The Impact of Path Selection Algorithm of Adaptive Routing for Implementing Deterministic Routing

Michihiro Koibuchi   Akiya Jouraku   Hideharu Amano

Dept. of Information and Computer Science
Keio University
3-14-1 Hiyoshi, Kohoku-ku, Yokohama, 223-8522 Japan
{*koibuchi,jouraku,hunga*}*@am.ics.keio.ac.jp*

Abstract

In PC clusters or high performance I/O networks including InfiniBand, network topologies often become irregular. Although various adaptive routings for irregular networks have been proposed, most of such commercial or experimental networks use a deterministic routing which enables a simple switch structure and in-order packet delivery. A strategy of path selection algorithm which fixes a single path among alternative paths between each pair of switches(hosts) is required but only a few studies have been asserted. In this paper, we propose three path selection algorithms which have different concepts using a static analysis of routing path to distribute the traffic, and investigate the influences of path selection algorithms on the throughput. Result of simulations shows that the throughput of each path selection algorithm depends on routing algorithm and topology, and the path selection algorithms using a static analysis of routing path achieves higher throughput compared with one without using it.

Keywords Irregular networks, deterministic routing, adaptive routing, deadlock avoidance, networks of workstations

1 Introduction

Switch-based irregular networks are commonly used in high performance distributed computing systems with commodity personal computers[12],[14],[15] and also in high performance I/O networks including InfiniBand[2]. Adaptive routing techniques for irregular networks have been widely studied[1],[9],[6], and their superior performance compared with deterministic routings have been demonstrated. Nevertheless, a lot of real networks only support deterministic routings because of the following reasons: (1) in-order packet transfer property is important in PC (Personal Computer) networks, (2) once a system trouble occurs, it is hard to trace adaptive routed packets in complicated irregular network, and (3) switch structures with a simple control mechanism are preferred in irregular networks. Also in InfiniBand, although multiple paths can be selected between CA (Channel Adapter)s in a sub-net, a deterministic routing with tables in each switch is mostly used. This comes from that the source CA indicates a path via its selection of them[2],[13].

In order to apply an existing adaptive routing, such as, up*/down* routing[8] or L-turn routing[9], to such real irregular networks, a policy of path selection which chooses a path among alternative paths between each pair of switches(hosts) is essential to performance. In this paper, such a policy is called "path selection algorithm". Path selection algorithm considers the traffic distribution regardless of guarantee of deadlock-free because adaptive routing guarantees deadlock-free, and it would be a key to implement a deterministic routing using techniques for adaptive routings in irregular networks [1] .

Unfortunately, only a few researches into path selection algorithm have been done[4], and the impact to the performance has not been well analyzed. So, when designing a deterministic routing for real irregular networks based on techniques used in an adaptive routing, it is difficult to select a suitable path selection algorithm.

Here, we present three path selection algorithms, which use static analysis results of routing path by different manners. The performance evaluation results with computer simulation are shown to demonstrate their efficiency.

[1] Noting that path selection algorithm can not apply to a special adaptive routing called Silla's minimal routing because it guarantees deadlock-free through selecting a path between original channel(deadlock-free path) and new channel(fully adaptive path) dynamically.

2 Existing path selection algorithms

An adaptive routing is a technique to select a route of packet dynamically, and so it can dynamically avoid the network congestion. However, in order to implement a deterministic routing, a path selection algorithm must be applied to an adaptive routing for fixing a single path from alternative paths, and it can not dynamically avoid the network congestion. Nevertheless, path selection algorithms are essential to performance since it can mitigate the congestion around the hot spot in most case if well-distributed paths are set.

The simplest path selection algorithm is random selection. Another simple one selects a path for the port with smaller port-ID when more than two channels are available in a switch. In this paper, this is called "low port first". However, above two path selection algorithms possibly select a path to congestion points even if there exist some candidates which can avoid it.

To address this problem, traffic balancing algorithm using a static analysis of routing path is proposed by Sancho[4] as follows.

1. All possible routing paths between every pair of switches are calculated. Then, this algorithm associates a counter to every channel, and each counter is initialized to the number of routing paths crossing the channel.

2. A routing path crossing the channel with the highest value of counter is selected to be removed if there is more than one routing paths between the source and the destination switches of it. If there is more than one routing path which can be removed in a channel, the routing path whose source and destination hosts have the highest number of routing paths between them is selected.

