The Impact of Output Selection Function on Adaptive Routing

A. Funahashi
Dept. of Information Technology
Mie University
Tsu, Mie 514-8507

M. Koibuchi, A. Jouraku, H. Amano
Dept. of Computer Science
Keio University
Yokohama, Kanagawa 223-8522

Abstract

Recently, adaptive routing algorithms which achieve a high performance compared with deterministic ones are commonly used in large scale parallel machines. Although their performance is influenced with an output selection function or output channel selection algorithm, only a few studies depending on specific topologies or specific traffic patterns have been done. In this paper, we propose two output selection functions called LDSF (load-dependent selection function) and LRU(Least Recently Used) selection function, which are not based on a specific topology nor traffic pattern. Result of simulations shows that proposed output selection functions improved the performance compared with traditional ones up to 30%.

1 Introduction

Communication network is one of the critical components of a highly parallel multicomputer. Recently, multicomputers providing more than a thousand computation nodes are commercially available, and efforts have been exerted to implement Massively Parallel Computers (MPCs) with tens of thousands nodes. In these systems, the interconnection network often dominates the total system performance.

In order to make the best use of potential performance of interconnection networks, adaptive routing algorithms which can select the route of packet dynamically have been introduced. In such algorithms, when a packet encounters a faulty or congested node, another bypassing route can be selected. However, we must not forget that an adaptive routing has a possibility of deadlock. There are a lot of researches on deadlock-free adaptive routing techniques[1][2][3][4]. These techniques are classified into two methods: using only minimal paths, and using alternative paths with additional routing steps. The former method does not require extra routings while the latter can use alternative routes more flexibly.

Since multiple paths can be selected in both methods, an algorithm for selecting an available output channel should be established. Such algorithm is called "output selection function". Although the output channel selection function often dominates the system performance, only simple algorithms which depend on specific topologies or traffic patterns[5][6][7] have been proposed. Here, novel output channel selection functions (LDSF and LRU selection function) which are independent of traffic patterns nor topologies are proposed and evaluated.

The following sections are organized as follows. Section 2 shows traditional output selection functions. In Section 3, we will propose a new output selection function called the LDSF (load-dependent selection function). In Section 4, another simpler algorithm called the LRU(Least Recently Used) selection function is proposed. Section 5 shows the performance evaluation results.

2 Output selection functions

An adaptive routing is a technique to select a route of packet dynamically. By using it, there exist plural routes between source and destination node, and each packet can select them by choosing output channel dynamically. Selection of the output channel is done depending on the condition of channels. For example, if a channel is being used (that is, in busy condition), the other channel has priority over the busy channel. But if both output channels are not being used (that is, in free condition), output selection function will decide which output channel should be used. These selection functions are sometimes called routing policies[8][9].

An ideal output selection function is that restrains a generation of complex cyclic dependency, and distributes a traffic fairly. Using this ideal output selection function, all channels in the network would be effectively used and packets would be smoothly transferred, thus much improves the performance.

Although a lot of researches have been done on adaptive routing algorithms[1][2][3][4][10], there are few studies on an impact of output selection function to the throughput and latency of networks.

The simplest output selection function is called "random selection function". The random selection function chooses one output channel among available output channels at random. By using it, traffic will be distributed to all directions randomly.

"Dimension order selection function" has been proposed for k-ary n-cubes and meshes. It chooses an output channel which belongs to the lowest dimension among available output channels. For example, if there exist free output channels on x, y dimension, this selection function chooses the x direction output channel.

On the other hand, "zigzag selection function" [8] chooses an output channel whose direction has the maximum hop to the destination. That is, by using this selection function on the mesh topology, packet will be transferred to a diagonal direction toward the center of the network. For example, on a two-dimensional mesh, when a packet is send from source node $s\left(x_s, y_s\right)$ to destination node $d\left(x_d, y_d\right)$ and both the x direction channel and y direction channel are free, compare the value $|x_d - x_s|$ and $|y_d - y_s|$ and the packet will be send to the larger direction.

