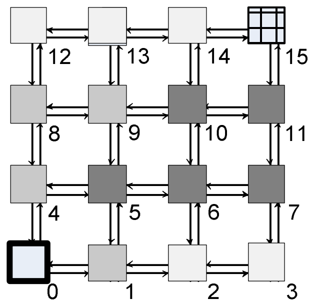

Fig. 3. An example of the NoP method

6 Agent-based Network-on-Chip (ANoC) along with Congestion-Aware Selection method (CAS)

In the Agent-based Network-on-Chip (ANoC) structure [17], the network is divided into several clusters in which a cluster includes a number of routers and a cluster agent. The design consists of two separate mesh networks: main data network and lightweight congestion network. The main data network connects the routers to each other to propagate packets over the network; while in the congestion network, cluster agents are communicated with each other to spread the congestion information. Each cluster agent performs two simple tasks. First, it collects the congestion information from the attached routers (local routers) and distributes the information to the neighboring cluster agents as well as the local routers; second, it forwards the received congestion information from the adjacent cluster agents to the local routers.

By distributing congestion information over the network, routing decision can be assisted by the local and non-local congestion information received from different regions of the network.

Depending on the relative position of the source and destination nodes, the Congestion-Aware Selection (CAS) method can be described in two parts as follow:

*1) **The source and destination cluster agents are located in the same agent-row or agent-column***

The congestion value for one output channel is calculated using the weighted sum of the 1-hop, 2-hop and 3-hop neighboring nodes. These nodes must be located in the minimal path and in the same network-row (network-column) as the source node.

Consider an example in Fig. 4 where the node 0 wants to communicate with the node 7. As can be seen in this figure, the nodes 0 and 7 are connected in the first row of the congestion network. The node 0 has to choose whether to send a packet to the node 1 or node 4. The congestion value at the X direction is computed by considering the congestion values of nodes 1, 2, and 3 while the congestion value at the Y direction is calculated by using the congestion statues of nodes 4, 5, and 6.

To put more emphasis on the congestion condition of nearby nodes, the higher weights are assigned to the closer nodes. In the CAS method, the weight of 3, 2 and 1 is given to the 1-hop, 2-hop and 3-hop neighbors, respectively.

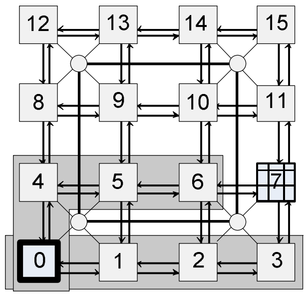

Fig. 4. An example of CAS method when source and destination are in the same row

2) Source and destination are not located in the same agent-row or agent-column

In this case, the congestion value for each selected output channel is provided by the values of the adjacent node and the neighboring cluster. To place emphasize on the local congestion values more than non-local information, the neighboring nodes are assigned the weight of 3 while the congestion value of the adjacent clusters are given the weight of 2.

An example is shown in Fig. 5 where node 0 sends a message to the node 15. For the X direction, the congestion value is calculated by the weighting sum of the congestion values of the node 1 and the cluster 1, while for the other output channel, the congestion value of the node 4 is combined with the congestion value of the cluster 2.

The routing decision in this method is better than the other proposed methods. However, the structure of the congestion network is changed depending on whether the network dimensions are even or odd.

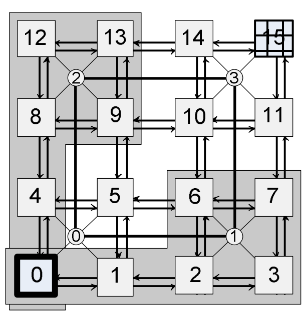

Fig. 5. An example of CAS method when source and destination are in different rows

7 Experimental Results

To compare the efficiency of the methods, a 2D-NoC simulator is implemented with VHDL to model all major components of the NoC. Simulations are carried out to determine the latency-throughput characteristics of each network. For all the routers, the data width was set to 32 bits. Each input virtual channel has a buffer (FIFO) 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. In simulations, the latency is measured by averaging the latency of the packets when each local core generates 3000 packets. As a performance metric, we use latency defined as the number of cycles between the initiation of a message operation issued by a Processing Element (PE) and the time when the message is completely delivered to the destination PE. The request rate is defined as the ratio of the successful message injections into the network interface over the total number of injection attempts. For all routers, the frequency is set to 1GHz and the packet size is set to 5 flits.

8 Uniform Traffic Profile

In the uniform traffic profile, each processing element (PE) generates data packets and sends them to another PE using a uniform distribution [18][19][20]. The mesh sizes are considered to be 8×8 and 14×14. In Fig. 6, the average communication delay as a function of the average packet injection rate is plotted for both mesh sizes. As observed from the results, CAS leads to the lowest latency, and then DyXY, EDXY, and NoP. This was expected due to the distribution of traffic over less congested areas. Because of the ANoC structure (along with CAS method), each router can observe the congestion information of not only the neighboring routers, but also the routers residing beyond the neighboring routers.

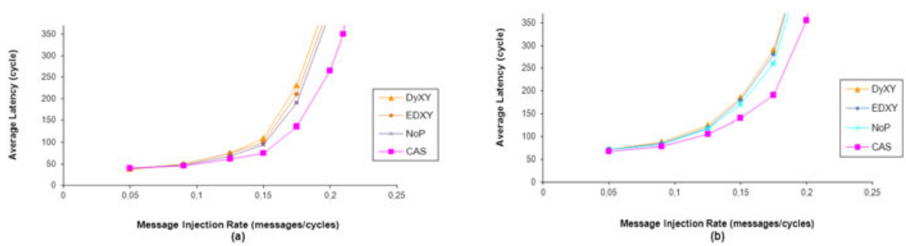

Fig. 6. Performance under different loads in (a) 8×8 2D-mesh and (b) 14×14 2D-mesh under uniform traffic model

9 Hotspot Traffic Profile

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 simulations, given a hotspot percentage of H, a newly generated message is directed to each hotspot node with an additional H percent probability. We simulate the hotspot traffic with a single hotspot node at (4, 4) and (7, 7) in the 8×8 and 14×14 2D-meshes, respectively. The performance of each network with H = 10% is illustrated in Fig. 7. As observed from the figure, the CAS method achieves better performance compared to those of the other schemes.

