

On the Correct Sizing on Meshes Through an Effective Congestion Management Strategy^{*}

Pedro Javier García¹, José Flich², José Duato², Francisco José Quiles¹,
Ian Johnson³, and F. Naven³

¹ Dept. de Informática. Escuela Politécnica Superior
Universidad de Castilla-La Mancha 02071-Albacete, Spain
{pgarcia,paco}@info-ab.uclm.es

² Dept. of Computer Science, Univ. Politécnica de Valencia 46071-Valencia, Spain

³ Xyratex, Haven, UK

Abstract. Interconnection networks used in clusters of PCs are often dimensioned with certain restrictions. One restriction could be the reduction of power consumption and overall cost. In this sense, the network size must be reduced. Another restriction is to guarantee that the system offers a minimum bandwidth. In this case, the network size must be increased. In both cases, the head-of-line (HOL) blocking effect (related to network congestion) may appear, degrading network performance and thus, preventing the correct sizing of the network. Therefore, some mechanisms should be implemented for reducing or eliminating this problem, in order to dimension the network as desired while keeping network performance at maximum. In this paper we analyze the impact on network performance when using different mechanisms for handling HOL blocking when interconnection networks with mesh topology are dimensioned in several ways. We show that the previously proposed RECN congestion control mechanism is key in order to efficiently eliminate HOL blocking in meshes and, therefore, it allows the correct network sizing.

1 Introduction

In the last years, clusters of PCs are becoming a challenging alternative to massive parallel computers dedicated to high performance computing (HPC). Also, cluster of PCs are becoming an alternative to build large Internet servers. The attractive performance/cost ratio of PCs makes building large cluster-based systems an interesting solution. Examples of clusters for HPC can be obtained from the top500 list [1] where 294 systems out of 500 are clusters of PCs (three in the top five list). Also, commercial Internet portal servers using clusters are being used at AOL, Google, Amazon or Yahoo.

In such systems, the interconnection network plays a key role in the performance achieved. For this, it is common to use high-speed interconnect networks like Myrinet [2], InfiniBand [3], and Quadrics [12]. Such networks provide high

^{*} This work was supported by CICYT under Grant TIC2003-08154-C06 and by UPV under Grant 20040937.

bandwidth and low latencies. However, these networks offer additional features. One of the most interesting feature is that network topology can be as flexible as needed. Indeed, nothing prevents a network designer to attach several endnodes to the same switch or to build a complete irregular network. This capability makes system scalability a reality in cluster-based systems.

However, new problems arise in these networks that may affect scalability. One of the main problems is congestion. As these networks usually do not drop packets (lossless networks), whenever a packet blocks (as it requests a resource not available) it may block other packets stored behind it, even in the case they could make forward progress (they would request available resources). This effect is referred to as head-of-line (HOL) blocking. HOL blocking will be propagated quickly because flow control spreads the congestion, thus collapsing the network.

The de facto solution to avoid congestion has traditionally been to overdimension the interconnection network. The introduction of wormhole switching [4] made feasible to integrate a switch into a single chip. In turn, this allowed such a dramatic increase in link bandwidth that interconnection networks could be overdimensioned at a low cost. As a consequence, reported network utilization in parallel machines and clusters has been quite low for almost two decades.

On the other hand, power consumption is becoming increasingly important. As VLSI technology advances and link speed increases, interconnects are consuming an increasing fraction of the total system power [5]. Taking this into account, there are only two ways of reducing network power consumption: a) reducing the number of links in the network (and using the remaining links more efficiently), and b) using some frequency/voltage scaling technique to reduce link power consumption [5]. Unfortunately, dynamic voltage scaling (DVS) techniques are quite inefficient due to their extremely slow response in the presence of traffic variations and the suboptimal frequency/voltage settings during transitions [6]. In fact, a recent paper shows that static voltage scaling (SVS) combined with adaptive routing achieves higher performance and lower power consumption than DVS techniques, as far as the network does not saturate [6].

