Efficient Congestion-Aware Selection Method for On-Chip Networks

Masoumeh Ebrahimi, Masoud Daneshtalab, Pasi Liljeberg, Juha Plosila, Hannu Tenhunen
Department of Information Technology, University of Turku, Finland
{ masdan, masebr, pakrli, juplos, hanten }@utu.fi

Abstract- the choice of routing algorithm can have a large impact on the performance of on-chip networks. As adaptive routing algorithms may return a set of output channels, a selection method (routing policy) is employed to choose the appropriate output channel from the given set. In this paper, we present a novel on-chip network structure to detect the local and non-local congested areas. Based on the presented structure, an efficient congestion-aware selection method is proposed to choose an output channel that allows a packet to be routed through a less congested area.

I. INTRODUCTION

Network-on-Chip (NoC) has been addressed as a solution for the communication requirements of MPSoCs [1][2]. The performance and efficiency of NoC largely depend on the underlying routing technique which decides the direction where a packet should be sent [2][4][5]. Adaptive routing algorithms can be decomposed into routing and selection functions. The routing function supplies a set of output channels for delivering packets based on the positions of current and destination nodes. The selection function chooses an output channel from the set of channels given by the routing function [2]. The selection function can be classified as either congestion-oblivious or congestion-aware schemes [6]. In congestion-oblivious algorithms, 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. Unlike congestion-oblivious methods, in congestion-aware algorithms, such as Dynamic XY (DyXY) [7], Neighbors-on-Path (NoP) [6], and BARP [8] the selection is usually performed using the congestion status of the network [2]. Most of congestion-aware algorithms consider local traffic condition in which each router analyses the congestion condition 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. Therefore, routing algorithms based on local congestion information are efficient only when the traffic is mostly local. In this paper, an Agent-based Network-on-Chip (ANoC) method is proposed to determine the congested areas in the network and route packets through the less congested areas based on the local/non-local congestion information.

This paper is organized as follows. In Section II, the proposed Agent-based Network-on-Chip is described; In Section III, the congestion-aware selection method is explained, while the results are discussed in Section IV and the conclusion is given in the last section.

II. AGENT-BASED NETWORK-ON-CHIP

In the Agent-based Network-on-Chip (ANoC) approach, at first, the congestion information is distributed over the network and then the congestion-aware selection method utilizes this local and non-local congestion information to route packets through the less congested areas. Fig. **1** shows the Agent-based Network-on-Chip structure where the network is divided into several clusters; each includes a number of routers and a cluster agent. The design consists of two separate mesh networks: actual data network and lightweight agent network. The actual data network connects the routers to each other to propagate packets over the network; while in the agent 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. Accordingly, each router is aware of the congestion condition of the routers located in its local and neighboring clusters.

Fig. 2 illustrates the congestion detection circuit. To compute the Congestion Level (CL) of a router, the congestion status of

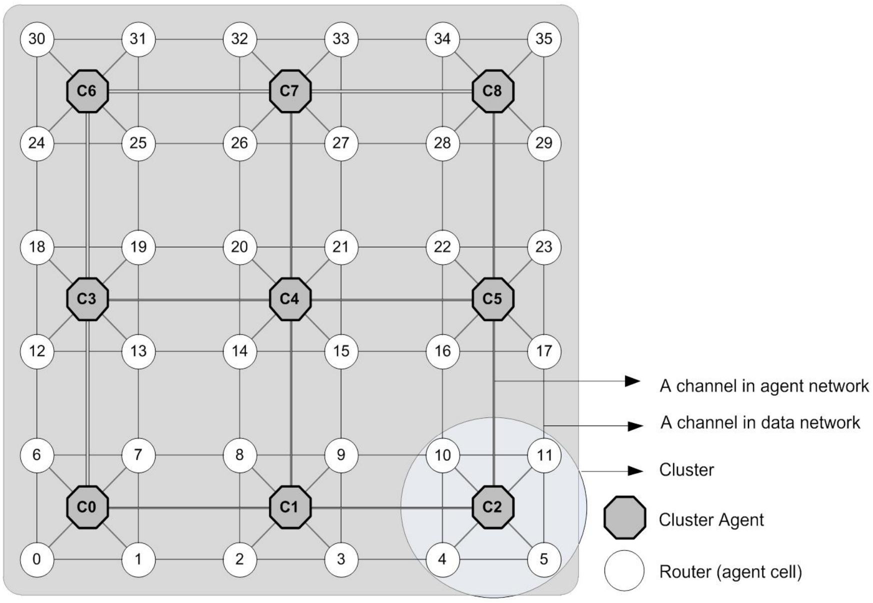

Fig. 1. Agent-based on-chip network.

