Path Selection Algorithm: The Strategy for Designing Deterministic Routing from Alternative Paths

Michihiro Koibuchi [a], [1] Akiya Jouraku [a], Hideharu Amano [a]

[a] *Department of Information and Computer Science, Keio University*
3-14-1 Hiyoshi, Kohoku-ku, Yokohama 223-8522, Japan

Abstract

System Area Networks (SANs), which usually accept irregular topologies, have been used to connect nodes in PC/WS clusters or high-performance storage systems. Although routing algorithms for SANs usually find out alternative paths, SANs usually accept only deterministic routings. Thus, path selection algorithm, which chooses a single path from alternative paths, becomes essential for advanced routings in SANs. However, a few studies of it have been done only for SANs without virtual channels, and its impact is not well analyzed. In this paper, (1) we propose four path selection algorithms which have different concepts to distribute paths in SANs with virtual channels, and (2) we investigate the performance influences of various path selection algorithms through a flit-level simulation. Simulation results show that one of the four algorithms improves up to 92% of throughput against simple path selection algorithms, and policies to remove paths crossing the bottleneck channels are more efficient than ones to keep paths crossing channels that are not crowded.

Key words: Deterministic routing, System Area Networks, path selection algorithm, interconnection networks, virtual channels

1 Introduction

Network-based parallel processing using commodity components, such as personal computers, has received attention as an important parallel-computing environment [1] [2] [3] [4]. System Area Network (SAN), which consists of

[1] Corresponding Author. Address: Department of Information and Computer Science, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama 223-8522, Japan
Email address: koibuchi@am.ics.keio.ac.jp (Michihiro Koibuchi).

switches connected with point-to-point links, is one of the crucial components of such an environment. Unlike Local Area Networks (LANs), wormhole[5] or virtual cut-through[6] is used in SANs for low-latency direct-communication. When such methods are used, deadlock-free routings are required. On the other hand, unlike interconnection networks used in massively parallel computers [7] [8], SANs usually accept irregular topologies, because connection flexibility and robustness are preferred over the uniformity of interconnection networks. The irregularity of interconnection introduces difficulty on guarantee of connectivity and deadlock-free packet transfer. Thus, the following simple routings have been received attention as practical solutions: (1) spanning-tree based routings, which use the connectivity and acyclicity of tree structure (e.g. Up*/Down*[9], L-turn/R-turn[10]); (2) methods using virtual channels to eliminate deadlocks (e.g. structured buffer pools[11], LASH[12], descending layers (DL)[13]). Most of them originally have alternative paths (adaptivity) because they initially search alternative shortest paths under the conditions enough to avoid deadlocks.

On the contrary, even in recent switching fabrics, deterministic routings, which have a single path between each pair of switches, are preferred over adaptive routings because of the following advantages: (1) they guarantee in-order packet delivery, which message passing libraries usually require, without sorting packets at the destination host; (2) high-speed switching fabrics can be implemented, because dynamical choice of an output path is not needed. Thus, a policy, which chooses a single path from alternative paths, becomes essential for advanced routings in SANs, and we call such a policy "path selection algorithm". However, a few researches on path selection algorithms have been done only for SANs without virtual channels [14], and their impact to the performance has not been well analyzed.

In this paper, (1) we propose four path selection algorithms which have different concepts to distribute paths in SANs with virtual channels, and (2) we investigate the influences of various path selection algorithms on throughput and latency through a flit-level simulation. The proposed basic concepts can be applied to modern SANs, such as RHiNET-2[2] or QsNET[4] which have virtual channels.

The rest of the paper is organized as follow. Section 2 shows traditional path selection algorithms. In Section 3, we propose four path selection algorithms, which have different concepts to distribute paths uniformly. In Section 4, simulation results under a flit-level simulation are shown. In Section 5, related researches on path choice are described and compared with the path selection algorithm, and in Section 6, the conclusion is presented.

2 Path Selection Algorithm

Routing algorithms for SANs usually have alternative paths (adaptivity) originally, because they initially search alternative shortest paths under the conditions enough to avoid deadlocks. However, it can not be directly implemented on SANs that only allow deterministic routings. Thus, a path selection algorithm, which statically selects a single path from alternative paths, is required for such SANs [2]. Since it can not dynamically avoid network congestion, it seems that its impact is limited. Nevertheless, a path selection algorithm becomes essential for advanced routings in SANs because performance of each deterministic routing is seriously influenced by the distribution of paths.

