Cost / Performance Trade-Offs and Fairness Evaluation of Queue Mapping Policies*

Teresa Nachiondo[1], José Flich[1], José Duato[1], and Mitchell Gusat[2]

[1] Dept. of Computer Engineering, Universidad Politécnica de Valencia, 46071–Valencia, Spain
{tnachion,jflich,jduato}@gap.upv.es
[2] IBM, Research, Zurich Research Lab. GmbH, Saeumerstr. 4,
CH-8803, Rueschlikon, Switzerland
mig@zurich.ibm.com

Abstract. Whereas the established interconnection networks (ICTN) achieve low latency by operating in the linear region, i.e. oversizing the fabric, the recent strict cost and power constrains demand more efficient utilization of future networks. Increasing the utilization of lossless ICTNs may, however, lead to saturation and performance degradation owing to HOL-blocking. The current solution to HOL-blocking consists of using Virtual Output Queueing (VOQ), whose quadratical scalability is expensive in large networks. To improve VOQ's scalability we have proposed the Destination-Based Buffer Management (DBBM), a scheme that compares well with VOQ. Whereas previously we have analyzed DBBM's basic operation and performance, in this paper we have set two different goals. First we focus on how the different DBBM mappings can impact the cost/performance of multistage ICTNs. Next, because DBBM can introduce unfairness, this constitutes the second theme of our paper. The new results show that DBBM with modulo-4/8 mapping performs very well for only a fraction of the VOQ cost. Also in terms of fairness DBBM shows promise, because it (i) keeps the unfairness degree independent of both topology and routing, while (ii) minimizing the number of flows affected by unfairness.

1 Introduction

Proprietary lossless ICTNs are frequently used to build large supercomputers such as BlueGene/L [7] and the Earth Simulator [6]. Alternatively, commercial ICTNs like InfiniBand [8], Myrinet [11], and Quadrics [14] are used to build large clusters such as the Myrinet-based Mare Nostrum IBM cluster [1] recently ranked 4th in the Top500 supercomputers [16]. However, these interconnect technologies are not following the cost/performance curve of other components, and therefore they have remained expensive relative to processor, memory and storage.

Hence the need to reduce the cost of the interconnect. One can drastically reduce the ICTN costs by decreasing the number of components – adapters, links and switches – and proportionally increasing the utilization of the remaining parts. The increased load, however, may require to operate the network as close to saturation as possible, which raises the probability of creating saturation trees and *congestion collapse* [13].

* This work was supported by CICYT under Grant TIC2003-08154-C06.

J.C. Cunha and P.D. Medeiros (Eds.): Euro-Par 2005, LNCS 3648, pp. 1024–1034, 2005.

The problem is derived from the initially blocked packets (addressed to the congested "hot" destination) that will also block packets addressed to other cold destinations. Known as Head-of-line (HOL) blocking, this is a key issue in packet switching. A blocked packet at the head of a queue prevents packets behind it from reaching idle outputs, thus leading to potentially severe throughput degradation.

2 Motivation

In addition to cost, more recent power constrains [15] also call for a higher utilization of the ICTN components (i.e. switches and links). With bursty traffic, however, increasing the link utilization can lead to saturation and performance collapse due to interference between flows and packet HOL-blocking. Factually any HOL blocking will reduce the ICTN throughput-delay performance.

First order HOL blocking results from using the FIFO queuing discipline into a switch element. Ethernet is an example of a switching standard that is widely deployed with FIFO queuing. Because Ethernet is not typically required to be lossless, as soon as HOL would occur, packets can be dropped. However, dropping packets is not an option in our study, i.e. lossless ICTNs

VOQ at switch level –referred in this paper to as VOQ_SW– solves HOL blocking at a reasonable cost [10] for single stage switches. Normally VOQ_SW strictly removes the first –but no higher– order HOL blocking [9]. The best performance would be reached by applying a global VOQ: at each queuing point there are as many queues as there are endnode destination ports. This resolves the higher order HOL blocking. Whereas attractive, the latter solution –called VOQ_Net in this paper– is not practically implementable for a large number of ports.