3. When a routing path is removed, the counters associated with every channel crossed by the path are updated.

4. Repeat the procedure 2 until the number of routing paths between every pair of hosts is reduced down to the unit.

The time complexity to compute this traffic balancing algorithm is $O(n^2 * diameter)$, where n is the number of switches.

3 Path selection algorithms based on a static analysis of routing path

Although Sancho's traffic balancing algorithm[4] is an efficient method based on a static analysis of routing path to distribute the traffic, there are other concepts worth to try.

In this section, we present three novel path selection algorithms: "high physical channel first", "low virtual channel first", and "low physical channel first". These have the same procedure flow as Sancho's traffic balancing algorithm, but the step 2 is different from Sancho's one as follows.

- *High physical channel first* selects the virtual channel with the highest value of counter on the physical channel with the highest value of sum of its virtual channel's counters, and removes the routing path on it if there is more than one routing paths between the source and the destination switches of this routing path. If there is more than one routing path which can be removed in a channel, the routing path whose source and destination hosts have the highest number of routing paths between them is selected to be removed.

- *Low virtual channel first* selects the virtual channel with the lowest value of counter, and fixes a routing path on it. That is, the other routing paths between the same source and the same destination switches are removed. If there is more than one routing path which is still not fixed in a channel, a routing path crossing the channel with the lowest value of counter is selected.

- *Low physical channel first* selects the virtual channel with the lowest value of counter on the physical channel with the lowest value of sum of its virtual channel's counters, and fixes a routing path on it. If there is more than one routing path which is still not fixed in a channel, a routing path crossing the channel with the lowest value of counter is selected.

Sancho's traffic balancing algorithm is designed for a network with a few virtual channels. However, virtual channels which can use the physical channel in time-sharing manner are plentifully equipped in recent switches, and the congestion of physical channels will tend to be a problem in such networks. So, *high physical channel first* is designed so as to avoid the physical channel bottleneck as well as virtual channels. On the other hand, *low virtual channel first* tries to use all virtual channels efficiently by avoiding a virtual channels with extremely small utilization. Note that Sancho's traffic balancing algorithm means *high virtual channel first* in the point of the selection policy.

Figure 1 shows an example of the irregular network with five switches using one bidirectional channel between switches, and up*/down* routing[8] is applied on it. which is a typical partially adaptive routing. In up*/down* routing, a packet must be transferred by using the channels which face to the root (if needed) followed by the channels which go away from the root(if needed) in order to avoid deadlocks. This restriction prevents a packet from turning from down direction to up direction.

In Figure 1, the value of counter to each channel is calculated according to the number of routing paths crossing it. For example, the value of the counter on the channel from a to c is four because $(a, e), (a, c), (b, c), (d, c)$ are crossing it, where (x, y) is the routing path from x to y. When implementing a deterministic routing, (a, e) and (e, a) have two candidates$(a \rightarrow b \rightarrow e), (a \rightarrow c \rightarrow e)$ and $(e \rightarrow b \rightarrow a), (e \rightarrow c \rightarrow a)$ respectively. Simple algorithms, *random* and *low port first* may select the former one which goes through the congestion channel from a or b to b or a respectively. On the other hand, the four path selection algorithms using a static analysis of routing paths select the latter one.

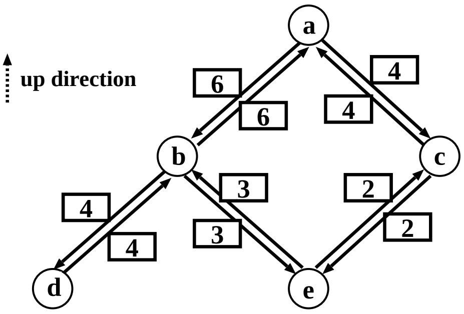


Figure 1. The example of counters to routing paths on up*/down* routing

Figure 2 shows the next example of the counter on up*/down* routing. Then, (a, e) and (e, a) have two candidates$(a \rightarrow b \rightarrow e), (a \rightarrow c \rightarrow e)$ and $(e \rightarrow b \rightarrow a), (e \rightarrow c \rightarrow a)$ respectively. In this case, Sancho's traffic balancing algorithm and *high physical channel first* select the former one, while *low physical channel first* and *low virtual channel first* select the latter one. This comes from that Sancho's one and *high physical channel first* try to remove the bottleneck channels, while *the low physical channel first* and *the low virtual channel first* are designed to avoid the channels with extremely small utilization.