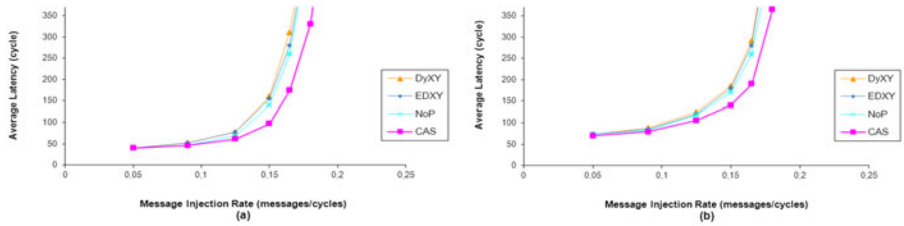

Fig. 7. Performance under different loads in (a) 8×8 2D-mesh and (b) 14×14 2D-mesh under hotspot traffic model with H=10%

10 Summary and Conclusion

In this paper, we have explained and investigated several congestion-aware routing methods in the realm of NoC. Among them, the decision making in the DyXY and EDXY methods are based on local congestion information; while the NoP and CAS methods consider not only the local information of the neighboring routers but also non-local congestion statuses of the nodes that are beyond the neighboring routers. We discussed about the advantages and disadvantages of each method and finally we compared the methods with each other in term of latency.

References

[1] Xu, J., Wolf, W., Hankel, J., Charkdhar, S.: A Methodology for design, modeling and analysis for networks-on-Chip. In: IEEE International Symposium on Circuits and Systems, pp. 1778–1781 (2005)
[2] Cesariow, O., Lyonnard, L., Nicolescu, G., et al.: Multiprocessor SoC platforms: a compo-nent-based design approach. In: Proc. Int. Conf. IEEE Design and Test of Computers, pp. 52–63 (2002)
[3] Towles, B., Dally, W.: Route packets, not wires: on-chip interconnection networks. In: Proc. DAC (2001)
[4] Benini, L., De Micheli, G.: Networks on chips: a new SoC paradigm. IEEE Computer (January 2002)
[5] Bertozzi, D., Benini, L.: Xpipes: A Network-on-Chip Architecture for Gigascale Systems-on-Chip. IEEE Circuits and Systems Magazine 2, 18–31 (2004)
[6] Wu, D., Al-Hashimi, B.M., Schmitz, M.T.: Improving Routing Efficiency for Network-on-Chip through Contention-Aware Input Selection. In: Proc. of ASP-DAC, pp. 36–41 (2006)
[7] Bertsekas, D., Gallager, R.: Data Networks. Prentice Hall (1992)

[8] Dally, W.J., Towles, B.: Principles and Practices of Interconnection Networks. Morgan Kaufmann (2004)
[9] Duato, J., Yalamanchili, S., Ni, L.: Interconnection Networks: An Engineering Approach. Morgan Kaufmann (2002)
[10] Liang, J., Swaminathan, S., Tessier, R.: Asoc: a scalable, single-chip communication architectures. In: IEEE Int. Conf. on PACT, pp. 37–46 (October 2000)
[11] Gratz, P., Grot, B., Keckler, S.W.: Regional Congestion Awareness for Load Balance in Networks-on-Chip. In: Proc. HPCA, pp. 203–214 (2008)
[12] Badr, H.G., Podar, S.: An optimal shortest-path routing policy for network computers with regular mesh-connected topologies 38(10), 1362–1371 (1989)
[13] Feng, W., Shin, K.G.: Impact of Selection Functions on Routing Algorithm Performance in Multicomputer Networks. In: International Conference on Supercomputing, pp. 132–139 (1997)
[14] Li, M., Zeng, Q., Jone, W.: 'DyXY - a proximity congestion-aware deadlock-free dynamic routing method for network on chip. In: Proc. DAC, pp. 849–852 (2006)
[15] Lotfi-Kamran, P., et al.: EDXY - A low cost congestion-aware routing algorithm for network-on-chips. Journal of Systems Architecture 56(7) (2010)
[16] Ascia, G., Catania, V., Palesi, M.: Implementation and Analysis of a New Selection Strategy for Adaptive Routing in Networks-on-Chip. IEEE Transaction on Computers 57(6), 809–820 (2008)
[17] Ebrahimi, M., Daneshtalab, M., Liljeberg, P., Plosila, J., Tenhunen, H.: Agent-based On-Chip Network Using Efficient Selection Method. In: Proceedings of 19th IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC), pp. 110–115 (2011)
[18] Boppana, R.V., Chalasani, S.: A Comparison of Adaptive Wormhole Routing Algorithms. In: Proc. Int. Symp. Computer Architecture, pp. 351–360 (May 1993)
[19] Flugham, M.L., Snyder, L.: Performance of Chaos and Oblivious Routers under Non-Uniform Traffic., Technical Report UW-CSE-93-06-01 (July 1993)
[20] Glass, C.J., Ni, L.M.: The Turn Model for Adaptive Routing. In: Proc. Symp. Computer Architecture, pp. 278–287 (May 1992)