Investigating the Viability of Maximum Flexibility Selection Function in Bufferless 2D Meshes

Mohamed A. Abd ElMohsen
Computer Engineering Department, Cairo University
Giza, Egypt
mohamedassem@eng.cu.edu.eg

Hatem M. El-Boghdadi
Computer Engineering Department, Cairo University
Giza, Egypt
helboghdadi@eng.cu.edu.eg

ABSTRACT

Bufferless NoCs have emerged as a solution to reduce power and area by eliminating buffers used for routing. Such networks handle contention using packet dropping or deflection. In this paper, we study the effect of MaxFlex selection function on 2D bufferless meshes for both a fixed and a variable step size.

For fixed step size, we perform an analytical study for the effect of using MaxFlex with different step size on the performance of 2D bufferless meshes. The analysis indicates that, as the step size increases the traffic in the central part of the network bisection relaxes. Simulation results show that, both average packet latency and average deflection count decrease as the step size used increases. Additionally, over different sizes of meshes, the results show that the network performs best if the step size is equal 60-80% of the mesh dimension. Then, we consider using variable step size in which a packet is routed using a step size dependent on the Manhattan distance, *d*, between the source and destination. Simulation results show that, using MaxFlex, a step size of 60% of the distance *d* enhances the packet latency over using fixed step size, straight line selection function and random productive port selection function by around 29%, 97% and 99% respectively.

Keywords

Bufferless NoC, Selection Function, Maximum Flexibility

1. INTRODUCTION

Network-on-Chip (NoC) is seen as a solution for the limitations in the traditional communications approaches (e.g. buses) especially after the tremendous increase in the number of the communicating modules within a single silicon chip [1, 2]. NoC is a group of switches connecting homogeneous or heterogeneous nodes in a multiple point-to-point fashion. Although buffered NoCs became the dominating approach for communication between cores within chip (more scalable, reliable, and predictable), they consume significant power and chip area. For instance, in [3, 4], the buffers within a single switch consume around 37% power and 80% area. In addition to consuming heavy power and area, buffered NoCs are more complex to design as they require extra handlers for packets placement and buffer overflow.

Bufferless NoCs have emerged as a solution to decrease power and area requirements. Bufferless NoCs eliminate the buffers used within switches which has a direct impact on power and area. In contrast to the traditional buffered NoC; when two packets compete for the same output port, the allocator either drops or deflects (misroute) the losing packet. Dropped packet should be retransmitted again. On the other hand, deflected packet follows non-productive port. In this paper, we adapt the deflection approach.

To perform routing, a routing function and a selection function are used in order. The routing function provides a set of productive ports that can be used as an output to get closer to the destination. Then, the selection function selects one of these ports. In [5-7], several selection functions were evaluated on 2D meshes showing their impact on NoC performance. While in [8, 9], various selection functions were evaluated under Fat-Tree topology showing that the selection function has a great impact on Fat-Tree NoCs performance.

One important selection function is Maximum Flexibility (MaxFlex) [11]. This selection function forwards the packet along the dimension with the longest distance to the destination to maximize the number of output ports provided by the routing function as the packet gets closer to its destination (not stuck in one dimension leading to one productive output port only). A packet initially follows the dimension with higher hop count. When it reaches a switch where the difference in the *X* dimension is equal to the difference in the *Y* dimension, it follows the diagonal. The path of the diagonal is dependent on the step size used (Step size of *SS* means that a packet moves *SS* steps in *X* dimension and then *SS* steps in *Y* dimension.)

As MaxFlex tries to move on the diagonals, it causes the traffic to be concentrated in the central part of the network bisection leading to uneven switches utilization which degrades the performance [12]. In this paper we study the effect of the applying the MaxFlex to bufferless NoCs with different step sizes.

For 2D meshes, Badr and Podar [11] proposed MaxFlex as an optimal selection function for meshes. However, in [5, 12, 13], the authors analyzed different selection functions for meshes and showed that MaxFlex is not the best for this topology.

In this paper, we study the performance of $n \times n$ 2D bufferless meshes if MaxFlex selection function is used (we refer to the value n by the *mesh dimension*). Our work has two main directions; one considers using fixed step size while the other handles variable step size. For fixed step size MaxFlex; we analyze the traffic in 2D meshes. We first identify 12 types (we use the terms *type* and *pattern* interchangeably) of traffic and prove that they collectively constitute the MaxFlex traffic in the 2D mesh. The analysis shows that increasing the step size leads to a better load distribution over the NoC switches. In other words, the central part of the network bisection becomes more relaxed.