4 Performance evaluation

In this section, performance of path selection algorithms on up*/down* routing or the UDWM[11] is evaluated by the computer simulation,

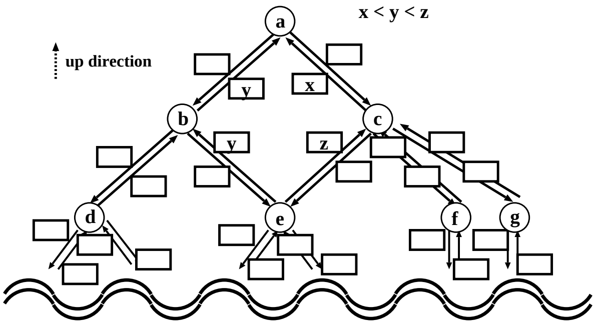


Figure 2. The example of counters to routing paths on up*/down* routing

4.1 Network model

A flit-level simulator written in C++ was developed for analysis. Topology, network size, and packet length are selected just by changing parameters. A switching fabric which provides eight bidirectional ports, using four ports to connect with hosts and remaining four ports for connecting other switches. Here, a simple model consisting of channel buffers, crossbar, link controller and control circuits is used for the switching fabric. 10 different topologies are randomly generated on condition that every different link must be connected with a different neighbor switches. A destination of a packet is determined by a traffic pattern used in the simulation. Here, the uniform traffic in which all destinations are selected randomly is used.

Simulation parameters are set as shown in Table 1.

Table 1. Simulation parameters

Simulation time	1,000,000 clocks (ignore the first 50,000 clocks)
Topology	irregular or torus
Network size	16 switches or 64 switches
The number of virtual channels	1 or 5
Packet length	128 flits
Adaptive routing	up*/down* or the UDWM
Switching tech.	virtual cut-through
Traffic pattern	uniform

4.2 Routing algorithms

Up*/down* routing Up*/down* routing is the most popular deadlock-free adaptive routing for irregular networks, and has been used in Autonet[8]. In order to guarantee connectivity and deadlock-free for irregular networks, up*/down* routing needs a spanning tree based directed graph in which up or down direction is assigned to each network channel. Several spanning trees for an irregular topology can be structured depending on the tree building and root selection policies. In this simulation, simple policies used in Autonet are applied, that is, the BFS (Breadth First Search) is used for building spanning trees, and the switch with identifier 0 is selected as the root. As mentioned in Section 3, a packet must be transferred by using the channels which face to the root (if needed) followed by the channels which go away from the root(if needed) in order to avoid deadlocks.

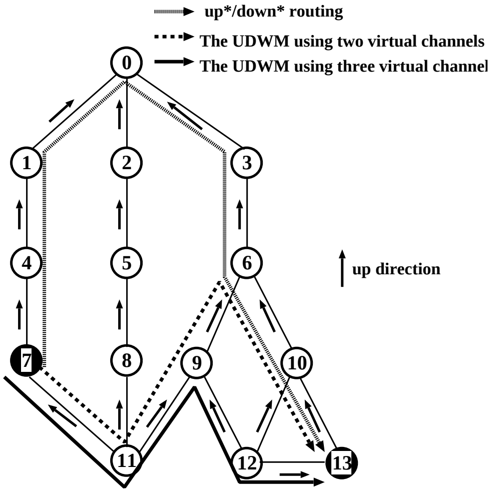

up*/down* routing
The UDWM using two virtual channels
The UDWM using three virtual channel
up direction
0
1
2
3
4
5
6
7
8
9
10
11
12
13

Figure 3. 14 switches irregular network

The UDWM The UDWM(up*/down* routing with multichannels) is an improved routing algorithm of up*/down* routing so as to best use of virtual channels[11]. The UDWM has the same restrictions except the following condition as up*/down* routing: The turn from down channel to up channel is only used with descending virtual channel number.

