



HARAQ-Routing Table for Networks On-Chip Congestion Control

Dr.J.Kaliappan

Professor, Department of ECE

Sri Sai Institute of Technology and Science, Rayachoti

Email: kaliappanphd@gmail.com

ABSTRACT- In on-chip networks the occurrence of congestion can severely degrade the performance due to increased message latency. In mesh topology, minimal methods can propagate messages over two directions at each switch, but in high traffic loads minimal methods are inefficient. In this paper, we present an adaptive routing algorithm for on-chip networks that provide a wide range of alternative paths between each pair of source and destination switches. Initially, the algorithm determines all permitted turns in the network including 180-degree turns on a single channel without creating cycles. The implementation of the algorithm provides best usage of all allowable turns to route messages more adaptively in the network. On top of that, for selecting a less congested path, an optimized and scalable learning method is utilized. The learning method is based on local and global congestion information and can estimate the latency from each output channel to the destination region.

Keywords: *Networks-on-chip, Highly Adaptive Routing Algorithm, Latency, Congestion, Deadlock, Livelock.*

I. INTRODUCTION

Networks-on-Chip (NoC) has emerged as a solution to address the communication demands of future multicore architectures due to its reusability, scalability, and parallelism in communication infrastructure [2]. The performance and efficiency of NoCs largely depend on the underlying routing model which establishes a connection between input and output channels in a switch.

In this paper, we proposed a novel routing algorithm, named Highly Adaptive Routing Algorithm using Q-Learning (HARAQ) where the main contributions of the paper are summarized as follow:

1. A low-restrictive non-minimal algorithm to provide several alternative paths between each pair of source and destination switches.

The algorithm uses only an extra virtual channel in the Y dimension and provides a large number of paths for routing messages. Different turns are defined on each virtual channel, such that the prohibited turns in one virtual channel are permitted in the other one. Another subtle point of the presented method is its ability to enable 180-degree turns on a single channel without creating cycles.

2. An efficient output selection strategy for finding a low latency path from a source to a destination.

The presented routing scheme utilizes an optimized Q Learning model for the output selection function. In this way, the output selection can efficiently estimate the latency of a message to reach its destination through each of the possible output channels. This information is extracted from a Q-Table available at each switch. Unlike typical Q-Routing methods, our proposed model is scalable and the size of Q-Tables is relatively small.

II. RELATED WORK

Most common implementations of minimal routing algorithms, e.g. FRA [3], NoP [4], RCA [5], and DBAR [6], have focused on collecting local or non-local congestion information to get an estimation of the congested areas in the network. However, due to a low degree of adaptiveness, minimal routing algorithms cannot distribute the traffic over the network efficiently even if they have accurate knowledge of the network condition.

Virtual channels can be used to avoid deadlock and increase adaptiveness. DyXY [7] and mad-y are the methods using only two virtual channels along one of the two physical channels. Although they are fully adaptive in minimal paths, messages are limited in terms of routing options and thus the traffic load cannot be efficiently balanced over the network. Generally, non-minimal routing schemes have been proposed for tolerating faults rather than avoiding congestion [7]. Most of these models are proposed to support special cases of faults, such as one faulty switch, convex or concave regions, which require a large number of virtual channels to avoid deadlock. In fact, adding virtual channels is expensive because of additional buffers and complex control logics. There are other fault tolerant approaches [8, 9 and 10] which do not require any virtual channels. However, these algorithms are partially adaptive and very limited in supporting faults. A highly resilient routing algorithm is proposed in [11] to tolerate a large number of faults without exploiting any virtual channel, but only one path can be selected between each pair of source and destination nodes. In general, each method defines a new tradeoff between the number of virtual channels, the ability to handle different fault models, and the degree of adaptiveness.

Q-Routing based models have been studied in several literatures, but they have rarely been investigated in the context of on-chip networks. FTDR-H [12] utilizes Q Routing methods to tolerate faults and find a path between each pair of switches as long as a path exists. Moreover, the size of Q- Tables is reduced by taking advantages of the clustering model. This clustering model is also extensively discussed in the C-Routing method.