Then, we study the effect on the NoC performance in terms of packet latency (Last Flit Ejection Time – First Flit Generation Time) and deflection count (Number of times a flit was forced to go through a non-productive port). We simulate a 10x10 mesh under uniform traffic while varying the step size from 1 to 9. The results show that increasing the step size leads to better packet latency and smaller deflection count thus enhancing the NoC performance. Also, for different mesh sizes, simulation results

MES '15, June 13 - 14, 2015, Portland, OR, USA
© 2015 ACM. ISBN 978-1-4503-3408-2/15/06…$15.00
DOI: http://dx.doi.org/10.1145/2768177.2768185

show that a step size of 60-80% of the mesh dimension leads to better performance.

As for the variable step size, we study the effect of having different step size for each packet on the traffic distribution. The value of step size is selected to be a function in the Manhattan distance, *d*, between the source and destination (in particular, a percentage of *d*). Simulation results show that increasing the percentage leads to better NoC performance (packet latency and deflection count) which comes in line with the results of increasing the fixed step size in the first part.

The next section presents the analysis of fixed step size under MaxFlex. In Section 3, we present the simulation results. Section 4 describes the use of variable step and presents the simulation results. Section 5 summarizes our results and makes some concluding remarks.

2. FIXED STEP SIZE MAXFLEX

In this section, we analyze the traffic in 2D meshes under MaxFlex. First, we prove that any packet passing through a node can be classified into one of twelve traffic patterns. Then in section 2.2, we present the results of increasing the step size on the traffic distribution among the different switches.

2.1 Analysis of Fixed Step Size MaxFlex

In this section, we study the effect of step size on the distribution of packets through 2D mesh network. In doing that, for a certain step size, we count the number of packets passing through each switch (all ports included). To ease this counting, we divide the traffic going through a switch into 12 different patterns.

In the following analysis, we assume that:

- Each node sends only one packet to each other node (i.e. each node sends n^2-1 packets)
- Packet length is one Flit
- No deflections (i.e. path of each packet is only affected by the value of step size and not by misrouting due deflection). This assumption is set to ease the analysis.

For an $n \times n$ mesh, let *W* be a switch (*i,j*), where $1 \leq i, j \leq n$. Let *P* be a packet going from source node *S* (X_{src}, Y_{src}) to destination *D* (X_{dst}, Y_{dst}). Let $\Delta Y = \left| Y_{src} - Y_{dst} \right|$ and $\Delta X = \left| X_{src} - X_{dst} \right|$. Any packet passing through switch *W* (due to any two nodes communication) falls under one of the following 12 patterns:

Type 1*:* Packets destined to node (*i,j*).

Type 2*:* Packets injected by node (*i,j*).

Type 3*:* Packets passing through *W* injected by node (*i,k*) and destined to node (*i,m*) where $1 \leq k, m \leq n$ and $j \neq k \neq m$ (i.e. same row communication).

Type 4*:* Packets passing through *W* injected by node (*k,j*) and destined to node (*m,j*) where $1 \leq k, m \leq n$ and $i \neq k \neq m$ (i.e. same column communication).

Type 5*:* Packets passing through *W* as a result of communication between nodes on the same diagonal as node (*i,j*).

Type 6*:* Packets passing through *W* as a result of communication destined to nodes on the same diagonal as node (*i,j*) from nodes with $\Delta X > \Delta Y$ (i.e. leads to moving on a row first).

Type 7*:* Packets passing through *W* as a result of communication destined to nodes on the same diagonal as node (*i,j*) from nodes with $\Delta X < \Delta Y$ (i.e. leads to moving on a column first).

Type 8*:* Packets passing through *W* as a result of communication between nodes on a diagonal other than node (*i,j*) diagonal.

Type 9*:* Packets passing through *W* as a result of communication destined to nodes on a diagonal other than node (*i,j*) diagonal from nodes with $\Delta X > \Delta Y$.

Type 10*:* Packets passing through *W* as a result of communication destined to nodes on a diagonal other than node (*i,j*) diagonal with from nodes with $\Delta X < \Delta Y$.

Type 11*:* Packets passing through *W* as a result of communication between node (*i,k*) from same row as node (*i,j*) and nodes on node (*i,m*) diagonal where $1 \leq k, m \leq n$ and $j \neq k \neq m$.