In the example shown by Figure 3, when a packet is transferred from **7** to **13** with up*/down* routing, the path takes 7 hops(**7**→**4**→**1**→**0**→**3**→**6**→**10**→**13**) regardless of the number of virtual channels. On the other hand, when the UDWM is used and each physical link splits into two virtual channels called "$ch.0$" and "$ch.1$" in Figure 3, the path takes only 5 hops(**7**→ $(ch.1)$→**11**→ $(ch.0)$ →**9**→ $(ch.0)$ →**6**→ $(ch.0)$ →**10**→ $(ch.0)$ →**13**) by decreasing a number of virtual channel. Moreover, when each physical link splits into three virtual channels called "$ch.0$", "$ch.1$", and "$ch.2$", the path of the UDWM takes only 4 hops(**7**→$(ch.2)$→**11**→ $(ch.1)$ →**9**→ $(ch.1)$ →**12**→$(ch.0)$ →**13**) by decreasing a number of virtual channel twice.

4.3 Simulation Results

4.3.1 16 switches irregular networks with 1 virtual channel

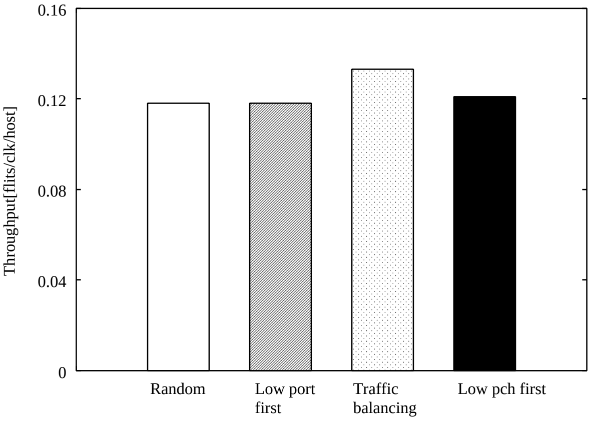


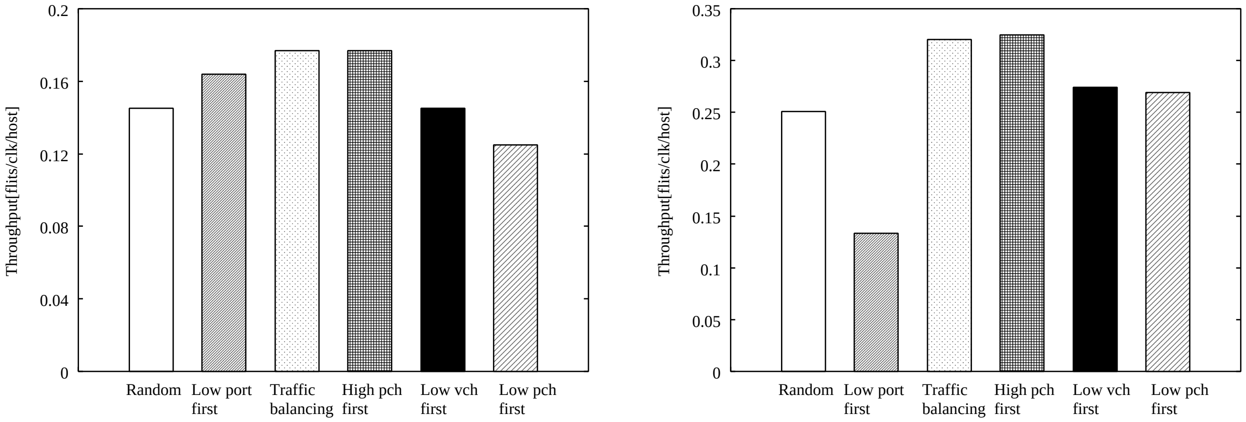
Figure 4. Throughput on 16 switches irregular networks with 1 virtual channel under up*/down* routing

Figure 4 shows average throughput of 10 irregular topologies with 16 switches. Here, throughput is defined as the maximum amount of accepted traffic. Accepted traffic is the flit reception rate in host in each clock cycle.

Table 2 shows standard deviation(SD) of channel crossing paths. Here, channel crossing paths are defined as the average number of routing paths crossing through any physical channel after selecting only one path between each pair of switches. It shows how uniformly the paths are distributed, that is, the small channel crossing paths mean that the paths are distributed uniformly.

In all figures and tables on this section, *high physical channel first, low virtual channel first,* and *low physical channel first* are shown as "high pch first", "low vch first", and "low pch first" respectively.