The simplest path selection algorithm is random selection. Another simple one selects a path for the output port with smaller port-UID (unique identifier) when more than two output ports are available at a switch. In this paper, this is called "low port first".

If a path selection algorithm makes well-distributed paths, it relaxes traffic congestion around the hot spot. However, the above path selection algorithms may select a path to congestion points even if there exist alternative paths which can avoid them. To alleviate this problem, a traffic balancing algorithm using a static analysis of paths is proposed by Sancho[14] et al. for Up*/Down* routing in Myrinet[1], which uses no virtual channels, as follows.

1. All legal paths between every pair of switches are calculated. Then, this algorithm associates a counter to every channel [3], and each counter is initialized to the number of the legal paths crossing its channel.
2. A path crossing the channel with the largest value of counter is selected to be removed if there is more than one path between its source and destination switches. If there is more than one path which can be removed in the channel, the path whose source and destination switches have the largest number of paths between them is selected.
3. When the path is removed, the counters associated with channels are updated.
4. Repeat Step 2 until the number of legal paths between every pair of switches is reduced down to the unit.

The time complexity to compute the Sancho's algorithm is $O(n^2 * diameter)$,

[2] Notice that path selection algorithms can not apply to a special routing algorithm called Silla's minimal routing[15], because it guarantees deadlock-free through selecting a path between original channel (deadlock-free escape path) and new channel(fully adaptive path) dynamically.

[3] Since each link consists of two uni-directional physical channels in Myrinet, two counters per link are needed.

where n is the number of switches[14].

3 Four Path Selection Algorithms using the Static Analysis of Paths

Sancho's algorithm[14] is an efficient method to uniformly distribute paths in SANs without virtual channels. However, virtual channels which can use a physical channel in time-sharing manner are plentifully equipped in advanced switches[2] [3] [4]. For such networks, there are other concepts worth to try.

In this section, we present four path selection algorithms for SANs with virtual channels: "high virtual-channel first", "high physical-channel first", "low virtual-channel first" and "low physical-channel first". Since they try to distribute paths uniformly using the static analysis, they have the same flow as Sancho's algorithm except that Step 2 is different as follows. Here, each counter is corresponding to a virtual channel unit in the flow.

- *High virtual-channel first* selects the virtual channel with the largest value of counter. Then, it removes a path on it if there is more than one path between its source and destination switches.

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

  If there is more than one path which can be removed in the virtual channel, the path whose source and destination switches have the largest number of paths between them is selected to be removed.
- *Low virtual-channel first* selects the virtual channel with the smallest value of counter, and it fixes one path on it. That is, the other paths between its source and destination switches are removed.

  If there is more than one path which are still not fixed in the virtual channel, the path whose source and destination switches have the smallest number of paths between them is selected to be fixed.
- *Low physical-channel first* selects the virtual channel with the smallest value of counter on the physical channel whose sum of counters associated to every virtual channel is the smallest. Then, a path on it is fixed. If there is more than one path which are still not fixed in the virtual channel, the path whose source and destination switches have the smallest number of paths between them is selected to be fixed.

High virtual-channel first is a quite simple extension of Sancho's algorithm for SANs with virtual channels. *High physical-channel first* is designed so as to avoid both the physical and virtual channels bottlenecks, while *high virtual-channel first* tries to avoid the virtual channel bottleneck. On the other hand, *low virtual-channel first* tries to uniformly use all virtual channels by keeping paths crossing virtual channels that are not crowded. Thus, it tries to avoid virtual channels with extremely small utilization. Moreover, *low physical-channel first* tries to uniformly use not only virtual channels but also physical channels by keeping paths crossing virtual and physical channels that are not crowded.

