Simple and Effective Adaptive Routing Algorithms Using Multi-Layer Wormhole Networks

Kyung Min Su, Ki Hwan Yum
Department of Computer Science
The University of Texas at San Antonio
San Antonio, TX 78249
Email: {ksu, yum}@cs.utsa.edu

Abstract— **Interconnection networks have been adopted in multicomputer systems, clusters, or chip multiprocessors (CMPs). Among various routing algorithms in interconnection networks, adaptive routing shows the best performance with most traffic types. In this paper, we propose new adaptive routing algorithms considering the remaining hops in addition to local network status. The proposed algorithms make adaptive decisions only when the remaining hops are less than some threshold and congestion is detected, or they do oblivious routing in other cases. As a result, the number of adaptive decisions is greatly reduced. Consequently our proposed algorithms have less adaptive overhead.**

We propose two practical adaptive routing algorithms which utilize the pipelined router architecture and multi-layer networks. The first proposed scheme is called Adaptive Injection. Since it does not affect the virtual channel allocation stage, its pipeline length is the same as non-adaptive routing pipelines. Adaptive Injection is especially good when the network size is small, because it has limited adaptability. While the first scheme has non-overhead but small improvement, the second adaptive routing algorithm we propose, called Adaptive Layer Selection, has some processing overhead and better performance.

The simulation results show that considering the remaining hops successfully decreases the number of adaptive decisions and two proposed routing schemes show better performance than previous adaptive algorithms. On the average, Adaptive Injection outperforms existing routing algorithms in terms of throughput by 7.1% $\sim$ 65.2%. Adaptive Layer Selection with Adaptive Injection shows better performance especially when the network size is large. Its throughput is improved by 12.5% $\sim$ 73.8% in an (8 x 8) mesh network.

I. INTRODUCTION

Interconnection networks have been adopted to connect multiple elements in multicomputer systems, clusters, or even chip multiprocessors (CMPs) that require high performance network support. One of the major factors that affect performance in interconnection networks is the routing algorithm. Oblivious routing algorithms (including deterministic and random algorithms), which decide the path between the source and the destination regardless of the network state, can be simple but may not adapt to the network load fluctuation. On the other hand, adaptive routing algorithms adjust to these changes, but require more complex hardware and incur more overhead. Therefore, the motivation of this paper is to devise simple and effective adaptive routing algorithms that can be used in interconnection networks.

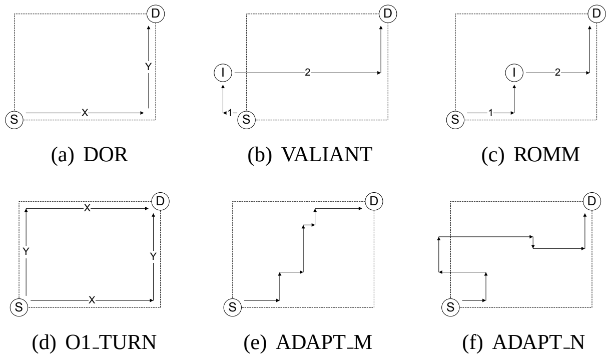
In this paper, we first consider using the remaining hops of a packet as a criterion for an adaptive routing decision. If the remaining hops are less than some threshold value, adaptive routing is performed. Otherwise we use oblivious routing. Using this simple scheme, we can greatly reduce the number of adaptive decisions in the network. Then we propose two new routing algorithms that combine oblivious and adaptive routing algorithms using multi-layer wormhole networks.

The first one, called Adaptive Injection (AI), does not incur any pipeline overhead, but performance improvement is small. Using the concept of multi-layer networks [1] in which different routing algorithms can be used at different layers in a network, a node adaptively selects the layer to which it injects a packet according to the current network status. After injection, the packet uses deterministic routing to be forwarded to its destination in the network. Because this scheme changes only the injection stage and the revised injection process does not increase the length of the router pipeline stages, it does not incur any pipeline overhead.

Next, we propose a deadlock-free layer transfer method, called Adaptive Layer Selection (AL), where a packet can change the layers during its delivery. With the proposed method, a packet can choose various routes in the multi-layer network, even after injection, by transferring from one layer to one of the other layers. So it can be regarded as another adaptive routing algorithm that is more complex and has better performance.

We compare two proposed routing algorithms with existing deterministic, random, and adaptive routing algorithms using simulations. Simulation results show that the first proposed routing algorithm, AI, has better performance than Dimension-Order Routing (DOR), deterministic routing and O1_TURN [1], random routing, without additional processing overhead. Especially, when the network size is small or the latency is measured with FO4 [2], [1], it shows the best performance. On the average, AI outperforms DOR by 65.2% and O1_TURN by 13.3%, respectively, in terms of throughput.

The second adaptive routing algorithm, AL, along with AI (AIAL) shows good performance when it is evaluated with the number of clock cycles or when the network size is large. Its throughput is improved by 73.8% than DOR and 19.6% than O1_TURN and 12.5% than DUATO [3], a fully adaptive routing algorithm, in an (8 x 8) mesh network.