Thus, the simplest way to reduce cost and power consumption is reducing the network size (less switches and links). Obviously, as the network size is reduced, the offered network bandwidth will be also lower. Therefore, the network must be dimensioned accordingly to the estimated bandwidth required by the endnodes of the system. For instance, the network designer can rearrange the endnodes and attach a higher number of endnodes at each switch. Imagine a system with 256 endnodes attached through a 16×16 mesh network using links with 1Gbps capacity. This network can be reduced to a 4×4 mesh network (attaching groups of 16 endnodes to the same switch) only if the available network bandwidth (16 Gbps^1) is enough for communicating the endnodes. Notice that this can be the case for a server system where the traffic is highly local (applications are exclusively run in groups of 16 endnodes attached to the same switch).

¹ Theoretically, for uniform traffic, the offered bandwidth of a mesh network is $2 \times BW_{bisection}$.

However, as the network size is reduced, the link utilization will be higher and, thus, the network will work closely to the saturation point. Additionally, notice that traffic is usually bursty (temporal congestion trees will be common). In this scenario, it will be usual that the network will be working beyond its saturation point. Therefore, it will be required an effective congestion control mechanism in order to allow an effective reduction in the network size.

Another restriction for network dimensioning is when more bandwidth is required by the system. This is the case of HPC systems where it is expected to have an intense traffic among all endnodes (working on the same application). Therefore, the way to dimension these systems is to put as much switches and links as necessary to meet the required traffic conditions. As an example, a system with 16 endnodes attached to a 4×4 mesh network can be scaled up by building a larger system with 256 endnodes attached to a 16×16 mesh network. At first sight it seems that there will be no problems when using more network components. This can be deduced from the fact that the utilization of links will be lower (as the network is overdimensioned). However, notice that as network size is increased, the average path length will also increase. Therefore, packets will travel longer distances and, in the presence of congestion trees, they will have more chances of being affected by the HOL blocking effect. Thus, again it will be necessary an effective congestion management technique that eliminates those effects in order to allow an effective increase in the network size without degrading performance.

HOL blocking has been studied for long, and very efficient techniques exist for avoiding it within a single switch (e.g. virtual output queues (VOQs) [7], dynamically allocated multiqueues (DAMQs) [8], congestion buffers [9], etc.). These techniques work by allocating separate buffers for packets destined to different output ports or by providing a way for non-blocked packets to pass blocked packets. However, these solutions either do not work efficiently for multihop networks (e.g. DAMQs) or are not scalable at all because the number of buffers required at every switch port increases linearly with the number of endpoints attached to the network. Thus, overall buffer capacity increases at least quadratically with the number of network endpoints. Although some implementations of network-level VOQs exist [10], they are very expensive and do not scale, and may even become infeasible beyond certain network size.

An intermediate solution is to use VOQ at the switch level. With this solution, every switch port has as many queues as output ports of the switch, and whenever a packet arrives to the port it is stored in the queue assigned to its requested output port. Although this solution does not eliminate completely HOL blocking it can minimize its impact. This solution will be referred to as VOQ_{sw} .

In [13] we proposed a new congestion management technique, referred to as RECN (Regional Explicit Congestion Management), focused in eliminating the HOL blocking effect produced by congestion trees rather than eliminating congestion. In particular, once incipient congestion is detected within the network, RECN assigns new queues to the congested points and thus, the congested traffic is isolated and the HOL blocking is avoided.

A recent technique has also been proposed in [11], referred to as DBBM (Destination-Based Buffer Management). In this approach, the whole set of network endpoints are divided into several sets, and all the packets addressed to a set of destinations are stored in the same queue. Thus, HOL blocking is avoided among destinations grouped in different sets. Notice that RECN differs from DBBM in the sense that dynamic queues are allocated for congestion trees whereas in DBBM queues are statically allocated to groups of destinations. Although DBBM is very efficient in the general case, there may be some special traffic situations that may introduce HOL blocking.