Type 12*:* Packets passing through *W* as a result of communication between node (*k,j*) from same column as node (*i,j*) and nodes on node (*m,j*) diagonal where $1 \leq k, m \leq n$ and $i \neq k \neq m$.

Lemma 1 In an $n \times n$ mesh, under MaxFlex, any packet going from a source node to a destination node falls under one of the mentioned twelve traffic patterns.

Proof Outline:

In the proof, we classify the patterns going through *W* (*i,j*) into two main categories: 1) the patterns due to moving to nodes on same row, column, or diagonal as (*i,j*) and 2) the patterns due to moving to nodes on different diagonal than that of (*i,j*) (i.e. the effect on (*i,j*) caused by category one). Here we show only the first category and the proof for the second category is omitted for space limitation.

Concerning the first category, consider the possible values for ΔX w.r.t ΔY. We have the following cases:

Case 1 $\Delta X = \Delta Y$ **:** *P* moves on the diagonal from *S* to *D*. This case causes the pattern defined in Type 5.

Case 2 $\Delta X = 0$ **:** *P* moves on a column from *S* to *D*. This case causes the pattern defined in Type 4.

Case 3 $\Delta Y = 0$ **:** *P* moves on a row from *S* to *D*. This case causes the pattern defined in Type 3.

Case 4 $\Delta X > \Delta Y$ **:** *P* moves on a row till $\Delta X = \Delta Y$ then follows Case 1. This case causes the pattern defined in Type 6.

Case 5 $\Delta X < \Delta Y$ **:** *P* moves on a column till $\Delta X = \Delta Y$ then follow Case 1. This case causes the pattern defined in Type 7.

■

2.2 Analysis Results

In order to study effect of increasing the step size for MaxFlex, we formally calculate the number of packets (based on each type) passing through each switch in a 10x10 2D mesh network. The equations representing the number of packets are omitted here for lack of space.

We choose some representative switches based on their locations in a 10x10 mesh network to represent border switches and core switches. We choose Switch (0, 0), Switch (0, 3) and Switch (0, 6) as border switches and Switch (3, 3), Switch (3, 6), Switch (5, 5) as core switches.

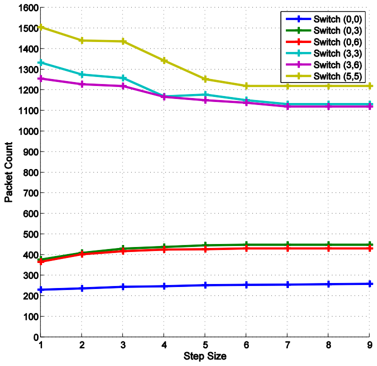

Figure 1 Number of packets passing through sample border and core switches over different fixed step size values

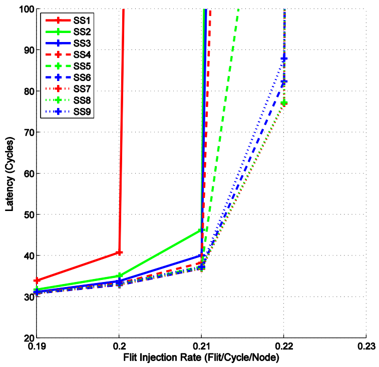

Figure 2 Average packet latency for different fixed step size values

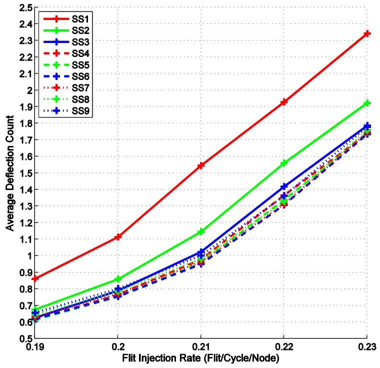

Figure 3 Average deflection count for different fixed step size values

Figure 1 shows the number of packets passing through each of the mentioned switches with different step sizes. From the figure, we notice different trends; for the border nodes, the number of packets passing through the switch increases as the step size increases, while for the core switches, the number of packets decreases as the step size increases. In other words, the concentration in the central part of network bisection is relaxed.

3. EXPERIMENTAL RESULTS

3.1 Experimental Methodology

We evaluate the network performance of bufferless NoCs using the General purpose Simulator gpNoCsim [14]. In this section, we evaluate MaxFlex selection function using different step sizes in terms of average packet latency and average deflection count. In addition, we calculate an approximate value for the optimal step size given a certain dimension.