(a) DOR (b) VALIANT (c) ROMM

(d) O1_TURN (e) ADAPT_M (f) ADAPT_N

Fig. 1. Routing Algorithms (S: Source Node, D: Destination Node, I: Intermediate Node, ADAPT_M: Minimal Adaptive Routing, ADAPT_N: Nonminimal Adaptive Routing)

Oblivious	Deterministic	Fig. 1 (a)
	Random	Fig. 1 (b),(c),(d)
Adaptive	Minimal	Fig. 1 (e)
	Non Minimal	Fig. 1 (f)

TABLE I

CLASSIFICATION OF ROUTING ALGORITHMS

The rest of the paper is organized as follows: In Section II, we discuss previous work on routing algorithms. Section III explains the basic router architecture. The proposed adaptive routing algorithms are presented in Section IV. Simulation results are shown in Section V, followed by the conclusions in Section VI.

II. PREVIOUS WORK

An oblivious routing algorithm differs from an adapitve algorithm in that it does not consider the network status. There are two oblivious routing algorithms: deterministic and random. Deterministic routing always makes the same path when the same source and destination nodes are given. Dimension Order Routing (DOR), shown in Fig. 1 (a), is the most common deterministic routing. Random routing adds a random node between the source and the destination nodes, and uses deterministic routing from the source node to the intermediate node and from the intermediate node to the destination node. Generally it shows better performance than DOR, since it distributes the traffic of the network evenly. Valiant [4], ROMM [5] and O1_TURN [1] are examples of random routing. As shown in Fig. 1 (b), (c) and (d), an intermediate node is randomly chosen within the whole network in Valiant methods [4], within the minimal rectangle in ROMM routing [5], and between the two corners of the minimal rectangle in O1_TURN [1].

There are many adaptive routing algorithms (Fig. 1 (e) and (f)) depending on how to choose the route from the source to the destination node with the given network conditions and how to prevent a deadlock [6], [3], [7], [8], [9], [10]. In the previous adaptive routing algorithms, they choose the next outport of each router with the following methods.

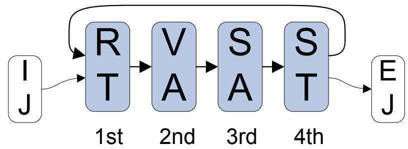


Fig. 2. Four Pipeline Stages (IJ: Injection, RT: Routing, VA: Virtual Channel Allocation, SA: Switch Arbitration, ST: Switch Traversal, EJ: Ejection)

- Avoid sharing the physical channel with other packets if possible [7].
- Choose the outport of the router that is Least Frequently Used (LFU) or Least Recently Used (LRU) [8].
- Choose the outport which has MAX-CREDIT or the shortest queue [8], [9].

Generally speaking, adaptive routing shows the best performance, but it requires a lot more processing overhead and needs additional resources like virtual channels (VCs) to avoid a deadlock [6], [3]. Adaptive routing algorithms are also usually suffering from wrong decision due to lack of global state information. Recently there are some studies to remedy the shortcomings of adaptive routing algorithms. To reduce the overhead of adaptive routing, [11] proposed DyAD routing that combines two routing schemes (deterministic and adaptive) according to the traffic congestion. To get better global load balancing, [9] proposed GOAL routing algorithm that makes global decision obliviously and local decision adaptively and [12] proposed the Regional Congestion Awarness (RCA) that propagates congestion information across the network.

Table I shows the classification of routing algorithms.

III. SYSTEM ARCHITECTURE

The router architecture used in this paper adopts a multi-layer four-stage pipelined wormhole model as shown in Fig. 2 and Fig. 3. In this architecture, if nodes inject new packets, the flits of the injected packets proceed to the destination by repeating the pipelined processes in Fig. 2. The demux moves the new flit from the inport to an appropriate VC queue according to the VC ID of the flit. If the head flit reaches the head of a VC, routing (RT) process checks the destination address of the flit and assigns an outport for the destination. Virtual channel allocation (VA) process assigns the inport's VC to one of the outport's available VCs. Even though there are some flits still using the VC, the VC can be assigned to other packets if it has some free space or credits because it is a wormhole router. The flits repeat the VA process until they are assigned to outport's VCs. The flits that succeed in VA perform switch arbitration (SA) process. Then the flits finally move to the next router through the switch traversal (ST) process. They repeat the same processes in the next node. If the flit reaches its destination node, it is ejected.

We also use the concept of multi-layer networks [1], a method to form one network with several independent layer of networks, each of which may have different features. Hence a multi-layer networks router can use different routing