In this paper we take on different challenges. Firstly, we apply the RECN mechanism to mesh networks. By doing this, we analyze the benefits that RECN will give to applications run on such networks. Secondly, we will analyze up to what extent the traditional VOQ_{sw} solution is able to efficiently handle the HOL blocking introduced when the network is dimensioned in different ways (downsizing the network to reduce cost and power consumption and upsizing the network to achieve a certain bandwidth). As a third challenge we will evaluate RECN as a way to allow an efficient system sizing. We will show that, contrary to the VOQ_{sw} solution, RECN allows to achieve ideal network sizing.

The rest of the paper is organized as follows. In Section 2, RECN is described. In Section 3, scalability issues by using RECN and VOQ_{sw} are analyzed in detail by means of simulation results of network performance under different situations of traffic, network size and congestion control mechanisms used. Finally, in Section 4 some conclusions are drawn.

2 RECN Description

RECN (Regional Explicit Congestion Notification)[13] is a new congestion management strategy that focuses on eliminating the main negative effect of congestion: the HOL blocking. In order to achieve it, RECN detects congestion and dynamically allocates separate buffers for each congested flow, assuming that packets from non-congested flows can be mixed in the same buffer without producing significant HOL blocking.

RECN requires the use of a kind of deterministic routing that makes possible to address a particular network point from any other point in the network. In fact, RECN has been designed for PCI Express Advanced Switching (AS) [14, 15], a technology that uses source routing. AS packet headers include a turnpool made up of 31 bits, that contains all the turns (offset from the incoming port to the outgoing port) for every switch in a route. Thus, a switch, by inspecting the appropriate turnpool bits, can know in advance if a packet that is coming through one of its incoming ports will pass through a particular network point.

In order to separate congested and non-congested flows, RECN adds a set of additional queues at every input (ingress) and output (egress) port of a switch. These queues (referred to as Set Aside Queues or SAQs) are dynamically allocated and used to store packets passing through a congested point. To do this, RECN associates a CAM memory to each set of queues. The CAM contains

all the control info required to identify the congested point and to manage the corresponding SAQ. In the aim of guaranteeing in order delivery, whenever a new SAQ is allocated, forwarding packets from that queue is disabled until the last packet of the standard queue (at the moment of the SAQ allocation) is forwarded. This is implemented by a simple pointer associated to the last packet in the standard queue and pointing to the blocked SAQ.

Whenever an ingress or egress queue receives a packet and fills over a given threshold, a RECN notification is sent to the sender port indicating that an output port is congested. When congestion is detected at the egress side, the congested point is this egress port. In order to detect congestion at the ingress side, the standard queue is replaced by a set of detection queues. The detection queues are structured at the switch level: there are as many detection queues as output ports in the switch, and packets heading to a particular output port are directed to the associated detection queue. So, when a detection queue reaches a threshold, it means that the associated output port is congested.

RECN notifications also include the routing information (a turnpool) to reach the congested output port from the notified port. Upon reception of a notification, each port maps a new SAQ and fills the corresponding CAM line with the received turnpool. From that moment, every incoming packet that will pass through the congested point (easily detected from the turnpool of the packet) will be stored in the newly allocated SAQ, thus eliminating the HOL blocking it may cause. If a SAQ becomes subsequently congested, a new notification will be sent upstream to some port that will react in the same way, allocating a new SAQ, and so on. As the notifications go upstream, the included information indicating the route to the congestion point is updated accordingly, in such a way that growing sequences of turns (turnpools) are stored in the corresponding CAM lines. So, the congestion detection is quickly propagated through all the branches of a congestion tree. Apart from the SAQs allocated due to notifications, when congestion is detected at the ingress side, a SAQ is also allocated at this port, and the detection queue and the new allocated SAQ are swapped.

RECN keeps track (with a control bit on each CAM line) of the network points that are a leaf of a congestion tree. Whenever a SAQ with the leaf bit set empties, the queue is deallocated and a notification packet is sent downstream, repeating the process until the root of the congestion tree is reached.

