Evaluation of Routing Algorithms for InfiniBand Networks*

M.E. Gómez, J. Flich, A. Robles, P. López, and J. Duato

Department of Computer Science, Universidad Politécnica de Valencia,
P.O.B. 22012, 46071 – Valencia, Spain,
{megomez,jflich,arobles,plopez,jduato}@gap.upv.es

Abstract. Storage Area Networks (SANs) provide the scalability required by the IT servers. InfiniBand (IBA) interconnect is very likely to become the de facto standard for SANs as well as for NOWs. The routing algorithm is a key design issue in irregular networks. Moreover, as several virtual lanes can be used and different network issues can be considered, the performance of the routing algorithms may be affected. In this paper we evaluate three existing routing algorithms (up*/down*, DFS, and smart-routing) suitable for being applied to IBA. Evaluation has been performed by simulation under different synthetic traffic patterns and I/O traces. Simulation results show that the smart-routing algorithm achieves the highest performance.

1 Introduction

In IBA [2] networks, switches can be arranged freely in order to provide wiring flexibility and incremental expansion capability. The irregularity in the topology makes the routing quite complicated. Several routing algorithms for irregular topologies have been proposed. The up*/down*, smart-routing and DFS routings[1] are suitable for IBA networks due to the fact that they can be implemented in a deterministic way. The three routing algorithms have been already evaluated in [1] and for wormhole-switched Myrinet networks by using different synthetic traffic patterns that might not be representative of SANs. However, in this paper, we also use I/O traces for IBA interconnects that use virtual cut-through switching.

In a SAN environment, the use of a particular routing algorithm together with the distribution of storage devices may significantly affect the overall system performance. Moreover, IBA allows the use of several virtual lanes (VL). In a SAN environment, it could be thought that some disks can be addressed through a particular VL. In this paper, we will also evaluate how the disk distribution affects the performance of the routing algorithms and the use of different VLs.

* This work was supported by the Spanish CICYT under Grant TIC2000-1151-C07 and by Generalitat Valenciana under Grant GV00-131-14.

[1] These routing algorithms can be implemented on IBA networks by the strategies proposed in [3,4].

B. Monien and R. Feldmann (Eds.): Euro-Par 2002, LNCS 2400, pp. 775–780.

The paper is organized as follows. In the next section, the main simulator considerations are described. In section 3 the simulation results are discussed. Finally, some conclusions are drawn in Section 4.

2 Simulation Model

We have developed a detailed simulator that allows us to model the network at the register transfer level following the IBA specifications [2]. We will use with a non-multiplexed crossbar on each switch with a simple crossbar arbiter based on FIFO request queues per output crossbar port. The routing time at each switch will be set to 100 ns. This time includes the time to access the routing tables, the crossbar arbiter time, and the time to set up the crossbar connections. The link injection rate will be fixed to the 1X configuration [2].

We have used different message destination distributions. In the uniform distribution, the destination of a message is chosen randomly. In the hot-spot distribution, a percentage of traffic is sent to one host. In the distribution with several hot-spot hosts, 10% of traffic is sent to them. When using synthetic traffic, we will use short packets with a payload of 32 bytes, and long packets with a payload of 256 bytes. Buffer size (input and output) will be fixed to 1 KB. We will analyze irregular networks of 8, 16, 32 and 64 switches randomly generated. We will assume that every switch in the network has 8 ports, using 4 ports to connect to other switches and leaving 4 ports to connect to hosts (servers and storage devices).

The I/O traces were provided by Hewlett-Packard Labs. They include all the I/O activity generated from 1/14/1999 to 2/28/1999 at the disk interface of the *cello* system. A detailed description of similar traces of 1992, collected in the same system, can we found in [5]. We will use packets with a payload equal to the size specified in the trace for the I/O accesses, but if the access is larger than 512 bytes, we will split it into packets with a payload of 512 bytes at most. Buffer size (input and output) will be fixed to 8KB. The disks will be attached to twenty-three ports. The rest of switch ports will be connected to hosts. When using I/O traces, three different evaluations[2] will be performed. Firstly, we will use only one virtual lane (VL). The disks will be randomly distributed over the network. Secondly, we will use different disk distributions. In particular we will distribute the disks: (1) randomly; (2) concentrated (disks are grouped in 6 switches selected randomly); and (3) uniformly (only one disk will be attached to a particular switch). And finally, we use different VLs. When using different VLs we need a different SL for each VL. We will refer to this assignment as SL/VL. All the traffic injected into a particular VL remains in the the same VL until delivered.

[2] Regarding performance of routing algorithms, latency is the elapsed time between the generation of a packet at the source host until it is delivered at the destination node, accepted traffic is the amount of information delivered by the network per time unit.

3 Performance Evaluation

3.1 Results for Synthetic Traffic

Figures 1.a and 1.b show the behavior of the three routing algorithms for uniform distribution of packet for different network sizes. SMART routing is not shown for the 64-switch network due to its high computation time. As it was expected, SMART achieves the highest network throughput for all the evaluated topologies. In particular, for 32 switches SMART increases network throughput by factors of 2.29 and 1.11 with respect to UD and DFS, respectively. The higher network throughput achieved by SMART and DFS routings is due to their better traffic balance, as can be seen in Figure 1.c.