III. HIGHLY ADAPTIVE ROUTING ALGORITHM USING QLEARNING MODEL

The mad-y and the presented routing methods utilize a double-Y network where the X and Y dimensions have one and two virtual channels, respectively (Fig. 1). Each switch in the double-Y network has seven pairs of channels, i.e. East (E), West (W), North-vc1 (N1), North-vc2 (N2), South-vc1 (S1), South-vc2 (S2), and Local (L). In 2D mesh-based network, three types of turns can be taken: 0-degree, 90-degree, and 180-degree turns (U turns) [24]. By taking a 0-degree turn, a message transmits in a same direction with a possibility of switching between virtual channels. The turn is called 0-degree-ch if in a turn neither the direction nor the virtual channel changes (Fig. 1(b)) while it is a 0-degree-vc turn if the virtual channel changes (Fig. 1(c)). By taking a 90-degree turn, a message transmits between the switches in perpendicular directions (Fig. 1(d)). By taking a 180-degree turn, a message is transferred to a channel in the opposite direction [20]. If the virtual channel is changed, the turn is called 180-degree-vc (Fig. 1(e)); otherwise, it is represented as 180-degree-ch (Fig. 1(f)). Note that, in all figures, the vc1 and vc2 are differentiated by – and = respectively.

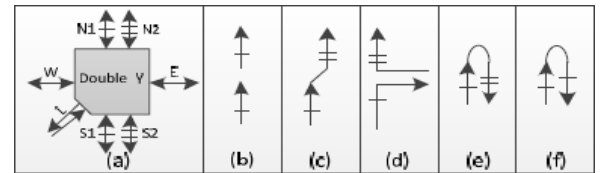


Fig. 1.(a) A switch in a double-Y network (b) 0-degree-vc (c) 0-degree-ch (d) 90-degree (e)180-degree-vc (f)180-degree-ch

A. Highly Adaptive Routing Algorithm

The aim of the proposed routing algorithm, named Highly Adaptive Routing Algorithm (HARA), is to enhance the capability of the existing virtual channels in mad-y to reroute messages around congested areas and hotspots. 180-degree-vc turns, those from vc1 to vc2 are taken in ascending order (Fig. 2(a), Fig. 2(b)), so that it is safe to employ them in the network. As all 180-degree-vc turns from vc2 to vc1 take place in descending order, so they cannot be used in the network (Fig. 2(c), Fig. 2(d)). Now, let us examine a 180-degree-ch turn when a message uses it in vc1 of the north direction (Fig. 2(e)). The label

on the output vc1 of the north direction is $(m-1-x, 1+y)$ and the label on the input vc1 of the north direction is $(m-1-x, n-1-y)$. The turn takes place in ascending order if and only if $n-1-y$ is greater than $1+y$. Therefore, this turn can be safely added to a set of allowable turns if the y coordinate of a switch is less than $(n-2)/2$. Similarly, in Fig. 2(f), 180-degree-ch turn on the vc2 of the north direction is permitted if the y value of a switch is less than $(n-2)/2$. 180-degree-ch turns on the vc1 and vc2 of the south direction are permitted (Fig. 2(g) and Fig. 2(h)) if and only if the y coordinate of a switch is greater than $n/2$. Finally, 180-degree-ch

turn on the west direction is always permitted (Fig. 2(i)) while 180-degree-ch turn on the east direction is prohibited in the network (Fig. 2(j)).

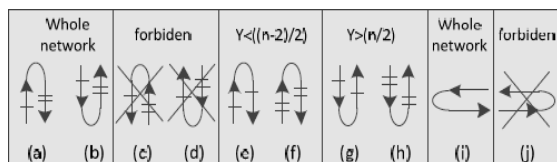


Fig.2. Allowable 180-degree turns in the HARA method

As shown in Fig.2, there are four conditional 180-degree turns. Two of those are allowable only in the northern part of the network and two others in the southern part of the network. This not only increases the complexity of the routing function but also imposes heterogeneous routing function for switches. To overcome this issue, we modify the numbering mechanism such that two turns are permitted in the whole network (Fig. 2(g) and Fig. 2(h)) and two other are prohibited (Fig. 2(e) and Fig. 2(f)). The numbering mechanism of HARA along with all permitted turns in the network. As can be observed from this figure, all allowable turns are taken in ascending order. All permissible output channels of HARA, for each pair of the input channel and destination positions are shown in Table1. HARA offers a large degree of adaptiveness to route messages. One of the drawbacks of non-minimal methods is the complexity of determining eligible output channels.

Theorem 1: HARA is deadlock-free.