Also theoretically the most suitable output selection function on mesh networks has been proposed by Badr and Podar[8]. However, this algorithm only stands up on an ideal assumption that all channels are utilized equally.

3 Load-dependent selection function (LDSF)

The traditional output selection functions described in Section 2 have a possibility to send a packet to a congested direction even if there exist plural free output channels. This comes from that traditional output selection functions take no thought of the network congestion situation. In practice, the hot spot congestion is sometimes generated even when many free channels exist in the network simultaneously. To address this problem, we propose a novel output selection function "load-dependent selection function(LDSF)" which avoids the generation of hot spots and bypasses it to equalize the channel utilization ratio. The LDSF can choose the output channel depending on the status of network.

Although there are a lot of methods to recognize the network congestion situation, it is hard to collect congestion information at each node and send to one centralized node on every clock cycle in practice. So, the LDSF will grasp the congestion information locally only by the utilization ratio of a link during a certain period.

For this purpose, a counter is provided on each physical link in a router, and incremented on every flit transfer. That is, when a packet arrives at the node whose counter is 0, the counter will be set to the packet length after the packet is transferred to the next node (i.e. the tail flit is transferred to the next node). Also, the counter is decremented at every clock cycle if there is no packet transfer on its physical link. If the counter becomes 0, then it is never decremented any more. On the routing, if there exist plural available outputs, counters corresponding to available output links are compared, and the output link with the smallest counter is selected. Examples of the LDSF operation on two dimensional torus are shown in Figure 1 and Figure 2.

In Figure 1, two output directions are available, and each counter is "120" and "64" respectively. With this condition, the output direction with counter number "64" is chosen for routing. When the output direction is busy(Figure 2), it is not selected even the counter number is small.

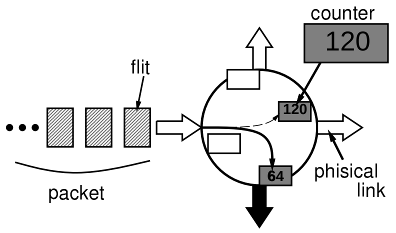

Figure 1: LDSF on 2D torus(when 2 output channels are available)

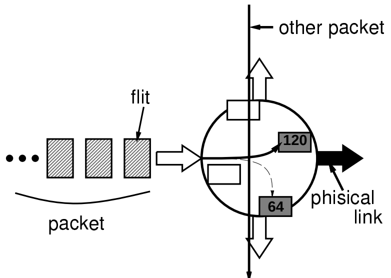

Figure 2: LDSF on 2D torus(when one output channel is busy)

Also, the counter is not changed if a packet occupies a series of channels simultaneously and is blocked under wormhole routing. Thus, the router which holds this packet can keep the congestion information.

Given that packet length is usually larger than network diameter, this method is an effective approach to grasp network status.

Structure of a router in modern parallel machine should be simple for high frequency operation [11], [12], [13]. However, since the LDSF simply requires a counter for each physical link, it does not degrade operation frequency of the router nor much increases the amount of hardware. Also, the LDSF can be applied to any network topology.

Here we will describe some features of the LDSF.

1. Don't use the channel which transferred the packet just before (if possible).
2. Routing information (congestion information) will be vanished in a certain period of time.
3. Counter is necessary for each physical link.

1. will restrain a generation of hot spot and 2. will exclude unnecessary information of packet which is far from the router or already reached to its destination, and preserve the congestion information of its neighboring nodes.

4 LRU selection function

The LDSF can choose an output direction depending on the congestion of the network unlike traditional output selection functions. However, when packet length becomes large, a large counter is required for each link, and the hardware cost may not be negligible.

Here, we propose another output selection function called "LRU (Least Recently Used) selection function". The LRU selection function is a simplified version of the LDSF, and thus requires much smaller hardware than that required in the LDSF.

In the LRU selection function, an output channel which is least recently used is selected. In this policy, the traffic is distributed to channels whose utilization ratio is small, and the similar effect as the LDSF can be achieved.

Unlike the LDSF which selects the output channel by the number of transferred flits on each physical link in a certain period of time, the LRU selection function chooses the output channel only by the passed time from the last transferred packet.