Table 1 shows minimum, maximum, and average increases of network throughput when comparing UD, DFS, and SMART using 10 random topologies for each network size. We observe that SMART always increases network throughput with respect to UD and DFS. We also observe that, as network grows, DFS also increases its improvement over UD. For 64-switch networks DFS improves over UD by a factor of 2.66, on average. For the hot-spot traffic pattern, on average, DFS and SMART decrease their throughput (with respect to UD) when changing

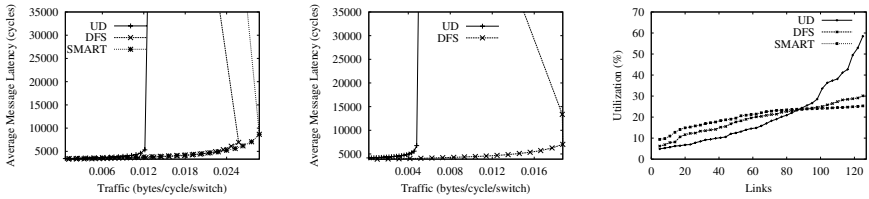

Fig. 1. (a) and (b): Average packet latency vs. traffic. Destination distribution is uniform. Network size is (a) 32, and (b) 64 switches. Packet length is 32 bytes. (c): Link utilization. Traffic is 0.021 flits/cycle/switch (32 switches). Packet size is 256 bytes. Uniform distribution.

Table 1. Factor of throughput increase between UD, DFS, and SMART for different traffic patterns. Packet size is 32 bytes.

				SMART vs UD			SMART vs DFS			DFS vs UD		
Sw.	Distr	Number HS	Percentage	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
16	Unif.	-	-	1.3	2.12	1.72	1.00	1.32	1.09	1.13	1.95	1.52
32	Unif.	-	-	1.67	3.29	2.46	1.07	1.75	1.23	1.23	2.63	2.03
64	Unif.	-	-	N/A	N/A	N/A	N/A	N/A	N/A	2.11	3.73	2.66
32	HS	1	5%	1.24	1.92	1.51	0.98	1.33	1.09	1.23	1.98	1.44
32	HS	1	10%	0.94	1.26	1.11	0.97	1.05	1.00	0.97	1.22	1.11
32	HS	1	20%	0.97	1.08	1.03	0.97	1.01	1.00	0.99	1.10	1.04
32	HS	2	10%	1.23	2.38	1.68	0.94	1.21	1.04	1.08	2.24	1.60
32	HS	4	10%	1.53	2.73	2.07	0.87	1.50	1.13	1.08	2.5	1.86
32	HS	8	10%	1.86	3.14	2.35	1.04	1.70	1.23	1.09	2.67	1.94

from a 5% hot-spot (1.51 and 1.44) to a 20% hot-spot (1.03 and 1.04). Finally, in Table 1 we show the same results for several hot-spot hosts. As the number of hot-spots increases in the network, the traffic is better balanced and therefore we can take advantage of a better designed routing algorithm (SMART and DFS).

3.2 Results with I/O Traces

First, we present results obtained with I/O traces and using only one SL/VL. Figure 2.a and Figure 2.b show the cumulative latency [3] versus simulated time using the three routing algorithms. The used traces are three years old, so it seems reasonable that nowadays I/O traffic has changed. In particular, the technology is quickly growing each year, allowing faster devices (hosts and storage devices) to be used, and thus generating higher injection rates. For this, we have applied different time compression factors to the traces. In Figures 2.a and 2.b we can see the performance of the routing algorithms with compression factors of 15 and 20, respectively. We can observe that, in these situations, the UD routing exhibits a very high latency. On the other hand, when using DFS and SMART, the behavior is much better. In Figure 2.c we can see the average number of packets enqueued per host. The UD routing is not able to manage all the injected packets.

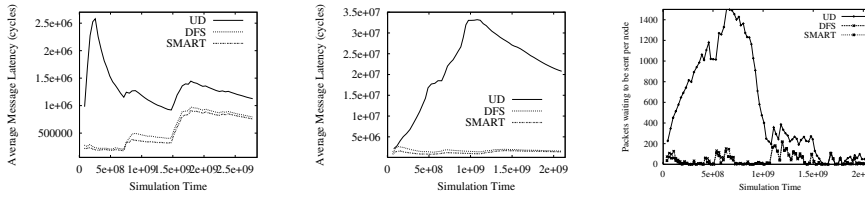

Fig. 2. (a) and (b): Cumulative average message latency vs simulation time (32 switches). Random disk distribution. Compression factor is (a) 15 and (b) 20. (c): Mean number of packets waiting to be sent per node vs simulation time. Compression factor 20.

Now, we analyze how the disk distribution over the network affects the different routing schemes. Figure 3 compares the three disk distributions for every scheme. UD is the most sensitive routing to disk distributions. For example, in Figure 3.a we can observe that, at the beginning of the simulation, concentrating the disks in some switches is better than randomly distributing them, whereas later the random distribution of disks has a much better behavior. The other routings (DFS and SMART) are much more robust to the disk distribution.