Proof: As can be observed from Fig. 5, all connections between input channels and output channels to form eligible turns in HARA take place in ascending order and thus HARA is deadlock free.

Theorem 2: HARA is livelock-free

Proof: In HARA, whenever a message transmits in the east direction, it can never be routed back in the west direction. Therefore, in the worst case, the message may reach to the leftmost column and then starts moving in the east direction toward the destination column. Therefore, after a limited number of hops, the message reaches the destination, and Theorem 2 is proved.

Table1. Potential output channels according to the input channel and relative position of source of destination switch

Pos/ In Ch	N	S	E	W	NE	NW	SE	SW
L	N1,N2, S1,W	N1,S1,S2 ,W	N1,N2,S1,S2, E,W	N1,S1, W	N1,N2,S1,S2, E,W	N1,S1, W	N1,N2,S1,S2, E,W	N1,S1, W
N1	N2,S1, W	S1,S2,W	N2,S1,S2,E,W	S1,W	N2,S1,S2,E,W	S1,W	N2,S1,S2,E,W	S1,W
N2	-	S2	S2,E	-	S2,E	-	S2,E	-
S1	N1,N2, S1,W	N1,S1,S2 ,W	N1,N2,S1,S2, E,W	N1,S1, W	N1,N2,S1,S2, E,W	N1,S1, W	N1,N2,S1,S2, E,W	N1,S1, W
S2	N2	-	N2, E	-	N2, E	-	N2, E	-
E	N1,N2, S1,W	N1,N2, S1,W	N1,N2,S1,S2, E,W	N1,S1, W	N1,N2,S1,S2, E,W	N1,S1, W	N1,N2,S1,S2, E,W	N1,S1, W
W	N2	S2	N2,S2,E	-	N2,S2,E	-	N2,S2,E	-

InCh:Input Channel; OutCh:Output Channel
Pos:Destination Position
IF Pos={L} THEN
OutCh(L)<='1';
IF Pos={E or NE or SE} THEN
OutCh(E)<='1';
IF InCh={L or N1orS1or E} THEN
OutCh(W)<='1';
IF InCh={L or S1or E} THEN
OutCh(N1)<='1';
IF InCh={Lor N1orS1orE}THEN
OutCh(S1)<='1';
IF (InCh/={N2}) AND
(Pos={N orE or NEorSE})THEN
OutCh(N2)<='1';
IF (InCh/={S2}) AND
(Pos={S or E or NEorSE})THEN
OutCh(S2)<='1';

Fig.3. Determining all eligible output channels by HARA.

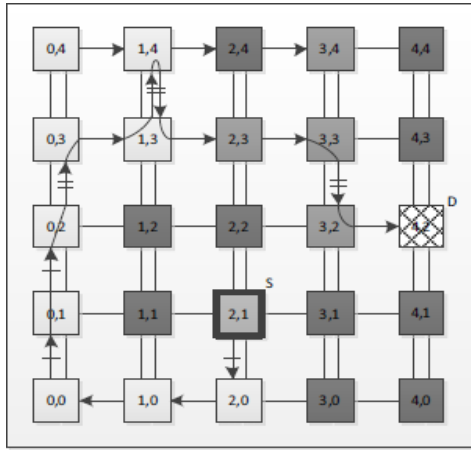


Fig.4. An example of HARA

Fig. 4 shows an example of the HARA method in a 5×5 mesh network in which the source switch at (2,1) sends a message to the destination switch at (4,2). According to Table 1, the message arriving from the local channel and delivering toward the destination in the northeast position has six alternative choices (i.e. N1, N2, S1, S2, E, and W); among them, the output channels N1, N2, and E introduce the minimal paths and S1, S2, and W indicate the non-minimal paths.