the input buffers is required by the Agent Cell Unit. We use a signal named Congestion Status (CS) to indicate the buffer status. The CS signal of an input buffer is determined by a history-based scheme capturing the input buffer threshold signal. A 4-bit shift register is adopted to store the threshold signal whenever a new flit enters or leaves the buffer (flit events: flit_tx or flit_rx). That is, in each flit event, if the number of occupied cells of the buffer is larger (smaller) than a threshold value, the threshold signal is assigned to one (zero) and stored in the shift register. According to Fig. 2, the CS signal is asserted if all shift register bits are one. The CL value of a router is computed by summing up the CS signals received from the input buffers (e.g. local, east, west, north, and south ports when no virtual channel is used). In fact, the CL value of each router indicates its load level, e.g. if the east and local input buffers of a router are congested (local CS = 1 and east CS = 1), then the CL value of the router will be 2. Afterward, each cluster agent receives and combines the CL values of local routers and transfers the CL string to the neighboring cluster agents as well as to local routers. As the channel width of cluster agents is identical to the CL string, the string can be transferred within one clock cycle.

III. CONGESTION-AWARE SELECTION METHOD

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. As mentioned earlier, a cluster agent collects and concatenates congestion information of routers within its local routers and transfers the combined information back to the local routers and to the neighboring clusters. In this way, each router has the congestion information of several routers in the network. For an example, consider the node 14 in Fig. **1**. This node not only knows the congestion information of the routers within its cluster (i.e. routers 15, 20 and 21) but also have the information about all the routers in the clusters 1, 3, 5 and 7 (i.e. routers 2, 3, 8, 9, 12, 13, 18, 19, 16, 17, 22, 23, 26, 27, 32 and 33).

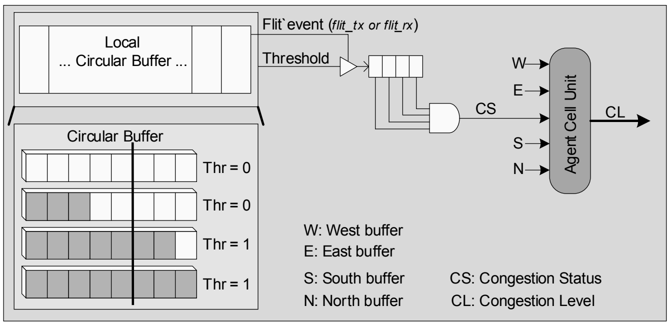

Fig. 2. Calculating congestion status and congestion level signals.

A number of different ways can be proposed to use the non-local information gathered by the Agent-based method in adaptive routing algorithms. For instance, an output port can be selected based on the congestion information of the neighboring clusters or/and the routers located in the minimal path. In this paper, we have used the non-local congestion information of the routers in 1-hop, 2-hop and 3-hop neighbors; and we have also considered the DyXY method as our base adaptive routing algorithm. We call this method Non-local DyXY.

In the DyXY method, a packet is sent to the X or Y direction depending on the congestion information of the neighboring routers. However, in the Non-local DyXY, the routing decision is made using the congestion information of up to 3-hop neighbors. To explain the idea, suppose a packet is sent from the router 18 to the router 4. So, the packet can be delivered to either the east or south port. The router 18 has the information of the routers in its own cluster and the clusters 0, 4 and 6. In the Non-local DyXY method, the congestion information of the routers 19, 20 and 21 are used for determining the congestion value in the east direction and similarly, the congestion information of the routers 12, 6 and 0 are utilized for the congestion value in the south direction.

To put more emphasis on the congestion condition of nearby routers, higher weights are assigned to closer nodes. In the Non-local DyXY method, different weights are assigned to the 1-hop, 2-hop and 3-hop neighbors. The congestion values in each direction (Cong_Dir) can be calculated by the following formula:

$$Cong\_Dir = (M) \times Cong_{1hop} + (M-1) \times Cong_{2hop} + (M-2) \times Cong_{3hop}$$
where $M = Min(\Delta x, \Delta y)$, $\Delta x = |x_d - x_s|$, $\Delta y = |y_d - y_s|$

Based on this formula, a packet is sent to an output channel which has the lowest Cong_Dir value.

IV. Experimental Results

To evaluate the efficiency of the proposed Agent-based NoC (ANoC) architecture, two other on-chip networks are also implemented. These on-chip networks, named NoC1 and NoC2, are formed by utilizing Dynamic XY (DyXY) [7] and Neighbors-on-Path (NoP) [6] approaches. For all the routers, the data width was set to 32 bits. Each input virtual channel has a buffer (FIFO) with the size of six flits. The congestion threshold value is set to four. 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. In order to know the real impact of the proposed network, we used traces [9] from some application benchmark suites selected from SPLASH-2 [10] and PARSEC [11]. We used the x264 application of PARSEC, and the Radix, Ocean and fft applications from SPALSH-2 for our simulation where the cache coherence protocol is MESI [12]. Fig. 3 shows the average packet latency across four benchmark traces, normalized to NoC1. ANoC provides lower latency than the other schemes, it shows the greatest performance gain on Ocean with 24% reduction in latency. The average performance gain of ANoC is up to 18% across all benchmarks vs. NoC1 and 10% vs. NoC2.