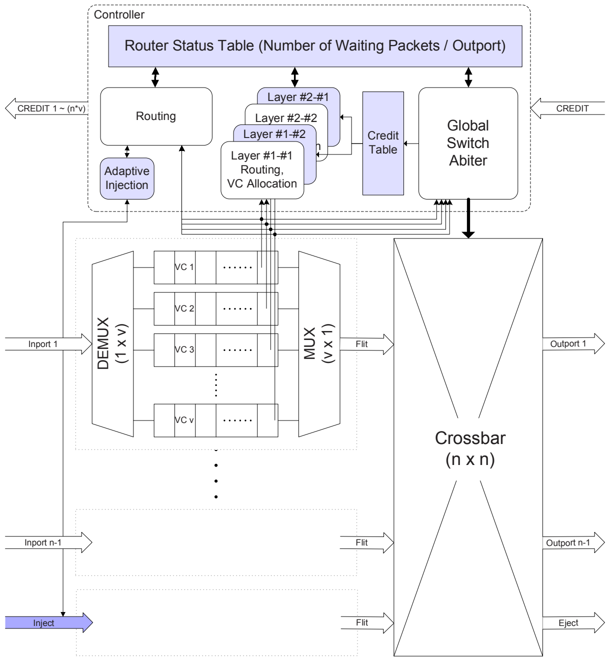


Fig. 3. Multi-Layer 4-Stage Wormhole Router Architecture

algorithms for each layer. Among pipeline stages in Fig. 2, generally VA stage takes the longest. Because the length of VA stage is decided by the number of VCs, if the router uses separate VA for each network, which has a part of virtual channels in a multi-layer networks router, it has shorter VA stage than a regular router [1]. Multi-layer networks can be implemented by assuming that the same VCs in each node consist one network and by applying different rules for each network. Since each network works as a separate network, if each network uses a deadlock-free routing algorithm, then the whole multi-layer networks is also deadlock-free [10]. There have been some routing proposals for multi-layer networks: O1_TURN [1], Positive-First-Negative-First (PFNF) [7] and Mad Postman[1] [10].

The detailed router architecture is shown in Fig. 3. In Fig. 3, the shaded parts are modified or added modules to the general multi-layer networks router model for the proposed schemes. And the details of the shaded modules are explained in Section IV.

IV. Proposed Schemes

In this section, adaptive decision with considering remaing hops is explained. Then two new adaptive routing algorithms are introduced: Adaptive Injection, adaptive routing that makes adaptive decision only in the injection stage, and Adaptive Layer Selection, adaptive routing that makes adaptive decisions in every intermediate nodes.

A. Adaptive Routing with Considering the Remaining Hops

The main reason previous adaptive routing algorithms make wrong decisions is that they only know local information around a node, without the global state of the network. To reduce the possibility of wrong decision; we propose that a router makes adaptive decisions with considering both the network status and the remaining hops. That is, the router makes adaptive decisions only when the remaining hops are less than a certain threshold value and congestion is detected. If the remaing hops are greater than the threshold, it uses an oblivious routing algorithm which does not consider the network status. For example, when the threshold is 5, if remaining hops from the current node to the destination node is 7, the router applies oblivious routing to decide the next route. After proceeding some hops, if remaing hops to the destination node is less than 5, the router uses adaptive rouitng for the next route. With this scheme we can reduce the number of wrong adaptive decisions which are made with insufficient information about the network.

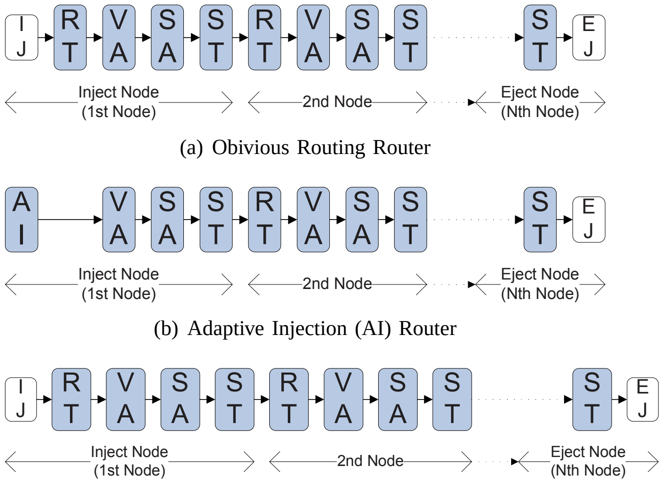
B. Adaptive Injection (AI)

In a multi-layer network, a new packet should select one of the layers for its injection. Although this injection layer for a new packet can be selected randomly [1], we propose a new injection scheme where a new packet selects a injection layer adaptively according to the current status of the node. Specifically a new packet is injected to a layer that has the least congested outport. In Fig. 3, three router components (*Inject, Adaptive Injection and Router Status Table*) are related to this scheme. If a new packet arrives at the *Inject* Port, *Adaptive Injection* decides an injection layer for the new packet according to the routing results and the current router status. Finally the new packet is injected to the selected layer. Note that since AI decides the outport through which the packet is forwarded, no further routing decision is required in the node as shown in Fig. 4 (b). This scheme has the following characteristics: no overhead, limited adaptability and deadlock-freedom.

1) No Overhead: Every flit proceeds by repeating the 4 pipeline stages in Fig. 2 from the inject node to the eject node. In our research, overhead implies more pipeline stages or longer pipeline stages that increase actual packet delivery latency. In other words, if a new architecture has the same number and the same length of pipeline stages as the original one, it means that the new architecture does not incur overhead.

Fig. 4 (a) shows the repeated processes of an oblivious router, while Fig. 4 (c) shows a general adaptive router. Most adaptive routing algorithms require the more complex and longer VA stage than that of oblivious routing algorithms since they have more candidate VCs for new packets. Because the VA stage is the longest stage among 4 pipeline stages, the longer VA stage means a longer pipeline stage and more packet delivery time as shown in Fig. 4 (c) for the same number of hops [1].

[1]Mad Postman [10] uses the different terms for multi-layer networks: *virtual network* for each layer and *virtual networks* for multi-layer networks.