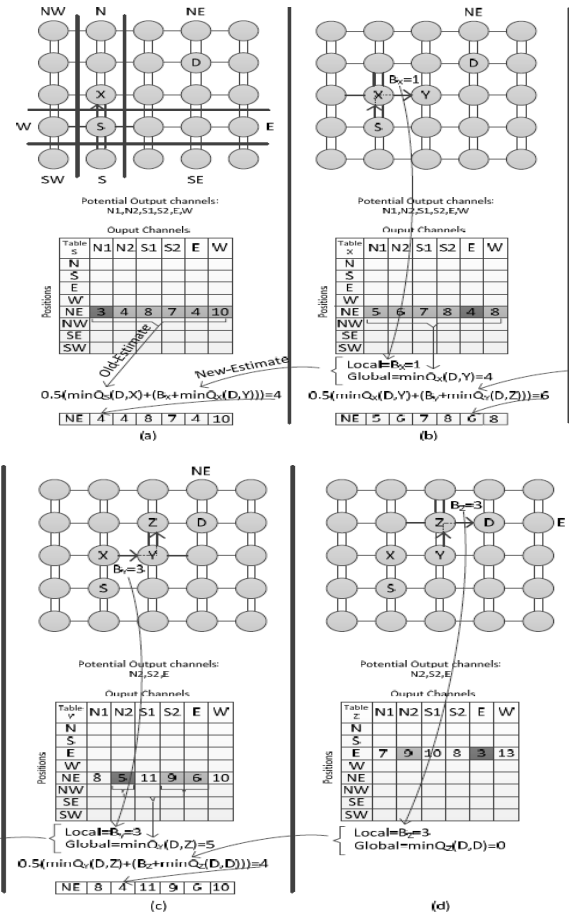


Fig.5. The process of updating Q-Table

B. Q-Learning-based Output Selection Function

As mentioned earlier, we utilize an optimized Q-Routing model for the selection function of HARA to estimate the latency of sending a message from each output channel to the destination switch. Let us explain the HARAQ using the example of Fig 5.

Transferring Local and Global Information: Q- Routing models are explored in on-chip networks but it is not obvious how the congestion is defined for local and global information and how many bits are considered for them. $Q_x(D,Y)=(1-\alpha)Q_x(D,Y)+\alpha(B_Y + \min Q_Y(D,Z))$

In this formula, α represents the learning rate at which newer information overwrites the older one. In our simulation, a 50-50 weight is assigned to old and new information so that $\alpha=0.5$.

AMS: Average Message Size
LCV: Local Congestion Value
WT: Waiting Time
IF (WT $\leq 3\times$ AMS) Then LCV $\leq$ "00"; END IF;
ELSEIF (WT $\leq 9\times$ AMS) Then LCV $\leq$ "01";
END IF;
ELSEIF (WT $\leq 27\times$ AMS) Then LCV $\leq$ "10";
END IF;
ELSE LCV $\leq$ "11"; END IF;
END IF;

Fig.6. Encoding the waiting time into 2-bit

C. Application Traffic Profile:

Application traces are obtained from the GEMS simulator using some application benchmark suites selected from SPLASH-2. We use a 64-switch network configuration. The average performance gain of HARAQ across all benchmarks is up to 22% vs. C Routing and 33% vs. DBAR.

D. Hardware Analysis

IV. RESULTS AND DISCUSSION

A. Performance Evaluation under Uniform Traffic Profile:

In Fig.7.Using minimal and non-minimal routes along with the intelligent selection policy reduces the average network latency of HARAQ in the 8x8 network about 18% and 37%, compared with C- Routing and DBAR.

B. Performance Evaluation under Hotspot TrafficProfile

In addition, the impact of using non-minimal paths on the link utilization for the hotspot traffic profile is summarized in Table 3.

Table 2. The impact of using non-minimal paths onthe link utilization

		8x8	14x14
Maximum link utilization (flits/cycles)	DBAR	0.365	0.263
	C-Routing	0.352	0.256
	HARAQ	0.274	0.202
	Reduction over DBAR	25%	23%
	Reduction over CR	22%	21%

Table 3 .Hardware implementation details

Network Platform	Area (mm ²)	Avg. Power(W) Dynamic & Static	Max. Power(W) Dynamic & Static
DBAR	6.791	2.41	3.33
C-Routing	6.954	2.52	3.46
HARAQ	6.822	2.81	3.06

V. CONCLUSION

In this paper, we proposed a highly adaptive routing algorithm based on minimal and non-minimal paths for on-chip networks. The presented algorithm provides a large number of paths for routing messages using only an extra virtual channel in the Y dimension. In the proposed method, the use of some 180-degree turns on a single channel is also permitted in the algorithm. To choose a less congested path, we have utilized an optimized and scalable learning model to estimate the latency from each output channel to the destination switch. In the learning model, switches maintain distributed tables to store the global congestion information from different regions of the network. Finally, a less congested output channel is chosen by the selection function