A new queuing discipline was described in [4, 12]. DBBM uses approximately the same number of queues as VOQ_SW or Virtual Channels (VC) [2], but has no direct association of these queues to the next stage output ports as VOQ_SW does or to bandwidth in case of VC. In this paper we investigate how the additional degree of freedom of DBBM –the mapping of queues to packet flows sorted per destination– can be exploited to address the higher order HOL blocking [9]. We do this by studying various mapping options across a variety of multistage topologies, also taking fairness into account. For reference we compare our results not only to the FIFO and the ideal global VOQ queuing mechanisms, but also to a few other relevant schemes.

The rest of the paper is organized as follows. In Sect. 3 we present the two typical queuing options in use today. In Sect. 4 DBBM and its main features are described. In Sect. 5 different VOQ and DBBM schemes are evaluated in terms of performance, scalability and fairness. In Sect. 6 conclusions are extracted and future work is outlined.

3 Traditional Queuing Options in Lossless ICTNs

As mentioned above, every switch of a modern ICTN will have a limited set of queues associated with every input and/or output port; normally the set cardinality is lower than the number of network endpoints. In some cases switches will have only one queue per input port, e.g. Myrinet. Otherwise, whenever using multiple queues per switch port, a

queuing architecture and a suitable mapping policy must be selected. Here we consider three alternatives.

The first one is to use VOQ at the switch/link level, i.e. hop-by-hop. Every input port will have as many queues as output ports, and an incoming packet will be mapped to the queue associated with the requested output port. Thus HOL-blocking at the switch and link level is eliminated. HOL -blocking can still occur, however, between flows sharing a subset of consecutive links along their paths. With switch-level VOQ and no special FC means [5], packet switching ICTNs are exposed to a form of flow interference known as high-order HOL -blocking [9].

As a second option we can use virtual channels (VC) -i.e. different queues with dedicated FC– as introduced in [2]. These channels can be load-balanced by allocating each outgoing packet to the (currently) emptiest VC. However, mis-order among the packets belonging to the same flow can be introduced, thus requiring a resequencing solution.

4 Destination-Based Buffer Management (DBBM)

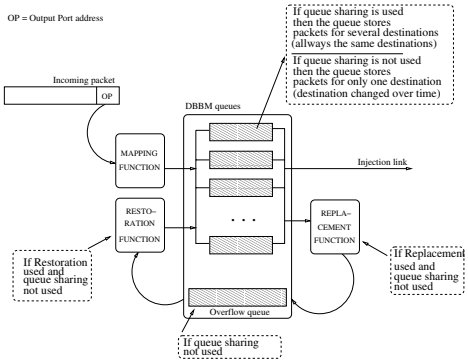
As a third option, in [4] we have introduced DBBM as a scheme to reduce HOL-blocking in ICTNs. Temporal and spatial locality in the packet destination distribution suggest that a small number of queues could be sufficient for storing all the incoming packets at each switch - while still classifying and demultiplexing them according to their destination. This allows DBBM, when used in conjunction with a suitable *mapping* strategy, to practically eliminate most –if not all– of the HOL-blocking. E.g., a simple mapping method will allow multiple flows to share –cyclically or in linear blocks– a single DBBM queue. Although some HOL-blocking will be introduced among the flows having to share the same queue, this approach can radically decrease the set of queues –and the cost– to be built and maintained in hardware.

Albeit more complex than the direct 1:1 mapping of flows to queues inherent to the VOQ disciplines, a simple DBBM mapping method could be based on the destination encoding. E.g. some bits of the destination address field in the packet header will select the DBBM queue where the packet is to be stored. As a subset of the lowest order bits are used, here this method is referred to as 'modulo' mapping; e.g, for a network with 256 destinations and 16-queue DBBM, the four least significant bits of the destination ID (8-bit field) will point to the queue to map the flow into.