We use the 2D mesh topology of varying size to model the network. Each switch has 5 input ports and 5 output ports, including the injection ports. Each of the switch latency and link latency is 1 cycle. In our configuration, we assume that each link is 128-bit wide and each data packet consists of 8 flits, each flit is assumed to have 128 bits. All packets are of fixed length. For comparing the effect of increasing the step size, we use a 10x10 mesh. On the other hand, for calculating the optimal step size given the 2D mesh dimension, we use a mesh size varying from 5x5 to 12x12. The destination address of a packet is determined by the statistical process of the uniform traffic pattern. Within each simulation there is a warm-up period of 100,000 cycles. After that 10,000 packets are injected per node and the simulation terminates when these packets are all received. We adapt the FLIT-BLESS [10] routing algorithm.

3.2 Evaluation Results

Here we show the results of increasing the step size under MaxFlex. Figures 2 and 3 show that as the step size increases, the average packet latency and the average deflection count decrease. These results matches the analysis results in section 2.2, as the better traffic distribution shown in the analysis can lead to better link utilization. In turn, this leads to faster delivery for the packets and hence better packet latency and less misrouting due to contention.

3.3 Estimation of the Value of the Step Size

In this section, given an $n \times n$ mesh, we estimate the best value of the step size. In order to do this, we simulate the MaxFlex under different 2D mesh sizes varying from 5x5 to 12x12 and within each network we use step sizes ranging from 1 to $n-1$. For example, for 7x7 network, we use step sizes ranging from 1 to 6. The results are shown in the Table 1.

Table 1. Step size value percentage

Mesh Size $n \times n$	Best Step Size *SS*	Percentage *SS*/*n*
5x5	4	80
6x6	4	66.67
7x7	5	71.43
8x8	6	75
9x9	6	66.67
10x10	8	80
11x11	8	72.73
12x12	9	75

Table 1 shows that using a step size with a value ranging from 60% to 80% of the 2D mesh dimension leads to better network performance.

4. VARIABLE STEP SIZE MAXFLEX

In this section, we examine the use of variable step size. We assign a different step size for each packet based on the Manhattan distance between the packet's source and destination. For packet *P*, *l*et the distance between the source and destination is distance *d*, the value of the step size for *P* is a percentage of *d*.

4.1 Evaluation Results

Here we show the results of varying the step size under MaxFlex. We examine different percentage value ranging from 10% to 90%. Figures 4 and 5 show that as the percentage value increases, the average packet latency and the average deflection count decrease. The best percentage value is about 60% of the distance. Also, Figure 4 and 5 show that using higher percentage values degrades the performance as it leads to step sizes that can be similar to using a large fixed step size. These results match the results for the fixed step size, as using the percentage value of 60% leads to larger step size value for the packets with long distance to go and smaller step size for the packets with short distance to go.

Now, we compare variable step size MaxFlex with fixed step size MaxFlex, straight line selection function and random productive

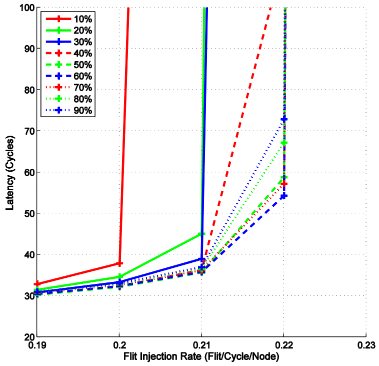

Figure 4 Average packet latency for variable step size MaxFlex using different % values

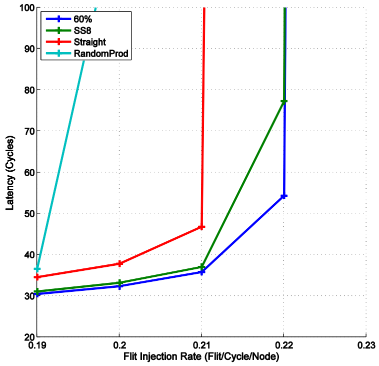

Figure 6 Average packet latency for different selection functions

port selection function. In the straight line selection function, the flit favors the X-axis movement till there are no steps remaining in X dimension then moves to the Y dimension. In the random productive port selection function, the flit randomly chooses from the list of productive ports available at each step. Figures 6 and 7 show that using variable step size MaxFlex leads to better average packet latency and less deflection count compared with fixed step size MaxFlex, straight line selection function and random productive port selection function. Specifically, using a variable step size of 60% of the distance enhances the average packet latency by around 29%, 97% and 99% over using fixed step size of 8 under MaxFlex, straight line selection function and random productive port selection function respectively.