The idea of the LRU selection function is the same as the LRU replacing policy used in virtual memory. While the LRU replacing policy is used to maximize the utilization of memory access locality, the LRU selection function is used to exclude a traffic locality and distribute the traffic uniformly by choosing the least recently used channel.

5 Performance evaluation

In this section, the performance of the above two output selection functions (LDSF, LRU selection function) is evaluated by computer simulation and compared with others (dimension order, random, zigzag and simple deterministic e-cube routing[14]). Adaptive routing used in this simulation is a fully adaptive routing called $*$-channel proposed by Duato[1]. A flit-level simulator written in C++ is developed. Network size, the number of virtual channels, and packet length are selected just by changing parameters. Each node consists of a processor, request queue and the router which provides bidirectional channels for connecting with neighboring nodes. A common router which consists of channel buffers, crossbar, link controller, virtual channel controller and their control circuits is assumed.

5.1 Simulation parameters

The destination of a packet is determined by the traffic pattern in this simulator. Two traffic patterns are used here:

- *uniform*
  All destination nodes are selected randomly, and so the traffic is distributed uniformly.
- *bit-reversal:*
  A node with the identifier $(a_0, a_1, \cdots, a_{n-1})$ sends a packet to the node whose identifier is the bit reversal $(a_{n-1}, \cdots, a_1, a_0)$ of the source node.

The following two measures are used for evaluations.

Network latency: Let the time when a node p inserts the first flit of a packet into the input buffer be t_0, and the time when the tail flit of the packet arrives at the processor of destination node q be t_1. Here, we call $T_{lat}(p, q) = t_1 - t_0$ the *network latency*, which is commonly used as a measure.

Throughput: Throughput is the maximum amount of information delivered per time unit. Here, throughput could be measured in flits per node in each clock cycle.

Simulation parameters are set as Table 1.

Table 1: Simulation parameters

Simulation time	50,000 clocks (ignore the first 5,000 clocks)
Network	2D torus or 3D torus
Network size	16 × 16 (256 nodes) or 32 × 32 (1024 nodes) or 8 × 8 × 8 (512 nodes)
The number of virtual channels	2 (deterministic routing) 3 (adaptive routing)
Packet length	128 flits (fixed)
Routing method	wormhole
Flit transfer time	3 clocks

As shown in the table, three clocks are required for a flit passes through a router, that is, one clock for routing, one for transferring the flit from input channel to output channel through a crossbar, and one for transferring the flit to the next node respectively.

We ignored the first 5,000 clocks for the evaluation, since the network is not stable in that period. When the network is saturated, the execution of simulation is aborted.

5.1.1 Routing algorithm

Here, we will describe the details of routing algorithms especially on the usage of virtual channels.

e-cube routing

In e-cube routing on two dimensional torus, a packet is transferred to the x dimensional direction first and then the packet will be transferred to the y direction. In case of torus, two paths with a wrap-around channel and without it are necessary to avoid deadlock on each dimention. Thus on the e-cube routing, more than two virtual channels are required.

***-channel**

Duato states a general theorem defining a criterion for deadlock freedom and then uses the theorem to propose a fully adaptive, profitable, progressive protocol[1], called *-channel. The theorem states that by separating virtual channels on a link into restricted and unrestricted partitions, a fully adaptive routing can be performed and yet be deadlock-free. This is not restricted to a particular topology or routing algorithm. Cyclic dependencies between channels are allowed, provided that there exists a connected channel subset free of cyclic dependencies.

Simple description of *-channel is as follows.

a. Provide that every packet can always find a path toward its destination whose channels are not involved in cyclic dependencies(escape path).

b. Guarantee that every packet can be send to any destination node using an escape path and the other path on which cyclic dependency is broken by the escape path(fully adaptive path).

By selecting these two routes (escape path and fully adaptive path) adaptively, deadlock can be prevented. On torus, more than three virtual channels are required.