(a) Obivious Routing Router

(b) Adaptive Injection (AI) Router

(c) Adaptive Routing Router

Fig. 4. Pipeline Stages from Injection to Ejection

Fig. 4 (b) shows the AI router. Since AI performs routing and an adaptive decision for a new packet in the injection process, its injection module is more complex than that of an oblivious routing algorithm. But AI has the *same length* of pipeline stages with those of the oblivious router, because it does not need to modify RT, VA, SA and ST stages and only affects the injection stage. Since AI does not add any processing step and does not lengthen any pipeline stages, it does not incur overhead.

2) Limited Adaptability: Since the proposed router model uses DOR as its basic routing algorithm, a packet should follow a deterministic path after injection. However, since this model can adaptively select the injection layer, it has adaptability to choose a path for packet delivery even though the chance is limited to only one time when the packet is injected.

Fig. 5 shows the limited adaptability of an AI router that uses DOR after injection for each layer. If we assume that one packet, Packet 1, is traversing from Node 10 to Node 02 in Fig. 5 (a), and another new packet, Packet 2, is just injected from Node 11 in Fig. 5 (b), Packet 2 will choose a less congested port as shown in Fig. 5 (c). It shows the same performance with full munimal adaptive routing in Fig. 5 (e). But if the Packet 2 is traversing in the network earlier than the Packet 1 , Packet 1 can not change its route, so it uses the same physical channel of node 12 as shown Fig. 5 (d), because it follows deterministic routing after injection. It is the limitation of AI. To remove this restriction, we will propose Adaptive Layer Selection, another complement adaptive selection scheme, in Section IV-C.

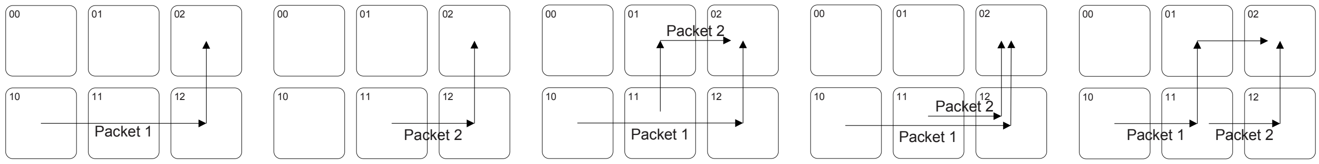
3) Deadlock-Freedom: To incur a deadlock, there should be a cycle of requesting and waiting resources in the network [6]. Our proposed scheme is an adaptive routing, but since no packets wait for inject buffer, the cycle for a deadlock cannot be formed with AI. Hence, AI does not incur any deadlock, it does not need any additional operations or resources to prevent or detect a deadlock.

C. Adaptive Layer Selection (AL)

If a transition between layers is allowed in the multi-layer network that uses a different routing algorithm for each layer, a packet can adaptively select the next path to the destination by selecting a different layer at each node. In Fig. 3, *Virtual Channel Allocation, Credit Table and Router Status Table* are related to Adaptive Layer Selection. The router in Fig. 3 has two layers. The shaded *Virtual Channel Allocation* allocates the new packet virtual channels from one layer to another layer. Before allocating a VC, the router compares the layers available for the new arrived packet. If a layer is less congested than the current layer and it has enough buffer space, the layer is selected as the next traversal layer. If there is no layer that has enough buffer space for layer transition, the current layer is selected to the next traversal layer. According to the next traversal layer, the packet uses different *Virtual Channel Allocation* module. Unlike AI, the routing algorithm in this section has some processing overhead because it uses more complex *Virtual Channel Allocation*. But this algorithm offers more routes because adaptive decisions can be made at all intermediate nodes. This scheme can be applied to all multi-layer routers which consist of multi-layers with a different routing algorithm for each layer, and it has following advantages:

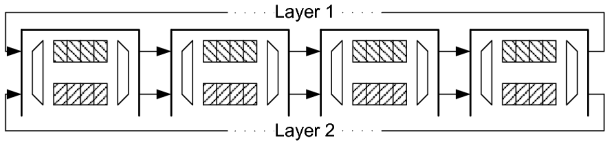
- Duato's adaptive routing [3] allows only one-way transition from the adaptive layer to the deadlock-free layer. But our proposed routing allows packet transition to any layer.
- Since it is a deadlock-free routing scheme, it does not need additional processing to detect a deadlock. Only, during the layer transition, the related buffers are held temporarily to guarantee deadlock-free of routing.
- When some layers are broken due to faulty nodes, networks can still work with other available layers.

1) Layer Transition without a Deadlock: With temporary buffer holding, a packet can safely move from one layer to another layer. A deadlock occurs when one packet, Packet 1, requests a resource which is used by another packet, Packet 2, and Packet 2 needs resources which are held by Packet 1 at the same time. Even though there is no deadlock in each layer of a multi-layer router, the packets across the layers can incur deadlocks with wormhole routing. This case is shown in Fig. 6. In the figures each row represents one layer of the multi-layer network. The shaded packets in Fig. 6 (b) incur a deadlock because the left shaded packet holds a flit in Layer 1 and is waiting for a free space in Layer 2 while the right shaded packet holds a flit in Layer 2 and is waiting for a free space in Layer 1. To prevent a deadlock, only when there is enough buffer space, which is one packet length, available in the next node, the packet can try layer transition by holding the buffers first. If a packet changes layers after reserving enough space in the buffer of the new layer, a deadlock will not occur during layer transition, because "hold and wait" never occurs.