5. CONCLUSION

In this paper, we evaluated the bufferless 2D mesh performance using different fixed values for the step size and using variable step size under MaxFlex selection function. We showed that using a variable step size under MaxFlex led to around 29% enhancement in latency over using a fixed step size. One possible extension to this work is to find other ways to calculate the variable step size in a way to enhance the network switches usage. Other directions include the evaluation of MaxFlex in 2D mesh under different traffic patterns.

REFERENCES

[1] L. Benini, G.De. Micheli, Networks on chips: a new SoC paradigm, IEEE Transactions on Computer 35 (1) (2002) 70–78.

[2] W.J. Dally, B. Towles, Route packets, not wires: on-chip interconnection networks, Design Automation Conference (2001) 684–689.

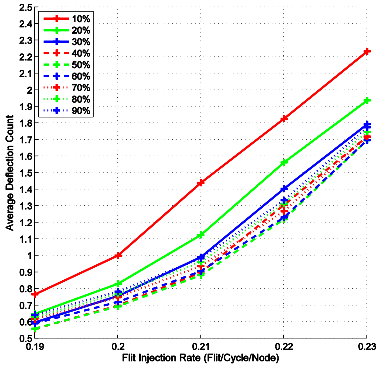

Figure 5 Average deflection count for variable step size MaxFlex using different % values

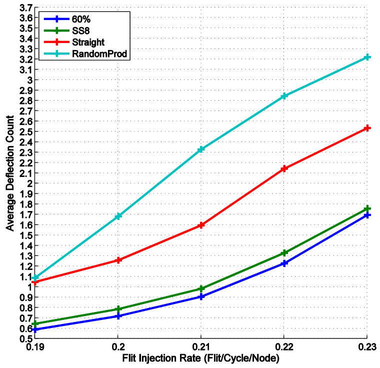

Figure 7 Average deflection count for different selection functions

[3] Y. Hoskote, S. Vangal, A. Singh, N. Borkar, S. Borkar, A 5-ghz mesh interconnect for a teraflops processor, Micro (2007) 51–61.

[4] P. Gratz, C. Kim, R. McDonald, S.W. Keckler, D. Burger, Implementation and evaluation of on-chip network architectures, International Conference on Computer Design ICCD (2006) 477–484.

[5] W. Feng and K. Shin, Impact of Selection Functions on Routing Algorithm Performance in Multicomputer Networks, ACM International Conference on Supercomputing (1997).

[6] A. Funahashi, M. Koibuchi, A. Jouraku, H. Amano, The Impact of Output Selection Function on Adaptive Routing, International Conference on Computers And Their Applications (2001).

[7] J.C. Mart´ınez, F. Silla, P. L´opez and J. Duato, On the Influence of the Selection Function on the Performance of Networks of Workstation, High Performance Computing, 2000.

[8] Abeer Farouk, Hatem M. El-Boghdadi, On the Influence of Selection Function on the Performance of Fat-Trees under Hot-Spot Traffic, International Conference on Computer Systems and Applications AICCSA (2011).

[9] F. Gilabert, M.E. Gomez, P. Lopez, J. Duato, On the Influence of the Selection Function on the Performance of Fat-trees, European Conf. on Parallel Computing (2006).

[10] T. Moscibroda, O. Mutlu, A case for bufferless routing in on-chip networks, ACM SIGARCH Computer Architecture News (2009) 196–207.

[11] S. Badr and P. Podar, An Optimal Shortest-Path Routing Policy for Network Computers with Regular Mesh-Connected Topologies, IEEE Transactions on Computers, (1989).

[12] W. J. Dally and H. Aoki, Deadlock-Free Adaptive Routing in Multicomputer Networks Using Virtual Channels, IEEE Transactions on Parallel and Distributed Systems (1993).

[13] Timothy Weller and Bruce Hajek, Comments on "An Optimal Shortest-Path Routing Policy for Network Computers with Regular Mesh-Connected Topologies", IEEE Trans. Computers (1994) 862-863.

[14] http://www.buet.ac.bd/cse/research/group/noc/index.htm