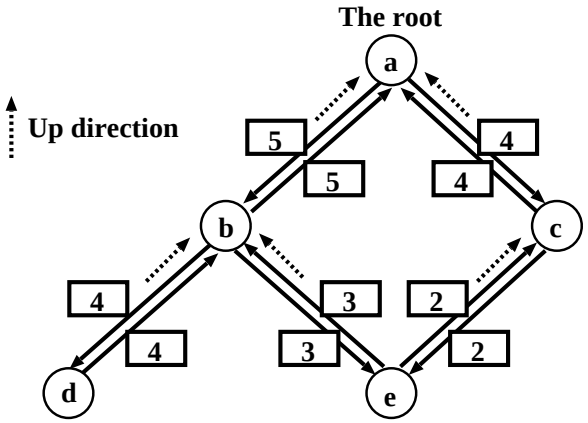


Fig. 1. The first example of counters to Up*/Down* paths

Figure 1 shows an example of a five-switches SAN with no virtual channels, and Up*/Down* routing is applied on it(see Section 4)[9]. In Figure 1, the value of counter to each channel is calculated according to the number of paths crossing it. For example, the value of counter on the channel from switch a to switch c is four because path $(a, e), (a, c), (b, c)$, and (d, c) are crossing it, where (x, y) represents the path from switch x to switch y. When designing a deterministic routing, path (a, e) and (e, a) have alternative routes$(a \rightarrow b \rightarrow e), (a \rightarrow c \rightarrow e)$ and $(e \rightarrow b \rightarrow a), (e \rightarrow c \rightarrow a)$ respectively. Then, simple algorithms, *random* and *low port first* may select the former path 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 paths select the latter better path.

Figure 2 shows the next example of the counter on Up*/Down* routing. Then, path (a, d) and (d, a) have alternative routes$(a \rightarrow b \rightarrow d), (a \rightarrow c \rightarrow d)$ and $(d \rightarrow b \rightarrow a), (d \rightarrow c \rightarrow a)$ respectively. In this case, *high virtual-channel first* and *high physical-channel first* select the former path, while *low physical-channel first* and *low virtual-channel first* select the latter path. In this case, we cannot easily judge which of the paths is suitable. Thus, through a simulator, we will show and discuss which policy is better in Section 4.

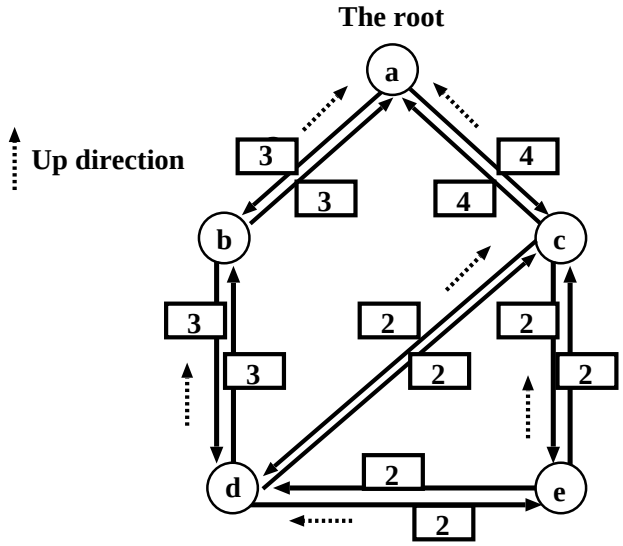


Fig. 2. The second example of counters to Up*/Down* paths

4 Performance Evaluation

In this section, performance of path selection algorithms under Up*/Down* routing[9], and the DL routing[13] is evaluated using a flit-level simulation.

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 provides eight bidirectional ports; four ports to connect with hosts and remaining ports for connecting other switches in all topologies. Here, a simple model consisting of channel buffers, crossbar, link controller and control circuits is used for the switching fabric. Two classes of network topologies, irregular and regular, are used as follows. As irregular topologies, ten different ones are randomly generated under the condition that only one link is connected between two different switches. That is, all of ten irregular topologies use $2n$ links to connect two switches and n links to connect host and switch, where n is the number of hosts. Two network sizes, small (16 switches) and large (64 switches), are used in irregular topologies. Evaluation trend under ten topologies is similar to the average one under the larger number of topologies. On the other hand, 64 switches two-dimensional torus is used as a regular topology. A destination of a packet is determined by the traffic patterns, *uniform* or *bit-reversal*, and hosts inject a packet independently of each other. In bit-reversal traffic, a host with the identifier $(a_0, a_1, \cdots, a_{n-1})$ sends a packet to the host whose identifier is the bit reversal $(a_{n-1}, \cdots, a_1, a_0)$ of the source host, while in uniform traffic a host generates a packet to the host,

that is randomly selected. Thus, every host will be both sender and receiver. A header flit transfer requires at least three clocks, that is, one for routing, one for transferring a flit from an input channel to output channel through a crossbar, and the rest for transferring the flit to the next switch or host. The model is simple compared with real switches, such as, the operation in RHiNET-2/SW[2]. Nevertheless, it is useful for larger systems and complicated topologies because more exact modeling of modern switching fabrics consumes a huge simulation time.