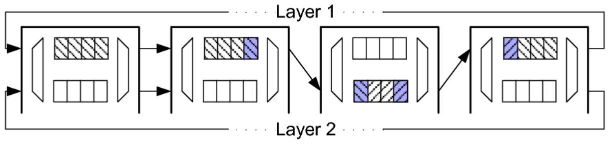


(a) DOR (b) DOR (c) AI: Packet 1 first (d) AI: Packet 2 first (e) Fully Adaptive

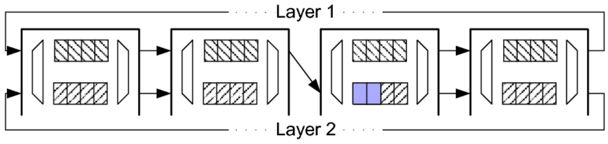
Fig. 5. Limited Adaptability of AI



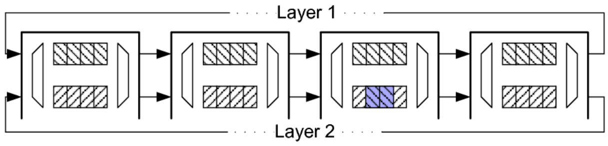
(a) 2 Layer Networks



(b) A Deadlock



(c) Buffer Holding and Layer Transition



(d) After Layer Transition

Fig. 6. A Deadlock and Layer Transition (Each packet has two flits.)

When the packet proceeds in the same layer, it does not need to hold any buffer because each layer is deadlock-free and they use wormhole routing. After completion of layer transition, the packet uses the routing algorithm used in that layer; so there is no deadlock as well. Fig. 6 (c) and (d) show the layer transition with buffer holding. If a packet waits for enough buffer space infinitely for layer transition, this results is a starvation. To avoid the starvation, if the buffer of other layer is not enough for layer transition, the packet should proceed in the same layer without waiting.

2) Packet Length vs. Buffer Length: It should be considered what happens when the packet length is longer than the length of a VC. The layer transition always holds the packet length of the buffer first. Therefore, if the initial length of a VC is shorter than the packet length, there will be no layer transition. To relieve this problem, the router can use dynamic buffer allocation [13] [14] at the input buffers. Since dynamic buffer allocation (or buffer sharing) allows using other channel's buffer, the long packet, which is longer than a VC, can hold the needed buffers for layer transition.

Topology	Mesh (4 x 4, 8 x 8, 16 x 16)
Traffic Types	Uinform,Hotspot, Transpose,Reversal, Shuffle,Complement [6]
Number of Virtual Channels	4, 8
Number of Physical Channels	5
Input Buffer Size	5 (flits)
Packet Size	5 (flits)

TABLE II
SIMULATION CONFIGURATION

V. PERFORMANCE EVALUATION

A. Platform

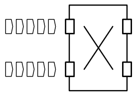
We have developed detailed wormhole router simulator models using C language. To check the characteristics of the proposed schemes, they have been tested with various traffic types. Table II shows the details of the simulation configuration. Among six traffice types, hot-spot traffic has four hot-spot nodes and 10% of traffic is towards the hot-spots. Table III shows the summary of test router models. We have tested three existing router models (two oblivious routings:DOR, O1_TURN and one adaptive routing: DUATO) and two proposed models: AI and AIAL. DOR is a dimension-order router which sends packets to X direction first and Y direction later. O1_TURN is a random routing router which randomly selects an intermediate node between the two corners of the minimal rectangle [1]. DUATO routing algorithm [3] is one of common adaptive that uses additional VCs to resolve a deadlock. AI and AIAL are proposed routers. AI is a router which adopts only Adaptive Injection scheme. AIAL router takes up both proposed schemes: Adaptive Injection and Adaptive Layer Selection schemes. In Table III, each pipeline router has different pipeline length according to the its routing algorithm [1]. The pipeline length of table III is measure by FO4[2]. Since AI routing algorithm does not change pipeline length of it, AI router has same pipeline length with O1_TURN, which injects new packets randomly. And the pipeline length of AIAL is assumed to be same with that of DUATO, most complex routing algorithm among tested models.

[2]FO4 is a unit which is used to measure the delay of circuit. [1] [2]

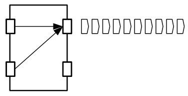
Name	Type	Pipeline Length
DOR	Deterministic Routing	20
O1_TURN	Random Routing [1]	17
DUATO	Adaptive Routing [3]	24
AI	Adaptive Injection	17
AIAL	Adaptive Injection and Adaptive Layer Selection	24