As shown in Figure 4, the throughput of Sancho's traffic balancing algorithm achieves better than *low vch first* a little. This comes from that Sancho's one distributes the routing paths more uniformly as shown in Table 2.



(a) Up*/down* routing

(b) The UDWM

Figure 5. Throughput on 16 switches irregular networks with 5 virtual channels

Table 2. Routing metric on 16 switches irregular networks with 1 virtual channel under up*/down* routing

Path selection algorithm	SD of channel crossing paths
Random	5.63
Low port first	5.61
Traffic balancing	5.06
Low vch first	5.39

Table 3. Routing metric on 16 switches irregular networks with 5 virtual channels

Path selection algorithm	SD of channel crossing paths	
	ud	UDWM
Random	5.54	2.87
Low port first	5.61	3.04
Traffic balancing	5.09	2.27
High pch first	5.07	2.23
Low vch first	5.38	2.70
Low pch first	5.52	2.75

4.3.2 16 switches irregular networks with 5 virtual channels

Figure 5 shows average throughput of 10 irregular topologies with 16 switches. The condition of simulation is the same in the before section except the number of virtual channels.

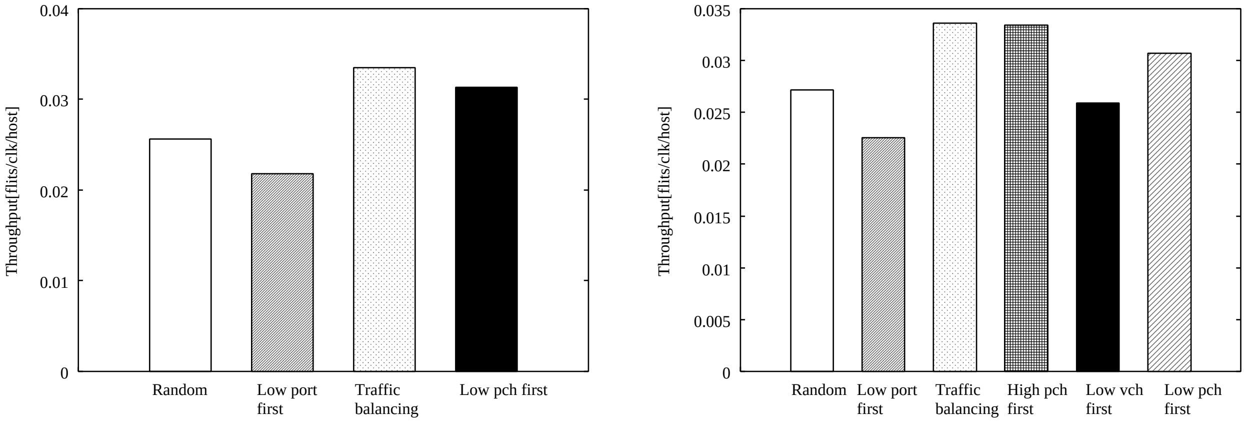
Figure 5 and Table 3 demonstrate that the throughput of each path selection algorithm is depending on the routing algorithm, but Sancho's one and *high physical channel first* outperform compared with *low physical channel first* and *lowest virtual channel first*. Consequently, in order to distribute the traffic, the methods to remove the bottleneck channels are more efficient than the methods to avoid the channels with extremely small utilization.

4.3.3 64 switches 2D torus

Figure 6 shows simulation results of 8×8 2D torus. The condition of simulation is the same in the above section except the network size and topology. Table 4 also shows routing metric in the case. As shown in Figure 6 and Table 4, the path selection algorithms using a static analysis of routing paths achieves higher throughput compared with ones without using it. From Figure 4, Figure 5, and Figure 6, the throughput of each path selection algorithm depends on routing algorithm and topology.

Figure 7 shows the distribution of channel utilization on 8×8 2D torus when the throughput shown in Figure 6 is obtained. Here, a switch whose switch number is $(0, 0)$ on the 2D torus is selected as the root.

Note that the uniform traffic is used in this simulation. Although 8× 8 torus is a uniform topology, all path selection algorithms tend to gather many packets around the root. This comes from that up*/down* routing essentially makes concentrated traffic around the root. Nevertheless, the path selection algorithms using a static analysis of routing path mitigate to this problem, and achieve high channel utilization.