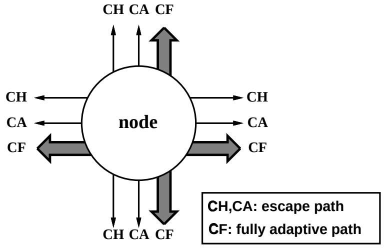

Figure 3: Virtual channels required by *-channel on 2D torus.

In the simulation, three virtual channels are provided on each physical link as shown in Figure 3. In Figure 3, two virtual channels (CA and CH) are used for e-cube routing and these channels provide escape path. A virtual channel CF is used for fully adaptive routing. Figure 3 shows an example on two dimensional torus, but this can be applied on any type of k-ary n-cube.

5.2 Uniform traffic

Figure 4, 5 and 6 show simulation results under uniform traffic on 2D torus(16×16, 32×32) and 3D torus ($8 \times 8 \times 8$) respectively.

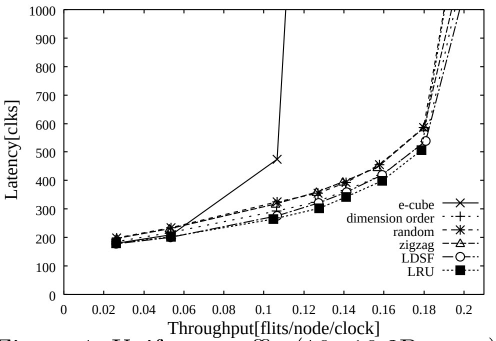

Figure 4: Uniform traffic (16×16 2D torus)

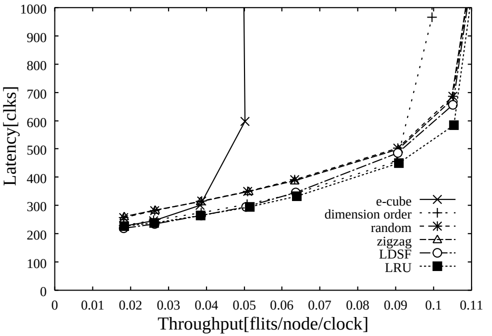

Figure 5: Uniform traffic (32×32 2D torus)

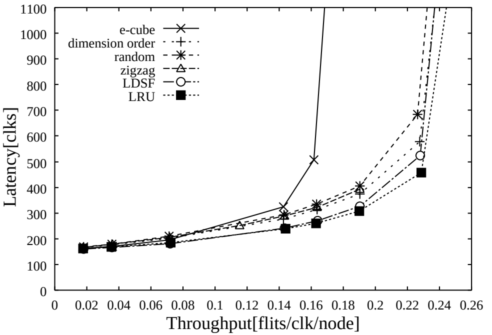

Figure 6: Uniform traffic (8×8×8 3D torus)

Each figure contains a simulation result of deterministic routing only for comparison with adaptive routing. It is quite obvious that adaptive routing greatly improves the performance especially when throughput grows higher. *-channel is a minimal adaptive routing so there is no difference in the number of hops between these two routing algorithms, but *-channel can use virtual channels effectively thus improves the performance.

From Figure 4, 5 and 6, it appears that the LRU selection function and the LDSF achieve only small improvement compared with other output selection functions. This is because the traffic is well distributed under uniform traffic and thus output selection function won't much effect the performance. However, traffic congestion sometimes occurs and this causes limited performance improvement. Thus, the improvement tends to grow with larger size and dimension.

Both the LDSF and the LRU selection functions cannot perfectly grasp a congestion information on the network, and it is shown that under well balanced traffic, simple output selection function like the LRU selection function achieves higher performance than the LDSF.