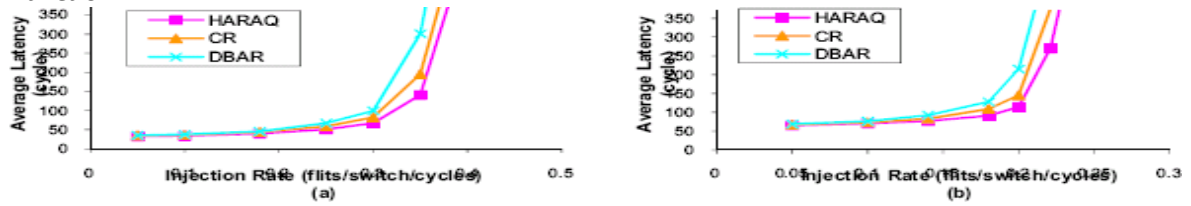


Fig.7. Performance under different loads in (a) 8x8 2D-mesh (b) 14x14 2D-mesh under the uniform traffic model

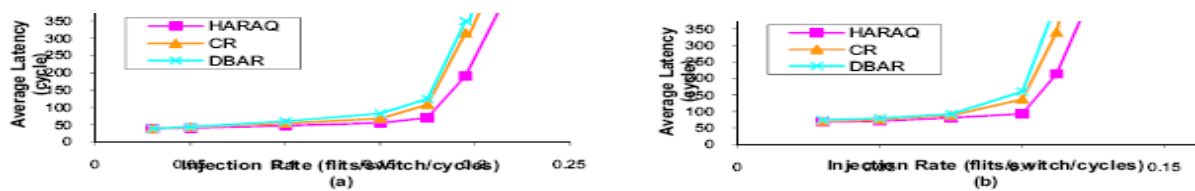


Fig.8. Performance under different loads in (a) 8x8 2D- mesh (b) 14x14 2D- mesh under the uniform traffic model

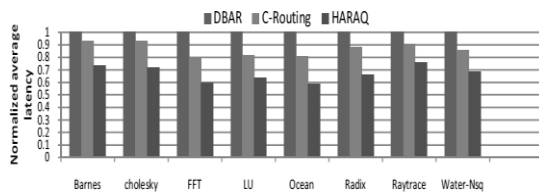


Fig: 9 Performance under different application benchmarks normalized to DBAR

REFERENCES:

- [1] HARAQ: Congestion-Aware Learning Model for Highly Adaptive Routing Algorithm in On-chip Network

- [2] M. Daneshtalab, et al., "Memory-Efficient On- Chip Network with Adaptive Interfaces", IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems, Vol. 31, No. 1, pp. 146-159, Jan 2012.
- [3] M. Dehyadegari, et al., "An Adaptive Fuzzy Logic-based Routing Algorithm for Networks-on- Chip," in Proceedings of 13th IEEE/NASAEISA International Conference on Adaptive Hardware and Systems (AHS), pp. 208-214, 2011.
- [4] G. Ascia, et al., "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.
- [5] P. Gratz, et al., "Regional Congestion Awareness for Load Balance in Networks-on-Chip," in Proc. HPCA, pp. 203-214, 2008.
- [6] S. Ma, et al., "DBAR: an efficient routing algorithm to support multiple concurrent applications in networks-on-chip", in Proc. of ISCA, 2011, pp.413-424
- [7] F. Chaix, et al., "A fault-tolerant deadlock-free adaptive routing for On Chip interconnects," in Proc. of DATE, pp. 1-4, 2011.
- [8] Z. Zhang, et al., "A reconfigurable routing algorithm for a fault-tolerant 2d-mesh network-on- chip", in Proc. of DAC, 2008.
- [9] S. Jovanovic, C. Tanougast, et al., "A new deadlock-free fault-tolerant routing algorithm for NoC interconnections", in Proc. of FPLA, pp.326- 331, 2009.
- [10] J. Wu and D. Wang, "Fault-tolerant and deadlock-free routing in 2-D meshes using rectilinear-monotone polygonal fault blocks", in Proc. Of Parallel Algorithms Appl., pp.99-111, 2005.
- [11] D. Fick, A. DeOrio, et al., "A Highly Resilient Routing Algorithm for FaultTolerant NoCs", in Proc. of DATE, pp. 21-26, 2009.
- [12] C. Feng, et al. "A reconfigurable fault-tolerant deflection routing algorithm based on reinforcement learning for network-on-chip", in Proc. of NoCArc, pp.11-16 , 2010.