(a) 1 virtual channel

(b) 5 virtual channels

Figure 6. Throughput on 64 switches 2D torus under up*/down* routing

Table 4. Routing metric on 64 switches 2D torus under up*/down* routing

Path selection algorithm	SD of channel crossing paths	
	1 vch	5 vch
Random	69.0	65.3
Low port first	71.7	71.7
Traffic balancing	54.1	54.1
High pch first	—-	53.8
Low vch first	61.0	64.0
Low pch first	—-	65.7

5 Related work

There are some researches on the path selection mostly for adaptive routings.

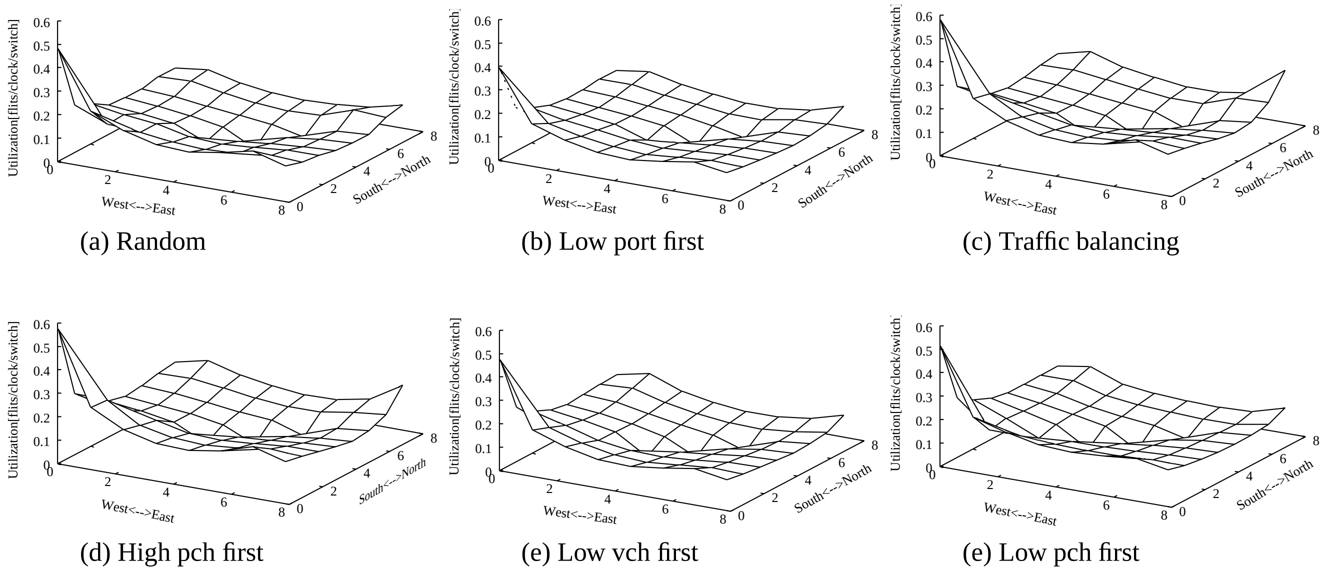
Output selection function In an adaptive routing, the output channel is dynamically selected depending on the condition of channels. For example, if a channel is being used (that is, in busy condition), the other channel has priority over the busy channel. However, if both output channels are not used (that is, in free condition), an output selection function decides the output channel[10],[7].

The output selection function is essentially required when an adaptive routing is implemented. On the other hand, path selection algorithm is required when a deterministic routing is implemented based on an adaptive routing. Although sophisticated output selection functions use a measure which indicates the congestion of each output channel, it decides the output only with the local data inside the switch[3],[10].

Source routing using dynamic selection of alternative paths There are basically two implementation of deterministic routing: the distributed routing and the source routing. In the source routing[12], all information of the path to destination is packed into the packet header in the source. Thus, each intermediate switch can determine the path only by referring the header information. In this case, the source can select a path among alternative paths dynamically. Simple examples of such selection policies are random selection and round robin[5]. However, using such policies, in-order packet transfer property is not guaranteed unlike the path selection algorithm treated here.