5.3 Bit reversal traffic

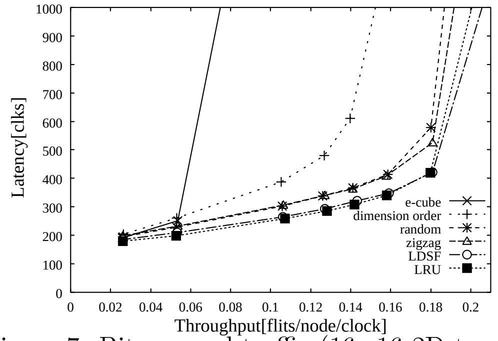

Figure 7: Bit reversal traffic (16×16 2D torus)

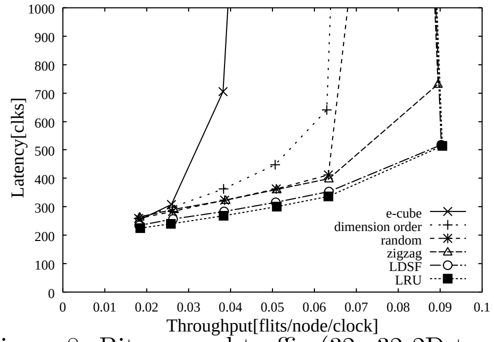

Figure 8: Bit reversal traffic (32×32 2D torus)

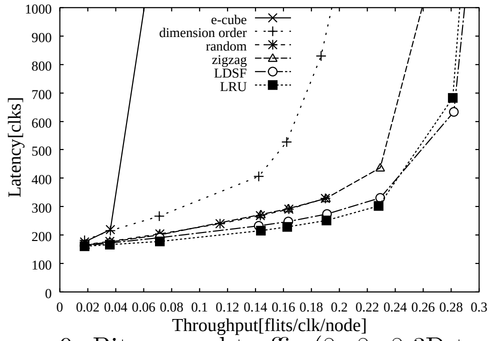

Figure 9: Bit reversal traffic (8×8×8 3D torus)

Figure 7, 8 and 9 show simulation results with the bit reversal traffic. The latency of dimension order selection function drastically increases under heavy traffic while others are moderately increased. This shows that the difference of output selection function greatly affects the performance. For example, a throughput under dimension order selection function shows twice as that of the deterministic routing while the LDSF shows three times as shown in Figure 9.

The LDSF guesses the network status by counting the number of transferred flits. On the other hand, the LRU selection function guesses by counting a passing time since the last flit has transfered. From these simulation results, though the LRU selection function is shown to be effective enough, the LDSF will become advantageous on large dimension networks and under non-uniform traffic.

5.4 Channel utilization characteristics

Here, channel utilization characteristics are evaluated.

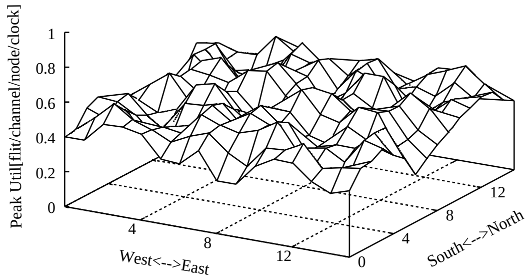

Figure 10: Peak channel util. for dimension order

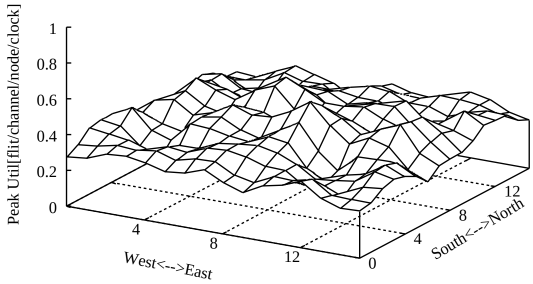

Figure 11: Peak channel util. for the LDSF

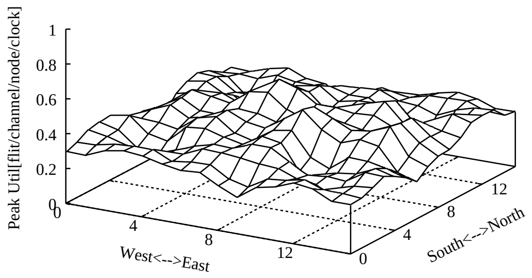

Figure 12: Peak channel util. for the LRU

Figure 10, 11 and 12 show the channel utilization characteristics for output selection functions. Three output selection functions (dimension order, LDSF, LRU) are choosen for the evaluation. These figures depict the case of 16 × 16 2D torus. The throughput is 0.18 and the bit reversal traffic is used (The performance evaluation under this assumption is shown in Figure 7).