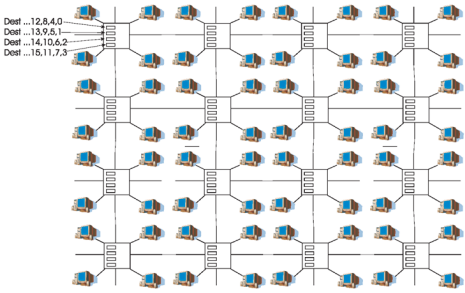
With DBBM, different mappings trade-off between performance and implementation cost - expressed in the number of hardware queues. In [12] we have performed an evaluation of the 'modulo' mapping in some multistage networks. We have shown that, to practically reach the maximum throughput, the number of DBBM queues required per switch was 8 times lower than for a full VOQ. However, DBBM's mapping policy now primarily determines the system performance.

Since the number of DBBM queues is lower than the number of ICTN endpoints, irrespective of the mapping policy in use any DBBM can –and eventually will– map flows addressed to different destinations to the same queue (i.e. intrinsic HOL-blocking).

DBBM's principle of operation is depicted in Fig. 1.(a); more details in [4, 12]. Its main functions are:



(a) The DBBM family of strategies



(b) Shared-queue DBBM with modulo-4 mapping

Fig. 1. DBBM description.

- **Queue sharing:** indicates whether packets with different destinations can be concurrently stored in the same queue or not. Both cases require a careful mapping of packet destinations to queues. In the first case to minimize the HOL-blocking. In the second one, when regular queues are no longer available, an auxiliary "overflow" queue will store either the newly incoming packets or the packets relocated from a regular queue. The overflow queue operation, however, introduces a few practical issues of implementation and re-ordering.
- **Mapping method:** determins the queue where an incoming packet will be stored. E.g., a mapping method may indicate a set of queues from which a free one will be selected (set-associative).
- **Replacement:** a binary value that indicates whether already stored packets can be relocated from a regular queue when an incoming packet requests that queue. Used only if queue sharing is not enabled.
- **Replacement function:** selects one of the queues indicated by the mapping method to (i) relocate a previously stored packet in order to (ii) store the incoming packet.
- **Restoration:** a binary value that indicates whether packets in the overflow buffers are allowed to move back to a regular queue when this has room. Used only if queue sharing is not enabled.
- **Restoration function:** selects the packets (with the same destination) to be relocated back into their initial regular queue, if restoration is enabled.

The simplest DBBM strategy allows queue sharing. With shared-queue DBBM (SQ-DBBM) each queue can only store packets for a subset of the destination ports. I.e. as if (i) the physical output ports of the network were virtually grouped into a smaller set of logical output ports and (ii) each SQ-DBBM queue stores packets destined *only* to a particular logical output port. Thus SQ-DBBM implements a 'set-VOQ' architecture organized on logical, instead of the physical, fabric ports. While this strategy does not directly avoid HOL-blocking, it may reduce it down to negligible values when a suitable mapping method is used. In-order delivery is simplified as all the packets of a flow will be mapped to the same queue, where they are stored in arrival order.

A mapping algorithm computes the address of the [SQ-DBBM] queue based on the flow ID - by decoding some bits of the destination ID in the packet header. Which bits are used depends on the mapping strategy. If the most-significant bits are decoded, consecutive port addresses are mapped to the same queue (block mapping); if the least-significant bits are decoded, consecutive addresses are mapped to different queues, cycling modulo-k (cyclic or modulo-k mapping; k=4,8,16 no. of DBBM queues). In this paper we study SQ-DBBM with modulo-4/8/16 mapping as depicted in Fig. 1.(b). Either scheme can be implemented in both inputs adapters and switches and will be referred to as DBBM 4Q, 8Q and 16Q, respectively.

Although most DBBM mappings show good results in the overall (aggregate) network throughput and/or average latency, they may also introduce notable unfairness, and possibly even starvation, between certain individual flows. To the best of our knowledge there is no analysis yet of the DBBM mapping unfairness - which here constitutes one of our two objectives.

Also in this paper we will analyze how different mappings impact the ICTN cost. We will evaluate the performance of each mapping method while varying number of endpoints attached per switch. Our goal is to maximize the system performance when minimizing the ICTN hardware resources available for a constant number of endpoints connected in 2D and 3D mesh topologies.

5 Performance Evaluation

To achieve our above stated goals we will compare the performance of 6 queuing and mapping schemes. Ordered per increasing cost, they are: (a) single FIFO (1Q), (b) DBBM 4Q, (c) DBBM 8Q, (d) load-balanced (EMPTIEST 8Q), (e) VOQ_SW and (f) VOQ_Net.