Other parameters are set as shown in Table 1.

Table 1
Simulation parameters

Simulation time	1,000,000 clocks (ignore the first 50,000 clocks)
The number of virtual channels	1 (Up*/Down*) or 5(DL)
Packet length	128 flits
Switching tech.	virtual cut-through

4.2 Routing Algorithms

4.2.1 Up/Down* Routing*

Up*/Down* routing used in Autonet[9] is a typical deadlock-free adaptive routing for SANs. Up*/Down* routing is based on an assignment of direction to network channels[9]. As a basis of the assignment, a spanning tree whose node (also called vertex) corresponds to a switch in the network is built. A traditional algorithm to build it is the breadth-first search (BFS), in which the switch with identifier *zero* is selected as the root. After building the BFS spanning tree, the "up" end of each channel is defined as follows: (1) the end whose node is closer to the root in the spanning tree; (2) the end whose node has the lower unique identifier (UID), if both ends are at the nodes at the same tree level. The restriction on Up*/Down* routing is simple: a legal path must traverse zero or more channels in the up direction followed by zero or more channels in the down direction. Thus, the Up*/Down* rule prohibits any packet transfer from the down direction to the up direction. Notice that Up*/Down* routing requires no virtual channels.

4.2.2 The DL Routing

Descending layers (DL) routing proposed by us is an improved routing algorithm of Up*/Down* routing using virtual channels[13]. It divides the network

into sub-networks with the same topology consisting of layers of virtual channels, and it establishes a large number of paths across sub-networks in order to reduce the path length and path congestion.

In the implementation, the DL routing uses Up*/Down* routing as long as a packet is routed inside a sub-network. Thus, a packet must not be transferred from a down channel to an up channel without switching sub-networks.

In this simulation, we use five virtual channels, which are enough to take deadlock-free minimal paths in both of these 16, and 64-switches SANs.

4.3 Simulation Results

4.3.1 Irregular Topologies with 16 switches

Table 2
Throughput of path selection algorithms under Up*/Down* routing on 16-switches irregular topologies [flits/host/clock]

	Uniform traffic		Bit reversal traffic	
	Avg	SD	Avg	SD
Low port first	0.117	0.016	0.175	0.028
Random	0.116	0.014	0.173	0.032
Sancho's algorithm	0.131	0.019	0.187	0.027
Low vch first	0.119	0.015	0.183	0.034

Table 3
Throughput of path selection algorithms under the DL routing on 16-switches irregular topologies [flits/host/clock]

	Uniform traffic		Bit reversal traffic	
	Avg	SD	Avg	SD
Low port first	0.169	0.008	0.277	0.052
Random	0.262	0.026	0.328	0.043
High vch first	0.324	0.023	0.365	0.049
High pch first	0.322	0.026	0.363	0.045
Low vch first	0.283	0.024	0.367	0.047
Low pch first	0.285	0.027	0.362	0.046

Table 2 and 3 show the average throughput of 10 irregular topologies with 16 switches, and its standard deviation (SD). In all figures and tables on this section, *high virtual-channel first*, *high physical-channel first*, *low virtual-channel*

first, and *low physical-channel first* are shown as "high vch first", "high pch first", "low vch first", and "low pch first" respectively. Under Up*/Down* routing in SANs without virtual channels, Sancho's algorithm is equal to *high physical-channel first* and *high virtual-channel first* because we can consider that there is a virtual channel (a single virtual channel corresponds a physical channel) in the SANs. *Low physical-channel first* is also equal to *low virtual-channel first* in such SANs. Here, throughput is defined as the maximum amount of accepted traffic. Accepted traffic is the flit reception rate in a host in each clock cycle. Throughput is the most important metric of routing algorithm in SANs. As shown in Table 2 and 3, there are no path selection algorithms that have high stability of throughput on irregular topologies, because the SD of throughput is influenced by each condition. However, the four algorithms using the static analysis of paths improve up to 92% of throughput compared with simple ones on the average throughput.

Table 2 and 3 also show that each routing algorithm in bit reversal traffic achieves higher throughput than one in uniform traffic. This comes from that, in uniform traffic, packets whose source hosts are different have possibility to collide at a consumption channel on the destination host. Such collisions drastically degrade the performance especially at smaller network sizes. On the other hand, on bit reversal traffic, such collisions at consumption channel are not occurred.