6 Conclusion

A path selection algorithm used in adaptive routings is also required in deterministic routing to select a path from possible multiple paths. In this paper, we present three path selection algorithms using a static analysis of routing paths in order to distribute the traffic more uniformly. Result of simulations shows that the throughput of each path selection algorithm depends on routing algorithm and topology, and the algorithms using a static analysis of routing paths achieves higher throughput compared with ones without using it. Policies attempting to remove the bottleneck channels are more efficient than ones to avoid the channels with extreme low utilization. We are planning to implement and evaluate path selection algorithms on a real system called RHiNET[15],[14], which is a network for cluster based parallel processing systems.



(a) Random (b) Low port first (c) Traffic balancing

(d) High pch first (e) Low vch first (e) Low pch first

Figure 7. Channel utilization on 64 switches 2D torus

References

[1] F.Silla and J.Duato. High-Performance Routing in Networks of Workstations with Irregular Toporogy. *IEEE Transactions on parallel and distributed systems*, 11(7):699–719, 2000.

[2] I.T.Association. Infiniband arch. specification volumen 1,release 1.0.a. *available from the InfiniBand Trade Association, http://www.infinibandta.com*, June 2001.

[3] J.C.Martinez and F.Silla and P.Lopez and J.Duato. On the Influence of the Selection Function on the Performance of Networks of Workstations. In *Proc. the 2000 International Symposium on High Performance Computing*, pages 292–300, Oct. 2000.

[4] J.C.Sancho and A.Robles. Improving the Up*/Down* Routing Scheme for Networks of Workstations. In *Proc. of EURO-PAR*, pages 882–889, Aug. 2000.

[5] J.Flich, M.P.Malumbers, P.Lopez, and J.Duato. Improving Routing Performance in Myrinet Networks. In *14th International Parallel and Distributed Processing Symposium(IPDPS)*, pages 27–32, 2000.

[6] A. Jouraku, M. Koibuchi, A. Funahashi, and H. Amano. Routing Algorithms on 2D Turn Model for Irregular Networks. In *Proc. of the Sixth International Symposium on Parallel Architectures, Algorithms, and Networks(I-SPAN)*, May.(to be appeared) 2002.

[7] Loren Schwiebert. A Performance Evaluation of Fully Adaptive WormholeRouting including Selection Function Choice. In *IEEE International Performance, Computing, and Communications Conference*, pages 117–123, Feb. 2000.

[8] M.D.Schroeder al et. Autonet: A high-speed, selfconfiguring local area network using point-to-point links. *SRC research report 59,DEC*, Apr. 1990.

[9] M.Koibuchi, A.Funahashi, A.Jouraku, and H.Amano. L-turn Routing: An Adaptive Routing in Irreglar Networks. In *Proc. of the International Conference on Paralel Processing(ICPP)*, pages 374–383, Sept. 2001.

[10] M.Koibuchi, A.Jouraku, A.Funahashi, and H.Amano. MMLRU selection function: An Output Selection Function on Adapt ive Routing. In *Proc. of ISCA 14th International Conference on Parallel and Distributed Computing Systems(PDCS)*, pages 1–6, Aug. 2001.

[11] M.Koibuchi, A.Jouraku, and H.Amano. A Deterministic Routing using Virtual Channels in Irregular Networks. In *Proc. of EURO-PAR 2002(submitting) or Joint Symposium on Prallel Processing(JSPP, In Japanese)*, May.(to be appeared) 2002.

[12] N.J.Boden et al. Myrinet: A Gigabit-per-Second Local Area Network. *IEEE Micro*, 15(1):29–35, 1995.

[13] P.Lopez, J.Flich, and J.Duato. Deadlock-free Routing in $InfiniBand^{TM}$ through Destination Renaming. In *Proc. of the International Conference on Paralel Processing(ICPP)*, pages 427–434, Sept. 2001.

[14] S.Nishimura, T.Kudoh, H.Nishi, J.Yamamoto, K.Harasawa, N.Matsudaira, and H.Amano. 64-Gbit/s Highly Reliable Network Switch Using Parallel Optical Interconnection. *IEEE Journal of Lightwave Technology*, 18(12):1620–1627, 2000.

[15] S.Nishimura, T.Kudoh, H.Nishi, J.Yamamoto, K.Harasawa, N.Matsudaira, S.Akutsu, K.Tasho, and H.Amano. RHiNET-3/SW: an 80-Gbit/s high-speed network switch for distributed parallel computing. In *Hot Interconnect 9*, pages 119–123, 2001.