1Q (a) sets the lower bound of performance, that of a single queue with FIFO service. As the simplest queuing structure, even in single-stage fabrics with uniform traffic the 1Q scheme is theoretically limited at 58% throughput. DBBM is represented by (b,c), id est SQ-DBBM with modulo-4/8 mapping, respectively. This scheme was briefly described above. EMPTIEST 8Q (d) is a load-balancing scheduling strategy with 8 queues per input port. I.e., packets will be always mapped to the queue with the lowest current occupancy. Whereas such load-balancing is based on the queue status of the next/downstream switch, this mapping is destination/port-oblivious, and thus it represents the opposite of VOQ (DST-based, load-independent). VOQ_SW (e) is the typical VOQ scheme implemented in some modern ICTNs. It applies a link-level VOQ at every hop; i.e. switch will have at every input port as many queues as output ports. VOQ_Net (f) sets the upper bound of performance, that of an end-to-end VOQ scheme globally applied across the entire ICTN. VOQ_Net requires in every switch and IA as many queues as destinations in the network. Its main use here is as a reference for other, more practical, schemes.

5.1 Simulation Model

We have developed a detailed event-driven simulator that allows us to model the network at a level adequate for our study. The simulator models an ICTN with switches,

nodes, and links. Buffers up to 4KB are modeled for both the input and the output ports of every switch. The buffer capacity is statically divided by the number of queues defined by each of the six schemes above, resulting in a fixed size per queue.

At every switch packets are forwarded from any input queue to any output queue through a multiplexed crossbar. We have considered a crossbar bandwidth of 1.5 GB/s with a speedup of 1.5. The crossbar is controlled by a scheduler that receives requests from the packets at the head of any input queue. A requesting packet is forwarded only if the corresponding crossbar input and crossbar output are free. At each output port a weighted round-robin arbiter selects the output queue to be served.

For links we assume serial full-duplex pipelined transmissions with 1 GB/s effective bandwidth. The link-level flow control (LL-FC) protocol is credit-based; a packet can be transmitted downstream only if a credit is available. Whenever a packet frees an input buffer location a new credit is sent to the output port upstream. A similar flow control scheme has been implemented for the internal switch traversal (input-output packet forwarding). The maximum number of credits per output (input) port depends on the buffer size at the next input (output) port and the total number of queues. The LL-FC packets share the link bandwidth with data traffic.

The endpoints are connected to switches using Input Adapters (IAs). Every IA is modeled by (i) a *fixed* number N of message *admittance* queues organized in VOQ; (ii) and a *variable* number of *injection* queues organized similarly to the output ports of a switch. When a new message is generated, first it is stored completely in the admittance queue assigned to its destination; then it is segmented into 64B packets before being transferred to an injection queue. The transfer from admittance queues to injection queues are controlled by a round-robin arbiter. The transmission of packets from injection queues into the network is controlled by a weighted round-robin arbiter.

5.2 Topologies and Traffic Patterns

In [12] performance of DBBM with modulo mapping was evaluated in different multistage ICTNs. Now, 2D/3D meshes and a bidirectional multistage network (BMIN) will be evaluated for performance and fairness. In all the cases deterministic routing is used; for the 2D and 3D meshes we use the Dimension Order Routing (DOR). The BMIN is built from 8-port switches interconnected in a perfect shuffle topology.

We have defined 8 different scenarios based on synthetic traffic patterns as (partially) shown in Table 1. All the cases cause a congestion tree by oversubscribing the hotspotted endpoint; for background traffic 70% of the sources inject at 20% of link bandwidth to randomly selected destinations, while the remaining 30% of sources inject full rate to a randomly selected hotspot destination. As the background traffic shares links and queues with the flows belonging to the congestion tree, substantial HOL-blocking is introduced in multiple switches.

5.3 Evaluation Results

First we analyze the overall performance achieved by each of the 6 schemes. Then the network (cost) is reduced by removing some switches and links. Finally we focus on fairness by analyzing the goodput patterns, i.e. the traffic arrived at each destination.