TABLE III

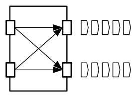
TEST ROUTER MODELS



(a) Input Packets

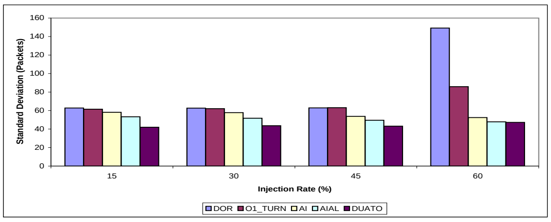


(b) Biased Outport Selection

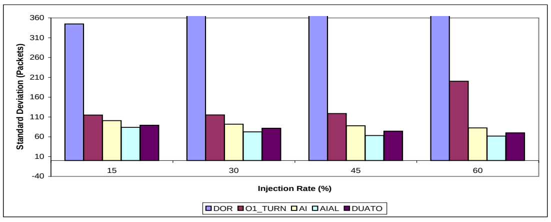


(c) Evenly Distributed Outport Selection

Fig. 7. Outport Selections



(a) Hotspot



(b) Transpose

Fig. 8. Average Traffic Distribution among Outports at Each Node

Traffic	DOR	O1_TURN	DUATO	AI	AIAL
UNIFORM	77.3	74.6	71.3	80.5	81.8
HOT-SPOT	55.8	61.2	62.9	64.8	71.4
TRANSPOSE	27.0	54.8	70.0	82.2	77.3
SHUFFLE	45.2	59.8	81.5	60.3	71.3
REVERSAL	26.8	54.2	61.9	62.5	70.9
COMPLEMENT	40.5	40.8	30.2	40.8	40.7

TABLE IV

THROUGHPUT WITH VARIOUS TRAFFICS (8 X 8, 4 VCS, THRESHOLD FOR AI = 6 HOPS, AIAL = 8 HOPS)

B. Outport Selection Function

The latency of a packet in a network is directly affected by the time needed to go through the crossbar. In Fig. 7 there are two extreme cases of traffic distribution for router outports. If we assume five packets for each inport as shown in Fig. 7 (a), Fig. 7 (b) shows the biased outport selection and Fig. 7 (c) shows the evenly distributed selection. If we assume that a packet needs one cycle to cross the switch, the average time required to cross the switch for the 10 packets is 5.5 cycles in Fig. 7 (b) and 3 cycles in Fig. 7 (c), respectively. Hence, evenly distributed packets needs less time to pass the crossbar.

To make more evenly distributed packet assignment, the proposed router chooses the next outport that has the least number of waiting packets.

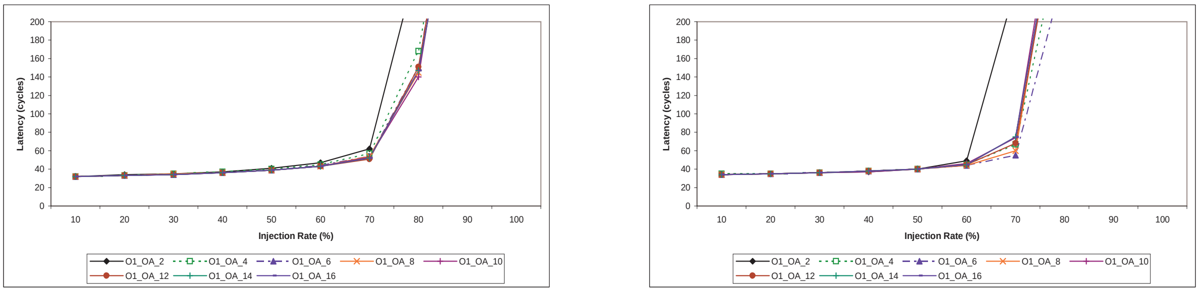
C. Simulation Results

Fig. 8 shows the average traffic distribution among the outports at each node with various routing algorithms. The graph shows that our outport selection function explained in Section V-B makes more evenly distributed traffic than the deterministic routing algorithm, DOR and the random routing algorithm, O1_TURN.

Fig. 9 shows the average latency with various thresholds, the dividing remaing hops between oblivious routing and adaptive rouitng. When the threshold is 2, the router shows the worst performance. As the threshold is changed, their perfomance is also changed. When the threshold is over than 8, its performance is not changed no more. That is, the performance of the router with threshold 8 is similar with other router which has the threshold 16. In this paper, more adaptive decision means more overhead. Therefore we decide the threshold as low as possible by repeated experiments with various threshold.

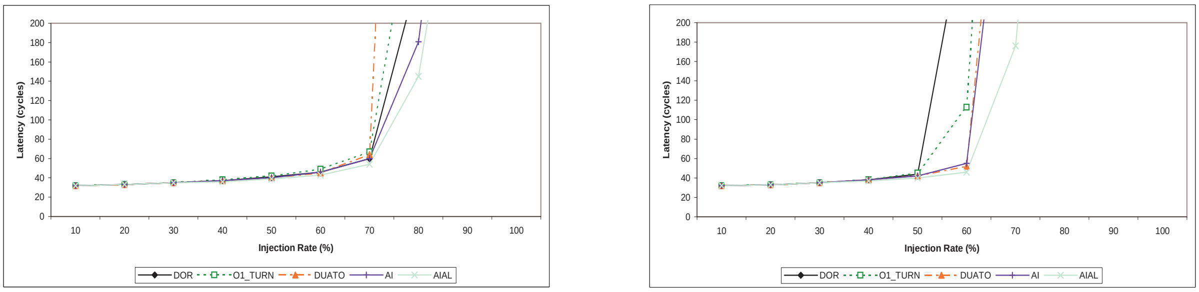
Fig. 10 and Table. IV show average latency and throughput of various routing algorithms when the threshold for AI is 6 hops and the threshold for AIAL is 8 hops. With uniform, hotspot and reversal traffics, AIAL shows the best performance. With transpose and complement traffics, AI shows the best performance. With transpose traffic, though AIAL does not show the best performance, it is still better than previous routing algorithms: DOR, O1_TURN and DUATO. Only with complement traffic, adaptive routing algorithms, DUATO and AIAL, show a little bit worse performance than other routing algorithms. Two adaptive routing algorithms, DUATO and AIAL, show similar performance when the number of VCs is 4 or 8, which is enough VCs for DUATO. On the average, AI outperforms existing routing algorithms in terms of throughput by 65.2% than DOR, 13.3% than O1_TURN and 7.1% than DUATO. The throughput of AIAL is improved by 73.8% than DOR, 19.6% than O1_TURN and 12.5% than DUATO, respectively.