4.3.2 Irregular Topologies with 64 Switches

Table 4
Throughput of path selection algorithms under Up*/Down* routing on 64-switches irregular topologies [flits/host/clock]

	Uniform traffic		Bit reversal traffic	
	Avg	SD	Avg	SD
Low port first	0.034	0.004	0.038	0.006
Random	0.034	0.004	0.039	0.006
Sancho's algorithm	0.037	0.006	0.041	0.007
Low vch first	0.035	0.004	0.039	0.006

Table 4 and 5 show average throughput of 10 irregular topologies with 64 switches. The condition of simulation is the same in the before section except the number of switches.

Table 5 demonstrates that path selection algorithms affect performance, and *high virtual-channel first* and *high physical-channel first* achieve high performance compared with *low virtual-channel first* and *low physical-channel first*. Thus, in order to uniformly distribute the traffic, it can be said the methods

Table 5
Throughput of path selection algorithms under the DL routing on 64-switches irregular topologies [flits/host/clock]

	Uniform traffic		Bit reversal traffic	
	Avg	SD	Avg	SD
Low port first	0.105	0.002	0.123	0.012
Random	0.155	0.005	0.199	0.012
High vch first	0.189	0.008	0.210	0.009
High pch first	0.191	0.008	0.210	0.011
Low vch first	0.167	0.012	0.208	0.008
Low pch first	0.162	0.012	0.208	0.008

Table 6
Channel crossing paths on 64-switches irregular topologies

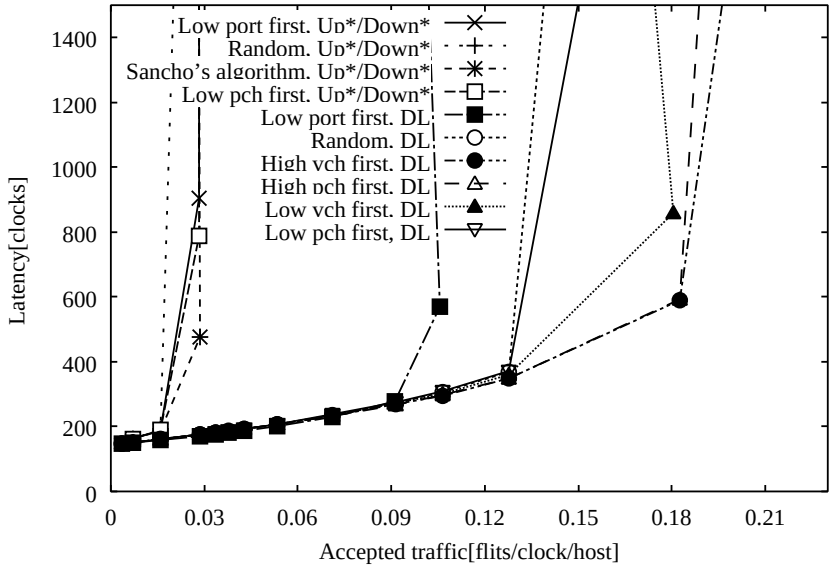
	Up*/Down* routing		DL routing	
	Avg	SD	Avg	SD
Low port first	61.86	59.94	50.05	14.54
Random	61.86	59.94	50.05	13.35
Sancho's algorithm	61.86	56.97	—	—
High vch first	—	—	50.05	10.00
High pch first	—	—	50.05	9.95
Low vch first	61.86	59.32	50.05	12.17
Low pch first	—	—	50.05	12.75

to remove the bottleneck channels are more efficient than the methods to keep paths crossing channels that are not crowded. As shown in Table 4 and 5, the impact of path selection algorithms on the DL routing is greatly enhanced compared with that on Up*/Down* routing. Thus, it can be said that when a routing algorithm, which provides the larger number of alternative paths, is used, path selection algorithms are more crucial to the throughput.