Table 1. Topologies and synthetic traffic patterns evaluated.

				Traffic Injected			
				to random destinations		to hotspot	
Case	Network evaluated	# Total endpoints	# Endpoints attached per switch	% of injecting Sources	Injection rate (% of link BW)	% of injecting Sources	Injection rate (% of link BW)
#1	8×8	64	1	70%	20%	30%	100%
#2	$8 \times 8 \times 4$	256	1	70%	20%	30%	100%
#3	BMIN (64×64)	64	4	70%	60%	30%	100%
#4	4×4	64	4	70%	20%	30%	100%
#5	16×16	256	1	70%	20%	30%	100%
#6	8×8	256	4	70%	20%	30%	100%
#7	4×4	256	16	70%	5%	30%	100%
#8	$4 \times 4 \times 4$	256	4	70%	20%	30%	100%

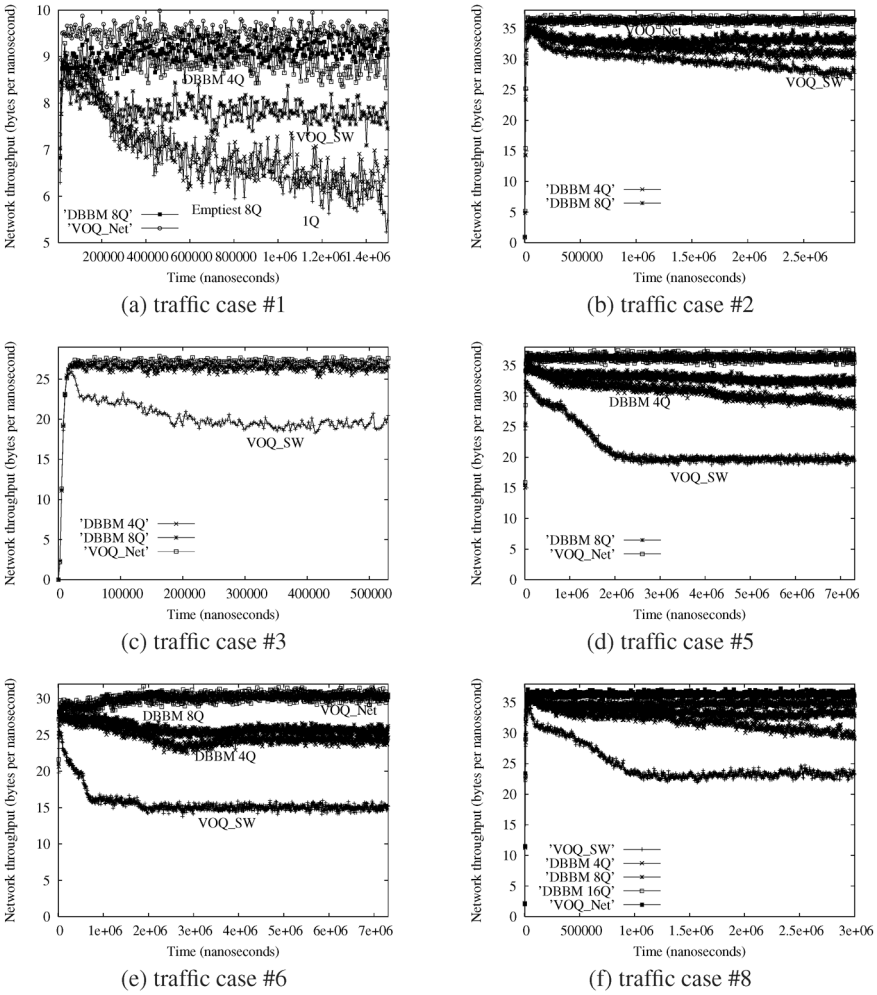
Overall Performance. Hotspot traffic (cases #1 and #2, Fig.s 2.(a) and 2.(b)) in 2D and 3D meshes show that VOQ_Net achieves the maximum throughput whereas 1Q and EMPTIEST perform the worst. Reason for EMPTIEST's poor performance: eventually most of its queues will be backlogged with packets belonging to the congestion tree. Similar results have been observed in all the studied cases that –for space reasons– can not be shown here; henceforth results for 1Q and EMPTIEST will be only plotted for case #1.