[3] The cumulative latency is obtained by adding the latency of all the messages (from the beginning of the simulation) and dividing by the number of messages received at each simulation time.

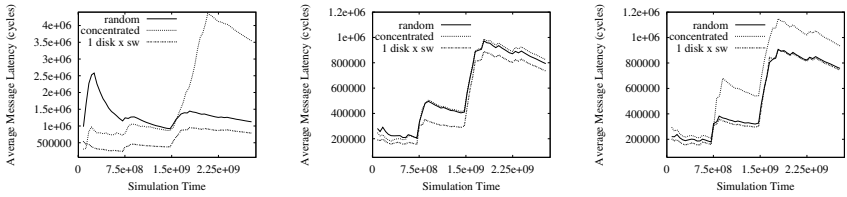

Fig. 3. Cumulative average message latency from generation vs simulation time using different disk distributions. 1 SL/VL (32 switches) for different disk distribution. Compression factor is 15. Routing scheme is (a) UD, (b) DFS, and (c) SMART.

For all the schemes, the best option is distributing the disks among switches, having one disk per switch. By doing this, the workload is better balanced in the network. Moreover, SMART obtains similar results for randomly distributed disks and one disk per switch distribution. Hence, SMART is the least sensitive routing algorithm to changes in the disk distribution.

Finally, we analyze how the use of SL/VLs affects the performance of the routing schemes. Figure 4 shows the behavior of the three routing schemes using different numbers of SL/VL. Disks are assigned randomly to SL/VLs. bAs we can see, the UD routing (Figure 4.a) benefits from using an additional SL/VL (2 SL/VL). Latency is noticeably reduced. The peak latency is reduced from 2.5 million of cycles to 1 million of cycles. Using two additional SL/VLs (4 SL/VL) helps even more. When using 8 SL/VL additional improvements are not achieved. The other routing algorithms (DFS and SMART) obtain low improvements on performance when using additional SL/VLs. The congestions caused by the UD routing is reduced when using different SL/VLs. However, the DFS and SMART routings can handle traffic in an efficient way. As a conclusion, we can see that even by using a large number of network resources (8 SL/VLs) the UD is not able to obtain the good network performance achieved by the other routings (DFS and SMART) with only one SL/VL.

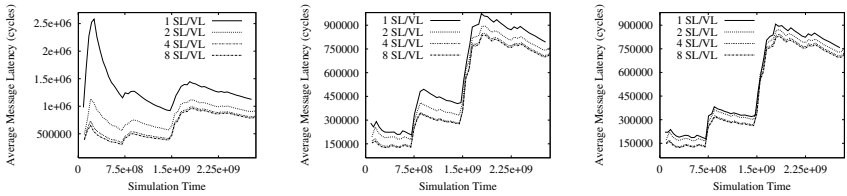

Fig. 4. Cumulative average message latency from generation vs. time using different numbers of SL/VL (32 switches). Compression factor is 15. Routing scheme is (a) UD, (b) DFS, and (c) SMART.

4 Conclusions

In this paper, we have evaluated by simulation three routing schemes (SMART, UD, and DFS) suitable for being applied to IBA networks. SMART is the routing strategy that achieves the best behavior under any network workload (synthetic traffic and I/O traces). However, its performance are very close to that of DFS. This behavior is mainly due to the better traffic balance exhibited by SMART and DFS. When analyzing the behavior under I/O traces, it is observed that UD has not enough capacity to manage the traffic generated by the trace. This causes an increase in the number of packets stored in queues and, in turn, a significant increase in the packet latency. However, SMART and DFS have no problem to follow near the injected traffic. Moreover, these routing algorithms exhibit a greater robustness than UD, facing eventual changes in the disk distribution. On the other hand, unlike SMART and DFS, UD takes advantage of using additional SLs/VLs in order to reduce the head-of-line blocking effect in the input buffers. Despite it, with only one SL/VL, SMART and DFS continue to outperform UD, even when the latter strategy uses 8 SLs/VLs.

References

1. J. Flich, P. Lopez, M.P. Malumbres, J. Duato, and T. Rokicki, "Combining In-Transit Buffers with Optimized Routing Schemes to Boost the Performance of Networks with Source Routing," *Proc. of Int. Symp. on High Performance Computing*, Oct. 2000.
2. InfiniBand^{TM} Trade Association, *InfiniBand^{TM} architecture. Specification Volumen 1. Release 1.0.a.* Available at http://www.infinibandta.com.
3. J.C. Sancho, A. Robles, and J. Duato, Effective Strategy to Compute Forwarding Tables for InfiniBand Networks, in *Proc. of 2001 International Conference on Parallel Processing (ICPP'01)*, Sept. 2001.
4. P. López, J. Flich, and J. Duato, Deadlock-free Routing in InfiniBand^{TM} through Destination Renaming, in *Proc. of 2001 International Conference on Parallel Processing (ICPP'01)*, Sept. 2001.
5. C. Ruemmler, J. Wilkes, *Unix Disk Access Patterns*, Winter Usenix Conference, Jan 1993.