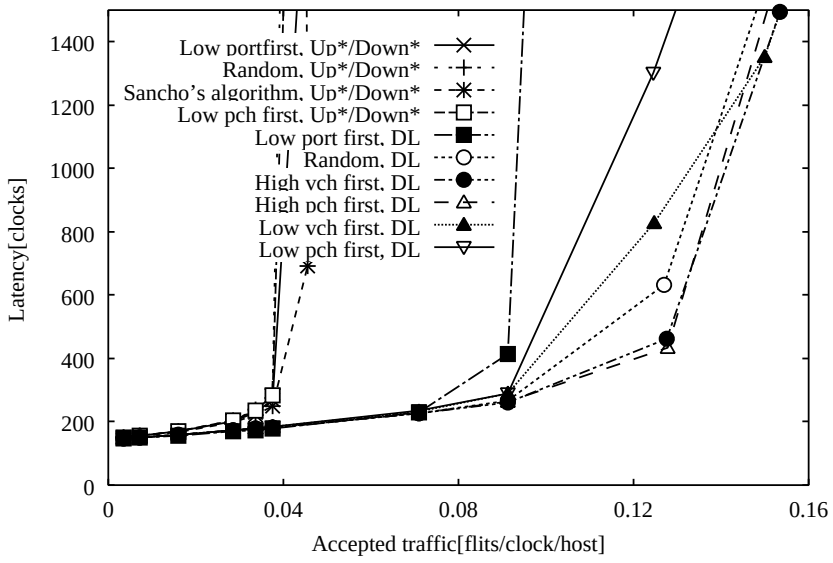
To investigate the distribution of paths, we introduce a metric called channel crossing paths. Channel crossing paths are defined as the number of paths crossing through a physical channel after selecting only a single path between each pair of switches. Table 6 shows the SD/average of channel crossing paths. The SD shows how uniformly the paths are distributed, and the small SD of channel crossing paths means that the paths are distributed uniformly.

Table 6 demonstrates that the four path selection algorithms using the static

analysis distributes the paths more uniformly than simple ones under the DL routing. As shown in Table 4, 5, and 6, the distribution of paths directly influences the throughput of path selection algorithms in most cases. The approach of the four proposed path selection algorithms is based on the analysis of channel crossing paths, thus decreasing the SD. This is the reason why the four proposed path selection algorithms increase throughput. Notice that the DL routing has smaller values of the average channel crossing paths than those of Up*/Down* routing, because the DL routing takes the smaller path hops. Since the average channel crossing paths only depend on the average path hops, path selection algorithms don't affect them.



(a) Uniform traffic



(b) Bit reversal traffic

Fig. 3. Accepted traffic versus Latency on typical 64-switches irregular topologies

Figure 3 shows the latency on irregular topologies taken from the ten irregular

topologies. Notice that this latency tendency of each path selection algorithm is similar to that under the other irregular topologies. Latency is the second important metric of routing algorithm in SANs. Figure 3 shows that Sancho's algorithm, *high virtual-channel first*, and *high physical-channel first* drastically decrease the latency (the number of packet collisions) because of their well-distributed paths. Table 2, 3, 4, and 5 show that the best path selection algorithm depends on the network size, routing algorithm, and traffic pattern, however *high virtual-channel first* and *high physical-channel first* always outperform the other path selection algorithms.

The impact of path selection algorithms on throughput under 16 switches is similar to that under 64 switches. Thus, when increasing the number of hosts, we consider that the impact of path selection algorithms is still large, and it can't be ignored to design SANs.

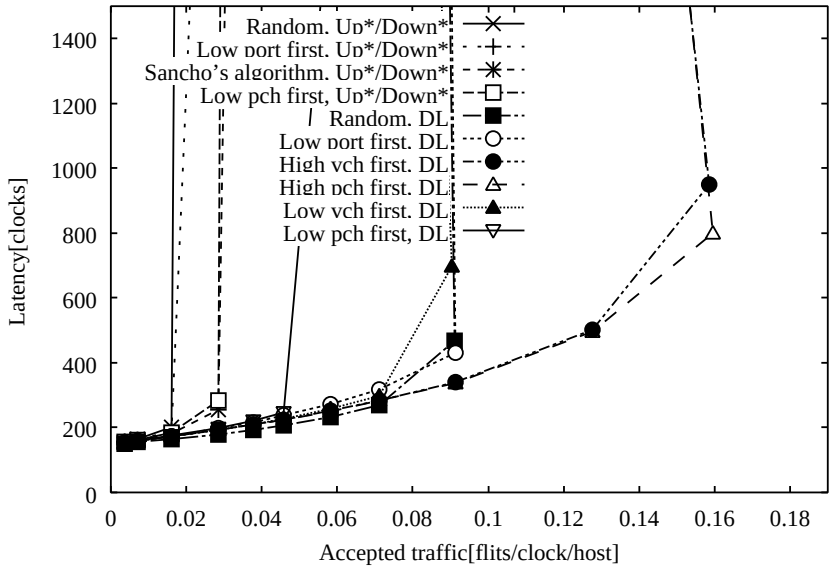
4.3.3 Two-Dimensional Torus

Figure 4 shows the relation between the average latency and the accepted traffic of path selection algorithms on 8×8 two-dimensional torus. Each x and y axis represents accepted traffic and latency respectively. The condition of simulation is the same in the above section except the switch topology. Table 7 also shows the routing metric in this case.