AI with limited adaptability generally shows the performance between the full path adaptive routing and non-adaptive routing. When we consider no overhead of the AI router,

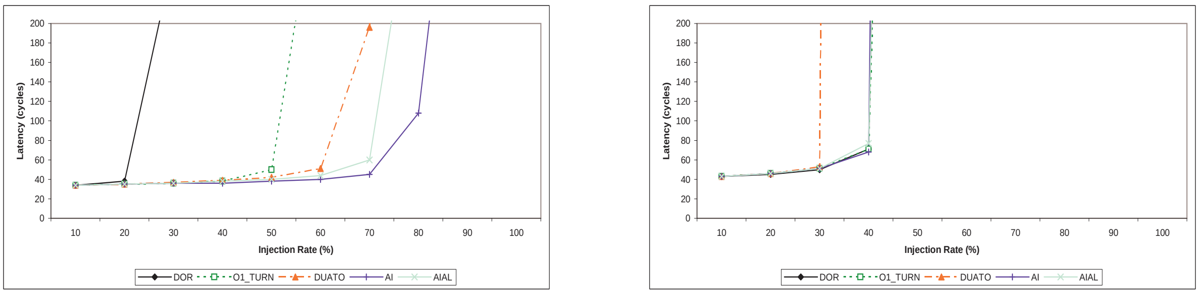


(a) Uniform (b) Transpose

Fig. 9. Average Latency with Various Threshold Values (8 x 8)



(a) Uniform (b) Hotspot



(c) Transpose (d) Complement

Fig. 10. Average Latency with Various Traffics (8 x 8, 4 VCs, Threshold for AI = 6 hops, AIAL = 8 hops)

the little improvement of the AI router with some traffic can have more significant meaning than that of other adaptive routing algorithms which require much more overhead. Fig. 11 shows the absolute latency of each routings with FO4 units in Table III. In most cases, AI shows the best performance.

Fig. 12 shows the performances of proposed schemes with various network sizes: (4 x 4), (8 x 8) and (16 x 16). The Y axis is the relative throughput to the DOR routing. The throughput of AIAL increases as the network size increases, so when the network is 16x16, AIAL shows much better performance and higher throughput, than other routing algorithms. The throughput of AI is less changed than AIAL because the adaptability of AI is not affected by the network size because AI has adaptivity only at an injection node. Therefore, if the network size is small, AI is the better choice than AIAL, because AI has less overhead than AIAL.

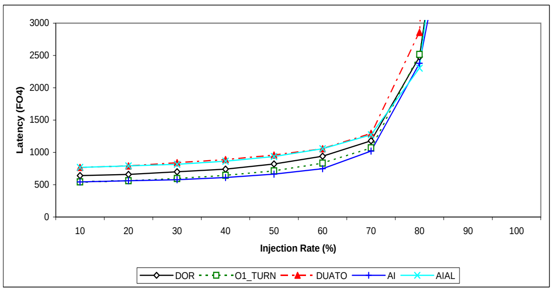
The case when a packet consists of more flits than the VC length was also simulated to verify correct working of adaptive layer selection with dynamic buffer allocation. The result is not shown here since it shows similar results with Fig. 10.

VI. Conclusions

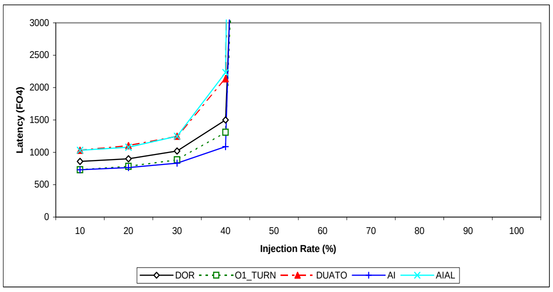
In this paper we propose new criterion for adaptive decision, remaing hops, and two new adaptive routing algorithms: Adaptive Injection (AI) and Adaptive Injection and Adaptive Layer Selection (AIAL).

To reduce adaptive overhead and wrong decision, we can apply the new criterion, the remaining hops, to adaptive routing algorithms.

The first proposed adaptive routing, AI, is on the adaptive selection of the inject network in multi-layer networks. The proposed scheme works like adaptive routing without any overhead by processing the adaptive decision in the injection stage of the router, which is before the repeated pipeline stages of the router. Simulation results show that AI has better performance than DOR and O1_TURN for all traffic types and

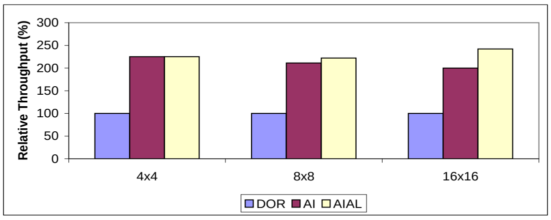


(a) Uniform

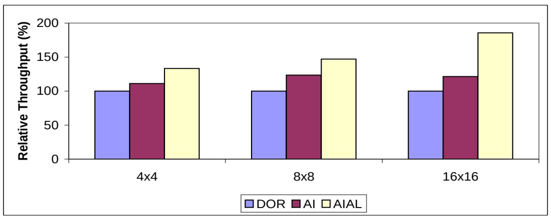


(b) Complement

Fig. 11. Absolute Latency (in FO4 unit, 8 x 8, 8 VCs)