Regarding flow control, RECN uses for each individual SAQ a level-based flow control (Xon/Xoff). This mechanism is different from the credit-based flow control used for standard queues, that considers all the unused space of the port data memory available for each individual queue. Xon/Xoff scheme guarantees that the number of packets in a SAQ will be always below a certain threshold. Further details about RECN can be found in [13].

3 Performance Evaluation

In this section we will evaluate the performance of the network, in several scenarios of network size, traffic load, and different mechanisms focused in reducing the HOL-blocking: VOQ at the network level (VOQnet), VOQ at the switch le-

vel (VOQ_{sw}) and RECN. For this purpose we have developed a detailed event-driven simulator that allows us to model the network at the register transfer level. Firstly, we will describe the main simulation parameters and the modeling considerations we have used in all the evaluations. Secondly, we will present the evaluation results and analyze them.

3.1 Simulation Model

The simulator models square meshes consisting of a variable number of switches and bidirectional links that connect a variable number of endnodes. Specifically, we have used five network configurations, shown in Table 1. In all the cases X-Y deterministic routing is used.

Table 1. Network configurations and traffic cases evaluated.

Network	Top	#sw	#endnodes/sw					
#1	16×16	256	1					
#2	8×8	64	4					
#3	4×4	16	16					
#4	8×8	64	1					
#5	4×4	16	1					

Traffic	normal traffic		congestion tree	
	#sources	dst	#sources	dst
#1	100%	random	-	-
#2	87.5%	random	12.5%	hot-spot
#3	75%	random	25%	hot-spot

Due to the different number of endnodes per switch, the number of bidirectional ports of the switches varies depending on network configuration. At these ports, the simulator models a 128 KB memory for both input and output ports. When VOQ is used, the total memory size per port is equally divided into as many queues as endnodes (VOQ_{net}) or into as many queues as ports in the switch (VOQ_{sw}).

RECN has been modeled in detail. The memory is shared by all the queues (detection or standard queues and SAQs) defined at this port at a given time, in such a way that memory cells are dynamically allocated (or deallocated) for any queue when it is required. In order to support the RECN detection at ingress ports, several detection queues are defined at ingress ports, and one standard queue at egress ports.

To model the links, we have assumed serial full-duplex pipelined links with 8 Gbps bandwidth. Inside every switch, packets are forwarded from any input queue to the corresponding output queue through a multiplexed crossbar. The crossbar access is controlled by an arbiter that receives requests from packets at the head of any input queue. A requesting packet is forwarded only when the corresponding crossbar input and crossbar output are free. Requests from packets in detection queues have preference over requests from packets in SAQs.

Regarding flow control, we have modeled several mechanisms. RECN uses credit-based flow control at the port level. So, whenever a new packet is transmitted from an output port to the corresponding input port of the next switch, a credit is consumed. When a packet leaves an input port, a new credit is granted

to the previous output port at the upstream switch or endnode. Output port credits can be consumed for transmitting packets from the standard queue or SAQs at this port. A similar flow control scheme has been implemented for the internal (input-output) switch packet forwarding. So, the maximum number of credits per output (or input) port depends on the total memory size at the next input (or output) port. In addition, Xon/Xoff flow control has been modeled for limiting the injection of packets between SAQs. When the occupancy of a SAQ grows up to a given threshold², an Xoff packet is sent to the corresponding upstream SAQ. Any SAQ that receives an Xoff packet stops the injection of packets until the reception of an Xon packet. Any SAQ that previously sent an Xoff packet sends an Xon packet when its occupancy goes below a given threshold. On the other hand, a credit-based flow control at the queue level has been implemented for the VOQs mechanisms. In these cases, the maximum number of credits per queue depends on the total memory size at the next input (or output) port and the number of queues at this port. Flow control packets have been modeled and they share the link bandwidth with data packets.