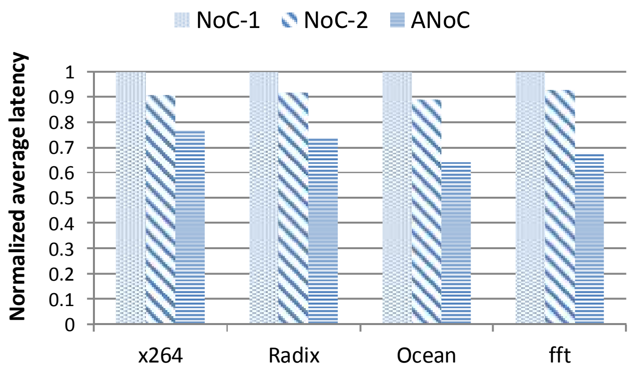

Fig. 3. Performance under different application benchmarks.

V. Conclusion

In this paper, we presented an Agent-based network-on-chip to diagnose the congested areas. An efficient selection method was presented for adaptive routing algorithms to choose the appropriate channel which avoids packets to be routed into congested areas. The results revealed that the proposed Agent-based strategy improves the performance significantly.

VI. Acknowledgment

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

References

[1] O. Cesariow, L. Lyonnard, G. Nicolescu, et al., "Multiprocessor SoC platforms: a component-based design approach", Proc. Int. Conf. IEEE Design and Test of Computers, pp. 52–63, France, 2002.

[2] J. Duato, S. Yalamanchili, L. Ni, “Interconnection Networks: An Engineering Approach”, Morgan Kaufmann, 2002.

[3] M. Daneshtalab, M. Ebrahimi, P. Liljeberg, Juha Plosila, and H. Tenhunen, “A Low-Latency and Memory-Efficient On-hip Network,” in Proceedings of 4th IEEE/ACM International Symposium on Network-on-Chip (NOCS), pp. 99-106, May 2010, France.

[4] M. Ebrahimi, M. Daneshtalab, P. Liljeberg, J. Plosila, and H. Tenhunen, "Exploring Partitioning Methods for 3D Networks-on-Chip Utilizing Adaptive Routing Model," in Proceedings of 5th ACM/IEEE International Symposium on Networks-on-Chip (NOCS), pp. 73-80, May 2011, USA.

[5] M. H. Neishaburi, Z. Zilic, “Reliability aware NoC router architecture using input channel buffer sharing”, in Proc. GLSVLSI, pp. 511-516, 2009.

[6] G. Ascia, V. Catania, M. Palesi, “Implementation and Analysis of a New Selection Strategy for Adaptive Routing in Networks-on-Chip”, IEEE Transaction on Computers, v.57, I.6, pp. 809 – 820, 2008.

[7] M. Li, Q. Zeng, W. Jone, “DyXY - a proximity congestion-aware deadlock-free dynamic routing method for network on chip”, Proc. DAC, pp. 849-852, 2006.

[8] P. Lotfi-Kamran, M. Daneshtalab, Z. Navabi, and C. Lucas, “BARP- A Dynamic Routing Protocol for Balanced Distribution of Traffic in NoCs to Avoid Congestion,” in Proceedings of 11th ACM/IEEE Design, Automation, and Test in Europe Conference (DATE), pp. 1408-1413, Mar 2008, Germany.

[9] M. Daneshtalab, M. Ebrahimi, T. C. Xu, P. Liljeberg, and H. Tenhunen, “A generic adaptive path-based routing method for MPSoCs,” Journal of Systems Architecture (JSA-elsevier), Vol. 57, No. 1, pp. 109-120, 2011.

[10] S. C. Woo, M. Ohara , E. Torrie, J. Pal Singh, A. Gupta, “The splash-2 programs: Characterization and methodological considerations”, in Proc. of ISCA, 1995.

[11] C. Bienia , S. Kumar , J. Pal Singh, K. Li, “The parsec benchmark suite: characterization and architectural implications”, in Proc. of PACT, 2008.

[12] A. Patel, K. Ghose, “Energy-efficient mesi cache coherence with pro-active snoop filtering for multicore microprocessors”, in Proc. of the 13th Int. Symp. on Low power electronics and design, pp. 247–252, 2008.