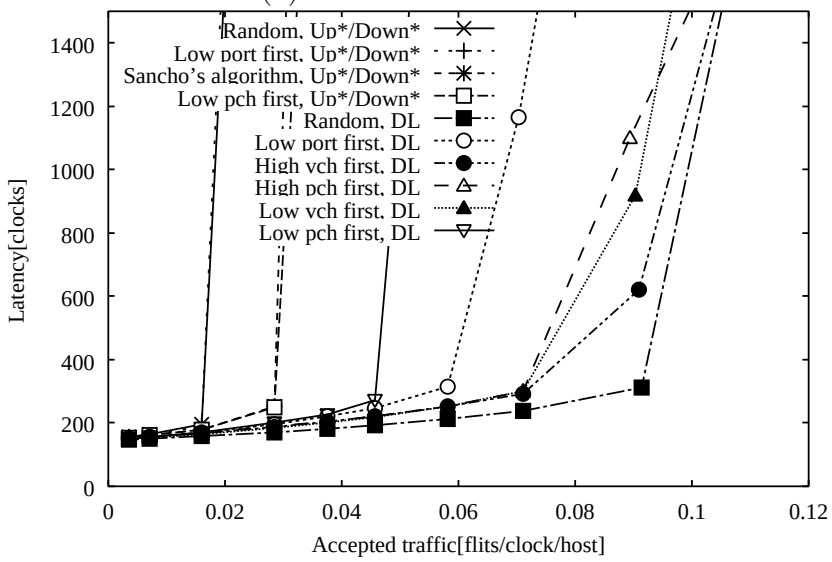
Table 7
Channel crossing paths on 8×8 2D torus

	Up*/Down* routing		DL routing	
	Avg	SD	Avg	SD
Low port first	72.00	71.75	64.00	16.00
Random	72.00	69.05	64.00	27.93
Sancho's algorithm	72.00	54.10	—	—
High vch first	—	—	64.00	6.36
High pch first	—	—	64.00	5.38
Low pch first	72.00	61.02	64.00	20.69
Low pch first	—	—	64.00	31.09

The evaluation on regular topologies is also important, since topologies of most SANs are not completely irregular but have some regularity in practice. As shown in Table 4, 5 and Figure 4, throughput under irregular topologies outperform one under torus in each path selection algorithm. This is because Up*/Down* routing and the DL routing are designed for irregular topologies by introducing spanning tree. Whereas both of them are sometimes difficult to make an efficient spanning tree under regular topologies[13].



(a) Uniform traffic



(b) Bit reversal traffic

Fig. 4. Accepted traffic versus Latency on 8 ×8 torus

Figure 4 and Table 7 show that the path selection algorithms using the static analysis of paths achieve higher throughput compared with low port first at two-dimensional torus as well as at most cases of irregular topologies. In particular, Sancho's algorithm, *high virtual-channel first* and *high physical-channel first* provide better performance than *low physical-channel first* and *low virtual-channel first*.

5 Related Work

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

5.1 Output Selection Function

In an adaptive routing, an 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 single output channel. The output selection function is essentially required when an adaptive routing is implemented. On the other hand, a path selection algorithm is required to implement a deterministic routing from alternative paths. Sophisticated output selection functions usually use a measure which indicates the congestion of each output channel, and they dynamically decide the output only with the local congestion data inside the switch[16], unlike path selection algorithms.

5.2 Source Routing using Dynamic Selection of Alternative Paths

There are basically two implementation of a deterministic routing: the distributed routing and the source routing. In the source routing used in Myrinet[1] and QsNET[4], 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. On the other hand, the only source can select a path from alternative paths dynamically. Simple examples of such selection policies are random selection and round robin[17]. However, using such policies, in-order packet delivery is not guaranteed unlike the path selection algorithm.