(a) Reversal



(b) Shuffle

Fig. 12. AI vs. AIAL (4 x 4, 8 x 8, 16 x 16)

shows the better performance than DUATO for transpose and complement traffic. When the latency is measured with FO4 unit, absolute latency unit, to compare its process overhead with other models, AI shows the best performance in most traffic types. When the performance is evaluated with various network sizes, AI shows better performance with small networks since its limited adaptability, which implies its adaptive decision is made only at the injection node.

The second adaptive algorithm, AIAL, is temporary changing of flit flow control method from wormhole to virtual cut-through in order to transfer a packet between different layers of the multi-layer networks without a deadlock. Because each layer adopts a different routing algorithm, the packet can take various routes by selecting different layers at each intermediate node. AIAL shows better performance than AI when the latency is measured with cycle unit, which is more general unit for latency measurement, because AIAL has more adaptability than AI. Especially AIAL shows better performance than AI with large networks. When AIAL is compared with another adaptive routing, DUATO, AIAL shows better performance when the number of VCs is less than 4 because AIAL does not have the restriction on the direction of layer transition and does not reserve any VCs to resolve a deadlock.

Therefore, AI is the best choice to improve the performance of a router when the process overhead is critical and the network size is small, while AIAL can be used to make a more adaptable router or for larger networks.

REFERENCES

[1] D. Seo, A. Ali, W.-T. Lim, N. Rafique, and M. Thottethodi, "Near-optimal worst-case throughput routing for two-dimensional mesh networks," in *ISCA '05: Proceedings of the 32nd Annual International Symposium on Computer Architecture*, (Washington, DC, USA), pp. 432–443, IEEE Computer Society, 2005.

[2] L.-S. Peh and W. J. Dally, "A delay model and speculative architecture for pipelined routers," in *HPCA '01: Proceedings of the 7th International Symposium on High-Performance Computer Architecture*, (Washington, DC, USA), p. 255, IEEE Computer Society, 2001.

[3] J. Duato, "A new theory of deadlock-free adaptive routing in wormhole networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 4, no. 12, pp. 1320–1331, 1993.

[4] L. G. Valiant and G. J. Brebner, "Universal schemes for parallel communication," in *STOC '81: Proceedings of the thirteenth annual ACM symposium on Theory of computing*, (New York, NY, USA), pp. 263–277, ACM Press, 1981.

[5] T. Nesson and S. L. Johnsson, "Romm routing on mesh and torus networks," in *SPAA '95: Proceedings of the seventh annual ACM symposium on Parallel algorithms and architectures*, (New York, NY, USA), pp. 275–287, ACM Press, 1995.

[6] W. Dally and B. Towles, *PRINCIPLES AND PRACTICES OF INTERCONNECTION NETWORKS*. Morgan Kaufmann publishers, 2004.

[7] J. Upadhyay, V. Varavithya, and P. Moihapatra, "A traffic-balanced adaptive wormhole routing scheme for two-dimensional meshes," *IEEE Trans. Comput.*, vol. 46, no. 2, pp. 190–197, 1997.

[8] A. S. Vaidya, A. Sivasubramaniam, and C. R. Das, "Lapses: A recipe for high performance adaptive router design," in *HPCA '99: Proceedings of the 5th International Symposium on High Performance Computer Architecture*, (Washington, DC, USA), p. 236, IEEE Computer Society, 1999.

[9] A. Singh, W. J. Dally, A. K. Gupta, and B. Towles, "Goal: a load-balanced adaptive routing algorithm for torus networks," in *ISCA '03: Proceedings of the 30th annual international symposium on Computer architecture*, (New York, NY, USA), pp. 194–205, ACM Press, 2003.

[10] C. R. Jesshope, P. R. Miller, and J. T. Yantchev, "High performance communications in processor networks," in *ISCA '89: Proceedings of the 16th annual international symposium on Computer architecture*, (New York, NY, USA), pp. 150–157, ACM Press, 1989.

[11] J. Hu and R. Marculescu, "Dyad: smart routing for networks-on-chip," in *DAC '04: Proceedings of the 41st annual conference on Design automation*, (New York, NY, USA), pp. 260–263, ACM, 2004.

[12] P. Gratz, B. Grot, and S.W.Keckler, "Regional congestion awareness for load balance in networks-on-chip," 2008. To appear in the Proceedings of the 14th International Symposium on High Performance Computer Architecture.

[13] C. Nicopoulos, D. Park, J. Kim, N. Vijaykrishnan, M. S. Yousif, and C. R. Das, "Vichar: A dynamic virtual channel regulator for network-on-chip routers.," in *MICRO*, pp. 333–346, IEEE Computer Society, 2006.

[14] Y. Tamir and G. L. Frazier, "Dynamically-allocated multi-queue buffers for vlsi communication switches," *IEEE Trans. Comput.*, vol. 41, no. 6, pp. 725–737, 1992.