VOQ_SW achieves 77% of the VOQ_Net throughput, whereas DBBM-4Q performs better. Overall DBBM matches VOQ_Net; e.g., for #1, DBBM-4/8Q achieves 90/95% of the VOQ_Net performance. Similar for #2, despite the increase in the number of endpoints. DBBM-4/8Q achieves 86/91% of the VOQ_Net throughput. However, with 16 queues (a reduction factor of 16 of VOQ_Net queues) DBBM achieves 97% of the VOQ_Net throughput.

Whereas in Fig. 2.(c) (case #3) VOQ_SW achieves 71% of the VOQ_Net, DBBM roughly matches the VOQ_Net performance - 92% and 96% with 4, resp. 8 queues. Confirming our results from previous work, regardless of the topology, DBBM can match VOQ_Net in performance - while using a reduced set of queues.

Performance on Reduced Networks. One way to reduce the ICTN cost is by sharing: connect more endpoints to each switch, thus also increasing the HOL-blocking probability. This is confirmed in cases #5 (Fig. 2.(d)), #6 (Fig. 2.(e)), and #7 (not shown); in each, 256 endpoints attached to 2D meshes with different sizes. In all them VOQ_SW shows worse performance than DBBM. DBBM-8Q reaches 91% of VOQ_Net for cases #5 and #7, and 83% for case #6.

For a $4 \times 4 \times 4$ mesh with 256 endpoints (traffic case #8, Fig. 2.(f)) DBBM-16Q achieves 97% of the VOQ_Net. VOQ_SW reaches 61% of the VOQ_Net, whereas with a larger network ($8 \times 8 \times 4$) and with the same number of endpoints (256) it achieves 77% of the VOQ_Net throughput. On the other hand, DBBM has constant performance, independent of the network size. Again, also in reduced networks DBBM achieves the VOQ_Net performance.



(a) traffic case #1 (b) traffic case #2

(c) traffic case #3 (d) traffic case #5

(e) traffic case #6 (f) traffic case #8

Fig. 2. Accepted traffic vs. simulated time.

Fairness. Thus far DBBM and VOQ_SW exhibit good performance for hotspot traffic, DBBM being more efficient. Also they perform well when the network size is reduced. However, as they map different flows to the same queue, they introduce a degree of unfairness. We analyze this effect by plotting for each scheme the accepted traffic per endpoint.

Figures 3.(a), 3.(b), 3.(c), and 3.(d) show the traffic received by each endpoint for traffic case #4 (4×4 mesh with 4 nodes/switch), when VOQ_Net, VOQ_SW, DBBM-8Q and EMPTIEST scheme is used, respectively. The *highest bar represents the hotspot* (endpoint 30), which reaches 90% of received traffic (axes are truncated at 50%). With VOQ_Net every destination, except the hotspot, receives roughly the same goodput. With VOQ_SW, all the flows that share two consecutive links with the congested flow suffer from HOL-blocking. Thus the number of affected flows does not only depend

on the mapping function, but also on routing algorithm and topology. Every 4 consecutive endpoints exhibit similar percentages of accepted traffic, since 4 is the number of rows and columns used in the case #4 topology. The routing algorithm used was DOR. This, together with VOQ_SW scheme, causes that most of the flows sharing a column or a row in its path with the packets addressed to the congested destination will be allocated to the same queue. Hence the reduction in the number of received packets by the 'victimized' destinations. With DBBM, the number of affected flows depends on the number of queues, but not on routing. Figure 3.(c) shows that one out of every eight flows receives less packets than the others. This is because only one queue out of eight is used to map congested packets. Only those destinations which share the queue with this congested destination will experience HOL-blocking, and thus, will exhibit a reduction in the number of received packets. 7 out of 64 destinations are affected by the congestion tree, with a goodput reduction of 8%. For VOQ_SW, however, half of destinations suffer - reduced their respective accepted traffic rate below 10%. As the number of endpoints attached per switch increases (reducing the network size), this effect will be amplified by VOQ_SW - i.e. more destinations will be affected by one congested destination. With DBBM the effect remains isolated to 'victim' destinations.