6 Conclusions

A path selection algorithm, which statically selects a single path from alternative paths, is usually required for advanced routings in SANs. In this paper, (1) we develop four path selection algorithms using the static analysis of paths in order to distribute the traffic uniformly, and (2) we investigate the influence of various path selection algorithms on throughput and latency. Result of simulations shows that high physical-channel first improves up to 92% of

throughput against simple path selection algorithms, and policies to remove paths crossing the bottleneck channels are more efficient than ones to keep paths crossing channels that are not crowded. We are planning to implement and evaluate path selection algorithms on a real PC cluster consisting of 64 hosts and 16 switches called RHiNET-2[18].

References

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

[2] S.Nishimura, T.Kudoh, H.Nishi, J.Yamamoto, K.Harasawa, N.Matsudaira, S.Akutsu, K.Tasho, H.Amano, High-speed network switch RHiNET-2/SW and its implementation with optical interconnections, in: Hot Interconect, 2000, pp. 31–38.

[3] I.T.Association, Infiniband architecture. specification volumen 1,release 1.0.a, available from the InfiniBand Trade Association, http://www.infinibandta.com.

[4] F. Petrini, W. Feng, A. Hoisie, The Quadrics network (QsNet): high-performance clustering technology, in: Proceedings of Hot Interconnects, 2001, pp. 125–130.

[5] W. J. Dally, C. L. Seitz, Deadlock-Free Message Routing in Multiprocessor Interconnection Networks, IEEE Trans. Comput. 36 (5) (1987) 547–553.

[6] P. Kermani, L. Kleinrock, Virtual cut-through: A new computer communication switching techniques, Computer Networks 3 (4) (1979) 267–286.

[7] S. L. Scott, G. T.Horson, The Cray T3E network: adaptive routing in a high performance 3D torus, in: Proceedings of Hot Interconnects IV, 1996, pp. 147–156.

[8] J.Carbonaro, F.Verhoorn, Cavallino: The teraflops router and NIC, in: Proceedings of Hot Interconnects Symposium IV, 1996, pp. 157–150.

[9] M.D.Schroeder, al et., Autonet: a high-speed, self-configuring local area network using point-to-point links, IEEE Journal on Selected Areas in Communications 9 (1991) 1318–1335.

[10] A.Jouraku, M.Koibuchi, A.Jouraku, H.Amano, Routing Algorithms Based on 2D Turn Model for Irregular Networks, in: Proceedings of the International Symposium on Parallel Architectures, Algorithms, and Networks, 2002, pp. 289–294.

[11] M.P.Merlin, J.P.Schweitzer, Deadlock Avoidance in Store-and-Forward Networks, IEEE Trans. Comput. COM-28 (3) (1980) 345–354.

[12] T. Skeie, O. Lysne, I. Theiss, Layered Shortest Path (LASH) Routing in Irregular System Area Networks, in: Proceedings of International Parallel and Distributed Processing Symposium, 2002, pp. 162–169.

[13] M.Koibuchi, A.Jouraku, H.Amano, Descending Layers Routing: A Deadlock-Free Deterministic Routing using Virtual Channels in System Area Networks with Irregular Topologies, in: Proceedings of the International Conference on Parallel Processing, 2003, pp. 527–536.

[14] J.C.Sancho, A.Robles, Improving the Up*/Down* Routing Scheme for Networks of Workstations, in: Proceedings of the European Conference on Parallel Computing, 2000, pp. 882–889.

[15] F. Silla, J. Duato, High-Performance Routing in Networks of Workstations with Irregular Topology, IEEE Transactions on Parallel and Distributed Systems 11 (7) (2000) 699–719.

[16] J.C.Martinez, F.Silla, P.Lopez, J.Duato, On the Influence of the Selection Function on the Performance of Networks of Workstations, in: Proceedings of the International Symposium on High Performance Computing, 2000, pp. 292–300.

[17] J.Flich, M.P.Malumbers, P.Lopez, J.Duato, Improving Routing Performance in Myrinet Networks, in: Proceedings of 14th International Parallel and Distributed Processing Symposium(IPDPS), 2000, pp. 27–32.

[18] M. Koibuchi and K. Watanabe and K. Kono and A. Jouraku and A. Amano, Performance Evaluation of Routing Algorithms in RHiNET-2 Cluster, in: Proceedings of IEEE International Conference on Cluster Computing, 2003, pp. 395–402.