The z-axis represents the peak channel utilization, while the x-axis and y-axis show the relative position of each channel in the network. The "peak channel utilization" is the maximum channel utilization value on each channel, which is calculated on every 1,000 clocks. If the output selection function can distribute the network traffic equally, the peak channel utilization on each node will be reduced.

As shown in Figure 10, 11 and 12, it is shown that both the LDSF and LRU obviously reduce and equalize the peak channel utilization compared with dimension order.

6 Conclusion

Two output selection functions, the LDSF and LRU selection function which choose the output direction depending on the status of network are proposed.

Both the proposed output selection functions can be used on any type of network topology and traffic pattern. From the simulation result, it is shown that they improved the performance especially in the case of large network size and dimension up to 30%. Also, they are especially effective with non-uniform traffic and larger network size.

Though the LRU selection function requires small hardware, it shows notable performance improvement. However, the LDSF is advantageous when a large dimension (more than three) network is used or traffic becomes heavy.

We plan to extend our instruction level simulator [15] to examine proposed output selection functions using realistic traffic patterns.

References

[1] J. Duato. A Necessary And Sufficient Condition For Deadlock-Free Adaptive Routing In Wormhole Networks. *IEEE Transaction on Parallel and Distributed Systems*, 6(10), 1995.

[2] C. J. Glass and L. M. Ni. Maximally Fully Adaptive Routing in 2D Meshes. *Proc. of ISCA92*, pages 278–287, 1992.

[3] A. A. Chien and J. J. Kim. Planar-Adaptive Routing: Low-cost Adaptive Networks for Multiprocessors. *Proc. of ISCA92*, pages 268–277, 1992.

[4] W. J. Dally and H. Aoki. Deadlock-Free Adaptive Routing in Multicomputer Networks Using Virtual Channels. *IEEE Transaction on Parallel and Distributed Systems*, 4(4):466–475, 1993.

[5] W. C. Feng and K. G. Shin. Impact of Selection Functions on Routing Algorithm Performance in Multicomputer Networks. *Proc. of the 11th Anuual Conference on Supercomputing*, July 1997.

[6] L. Schwiebert and R. Bell. The Impact of Output Selection Function Choice on the Performance of Adaptive Wormhole Routing. *Proc. of International Conference on Parallel and Distributed Computing Systems*, pages 539–544, October 1997.

[7] L. Schwiebert. A Performance Evaluation of Fully Adaptive Wormhole Routing including Selection Function Choice. *IEEE International Performance, Computing, and Communications Conference*, pages 117–123, February 2000.

[8] S. Badr and P. Podar. An Optimal Shortest-Path Routing Policy for Network Computers with Regular Mesh-Connected Topologies. *IEEE Transactions on Computers*, 38(10):1362–1371, October 1989.

[9] J. Wu. An Optimal Routing Policy for Mesh-Connected Topologies. *Proc. of International Conference on Parallel Processing*, 1:267–270, 1996.

[10] L. M. Ni and P. K. McKinley. A Survey of Wormhole Routing Techniques in Direct Networks. *IEEE Transactions on Computers*, February 1993.

[11] Baverton, OR, and Supercomputer Systems Division Intel Corporation. *Paragon XP/S Product Overview.* 1991.

[12] W. Oed. The Cray Research Massively Parallel Processing System: Cray T3D. *Cray Research*, 1993.

[13] C. B. Stunkel et al. Architecture and implementation of Vulcan. *Proc. of the 8th International Parallel Processing Symposium*, pages 266–274, April 1994.

[14] W. J. Dally and C. L. Seitz. Deadlock-Free Message Routing in Multiprocessor Interconnection Networks. *IEEE Transactions on Computers*, 36(5):547–553, May 1987.

[15] M. Wakabayashi, K. Inoue, and H. Amano. ISIS: Multiprocessor Simulator Library. In *Proc. of Sixteenth IASTED International Conference Applied Informatics – AI'99*, pages 198–200, Feb 1999.