With EMPTIEST all the destinations are equally affected by the HOL-blocking that the congested destination introduces; in Fig. 3.(d) congestion spreads across all the switches (in the path toward their destination).

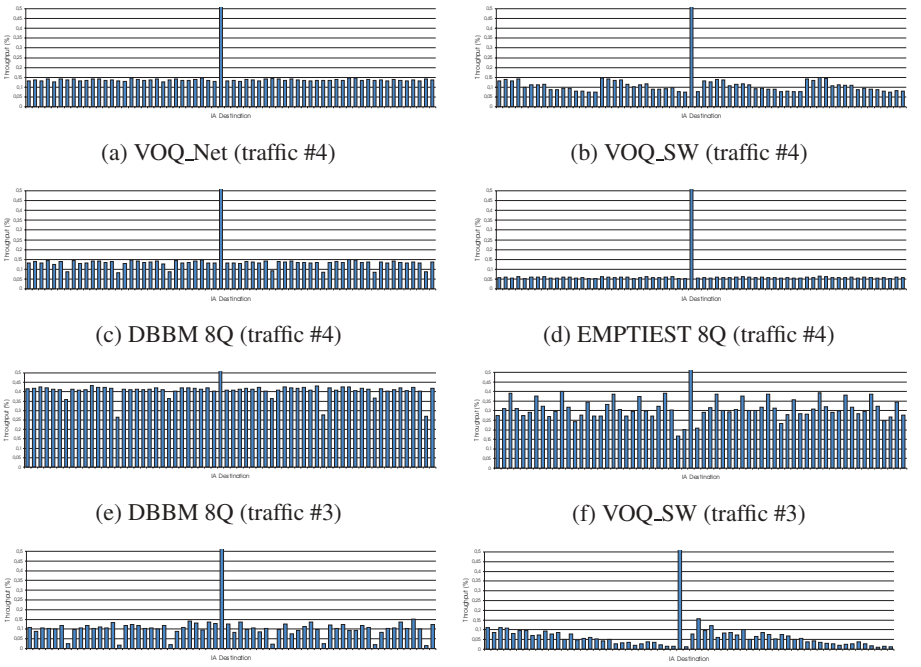
Figures 3.(e) and 3.(f) show the traffic received by the destinations for case #3 (BMIN network), when DBBM-8Q and VOQ_SW are applied, respectively. The behaviour for cases #3 and #4 is similar. The pattern of affected flows repeats independently of the traffic case. The main difference is in throughput. With DBBM, the number of affected flows depends only on the number of used queues, whereas with VOQ_SW more flows are affected by unfairness. In the latter scheme all the flows suffer from high-order HOL-blocking derived from the hotspot. Once more we see how the number of affected flows does not only depend on the mapping function, but also on the routing algorithm.

We observe that in certain situations DBBM is unfair to some flows. E.g., Fig.s 3.(g) and 3.(h) show the received traffic for the first 64 endpoints for case #6. For DBBM the affected destinations are the same as before but they have decreased their reception rate below 5%. With the same traffic, VOQ_SW behaves worse: half of the endpoints have a traffic percentage lower than 5%.

To conclude, excepting VOQ_Net, all the other schemes introduce some degree of unfairness under high load and congestion. However, DBBM is the only one that keeps the unfairness degree independent of the topology and routing used.

6 Conclusions

In order to reduce HOL-blocking a number of queuing schemes and mapping methods have been proposed. Theoretically only a full end to end VOQ, or at least a subset of non-interfering flows [3] is able to eliminate completely HOL-blocking. However, this solution is not scalable to large ICTNs. In order to overcome these problems, other mapping strategies have been proposed and evaluated. In these evaluations we have studied the trade-offs between performance and the number of required queues. We



(a) VOQ_Net (traffic #4) (b) VOQ_SW (traffic #4)

(c) DBBM 8Q (traffic #4) (d) EMPTIEST 8Q (traffic #4)

(e) DBBM 8Q (traffic #3) (f) VOQ_SW (traffic #3)

(g) DBBM 8Q (traffic case #6) (h) VOQ_SW (traffic case #6)

Fig. 3. Accepted traffic per destination.

have analyzed the unfairness that a mapping strategy can introduce in lossless ICTNs. Also, we have analyzed how the different mapping methods can help in reducing the cost of the ICTNs.

Simulation results have confirmed that both link/switch-level VOQ (VOQ_SW) and destination-oblivious load-balancing (EMPTIEST) schemes suffer from high-order HOL-blocking. On the other hand, for a moderate increase in complexity DBBM shows clear improvements, linearly proportional to the number of operating queues. Independent of the network size, DBBM with 8 queues has achieved roughly the same throughput as the 'ideal' VOQ, while using only a small fraction of the queues.

Excepting VOQ_Net, all the mapping strategies introduce some degree of unfairness. However, DBBM kept the unfairness independent of the topology and the routing in use. For DBBM, the number of affected flows by the congestion tree depends on the number of used queues, whereas for VOQ_SW the affected flows not only depend on the mapping function but also on the routing algorithm. As future work we are currently exploring other DBBM schemes, such as combinations of block and cyclical mapping.

Acknowledgments

We are indebted to Ton Engbersen and Ronald Luijten for significant contributions and careful review.

References

1. Barcelona Supercomputing Center (BSC), http://www.bsc.org.es, Nov. 2004.
2. W. J. Dally, *Virtual-channel Flow Control,* in Proceedings of the 17th Int. Symp. on Computer Architecture, ACM SIGARCH vol. 18, no. 2, pp. 60-68, May 1990.
3. W. J. Dally and B. Towles *Principles and Practices of Interconnection Networks,* San Francisco, CA, Morgan Kaufmann, 2004.
4. J. Duato, J. Flich, and T. Nachiondo, *Cost-Effective Technique to Reduce HOL-blocking in Single-Stage and Multistage Switch Fabrics,* Euromicro Conference on Parallel, Distributed and Network-based Processing, pp. 48-53, Feb. 2004.
5. J. Duato, I. Johnson, J. Flich, F. Naven, P. García, and T. Nachiondo, *A New Scalable and Cost-Effective Congestion Management Strategy for Lossless Multistage Interconnection Networks,* Int. Symp. on High-Performance Computer Architecture, Feb. 2005.
6. Earth Simulator Center. http://www.es.jamstec.go.jp/esc/eng/index.html.
7. IBM BG/L Team, *An Overview of BlueGene/L Supercomputer,* ACM Supercomputing Conference, 2002.
8. InfiniBand Trade Association, InfiniBand Architecture. Specification Volume 1. Release 1.0. Available at http://www.infinibandta.com/.
9. M. Jurczyk and T. Schwederski, *Phenomenon of Higher Order Head-of-Line Blocking in Multistage Interconnection Networks under Nonuniform Traffic Patterns*, IEICE Transactions on Information and Systems, Special Issue on Architectures, Algorithms and Networks for Massively Parallel Computing, Vol. E79-D, No. 8, pp. 1124-1129, August 1996.
10. C. Minkenberg, *On Packet Switch Design,* Ph.D. Thesis, Eindhoven University of Technology, Sep. 2001.
11. Myrinet, 2000 Series Networking. Available at http://www.cspi.com/multicomputer/ products/2000 _series_networking/ 2000_networking.htm.
12. T. Nachiondo, J. Flich, and J. Duato, *Efficient Reduction of HOL blocking in Multistage Networks,* Workshop on Communication Architecture for Clusters (CAC 2005), April 2005.
13. G. F. Pfister and V. A. Norton,*Hot Spot Contention and Combining in Multistage Interconnection Networks*, IEEE Transactions on Computers, vol. C-34:10,pp. 943-948, Oct.1985.
14. Quadrics QsNet. Available at http://doc.quadrics.com.
15. L. Shang, L. S. Peh, and N. K. Jha, *Dynamic Voltage Scaling with Links for Power Optimization of Interconnection Networks*, Proc. Int. Symp. on High-Performance Computer Architecture, pp. 91–102, Feb. 2003.
16. http://www.top500.org