Endnodes are connected to switches using Input Adapters (IAs). Every IA is modeled with a fixed number of message admittance queues following a VOQnet scheme, and a variable number of injection queues, that follow a scheme similar to that of the output ports of a switch. So, SAQs can be allocated dynamically at the output side of input adapters when the RECN mechanism is used. When a message is generated, it is stored completely in the admittance queue assigned to its destination, and is packetized before being transferred to an injection queue. We have used 64-byte packets. The transfer from admittance queues to injection queues is controlled by an arbiter that follows a round-robin scheme. The injection of packets from injection queues to the network is also controlled by an arbiter that selects the next packet to be transmitted, using a round robin scheme among all the queues.

3.2 Traffic Load

For all the network configurations we have made experiments under several traffic scenarios. We have used synthetic traffic patterns modeling simple but significant traffic situations in order to check how the analyzed mechanisms react to different traffic loads. Table 1 shows the traffic parameters of each traffic case.

For each traffic case, there is a variable percentage of sources injecting traffic to random destinations. This percentage is 100% in traffic case #1, but it is lower in traffic cases #2 and #3. In these cases, the rest of sources inject traffic to the same destination (endnode 32 for network configurations #1, #2, #3 and #4; endnode 10 for network configuration #5). Thus, in these cases, congestion trees will be formed in the network. All the endnodes inject traffic at the same rate during all the simulation period. This rate has been varied in an incremental way for obtaining a metric of the network performance under different loads of normal and congested traffic.

² Although several thresholds have been tested, all of them gave us similar performance results. Therefore, we fixed threshold to 1% of total port memory.

For all the cases evaluated, the network relative throughput³ as a percentage will be shown (for different injection rates). This will allow direct comparisons among different network configurations.

3.3 Performance Evaluation

In the following subsections we will show simulation results that allow us to analyze the impact of RECN and VOQsw when used as a mechanism to reach the maximum performance when sizing the network in different ways. Moreover, results for VOQnet will be also shown as a reference for maximum performance (no possible HOL blocking). Specifically, we will analyze first the impact of such mechanisms when the network is downsized while keeping constant the number of endnodes. Next, we will analyze their impact when the network size is increased in order to achieve higher bandwidth.

Reducing Network Cost and Consumption. Figure 1 shows the performance results for network configurations #1, #2 and #3 for different traffic patterns. For traffic case #1 (Figures 1.a, Figures 1.b, and Figures 1.c), all the mechanisms evaluated achieve roughly the maximum performance, although the performance slightly decreases when VOQsw is used for high traffic loads. This is because VOQsw does not correctly handle all the traffic, and some HOL blocking appears. Additionally, for higher traffic loads (beyond saturation point; not shown), VOQsw even significantly degrades performance. On the other hand, in these situations, RECN keeps relative throughput above 90%.

From the previous results, it could be deduced that VOQsw is an effective mechanism that allows to achieve maximum performance for low or medium traffic loads. However, real traffic is usually bursty, and a different behavior could be expected. Indeed, Figures 1.d, 1.e, and 1.f show the results for network configurations #1, #2 and #3 when a light hot-spot traffic pattern (traffic case #2) is present in the network. For all the network configurations, VOQsw is not able to obtain maximum performance, regardless of the injection rate. Indeed, for network configuration #1, it achieves only 50% of relative throughput. It can be seen that, as network size decreases, VOQsw tends to achieve higher relative throughput. This is due to the shorter average routes on the network, that reduce the HOL blocking effect. On the opposite side, RECN achieves roughly maximum throughput (90% of relative network throughput in the worst case). So, RECN eliminates the HOL blocking introduced by the congestion tree and, as a consequence, it uses efficiently all the bandwidth offered by the network.

For a more intense hot-spot traffic pattern (traffic case #3), the behavior is similar but more dramatic for VOQsw. Results for this traffic case are shown in Figures 1.g, 1.h, and 1.i for network configurations #1, #2 and #3, respectively.

³ Network relative throughput is computed as the network absolute throughput divided by the maximum theoretical throughput ($2 \times BW_{bisection}$). The maximum theoretical throughput for the $N \times N$ mesh is $4 \times N$ bytes/ns.

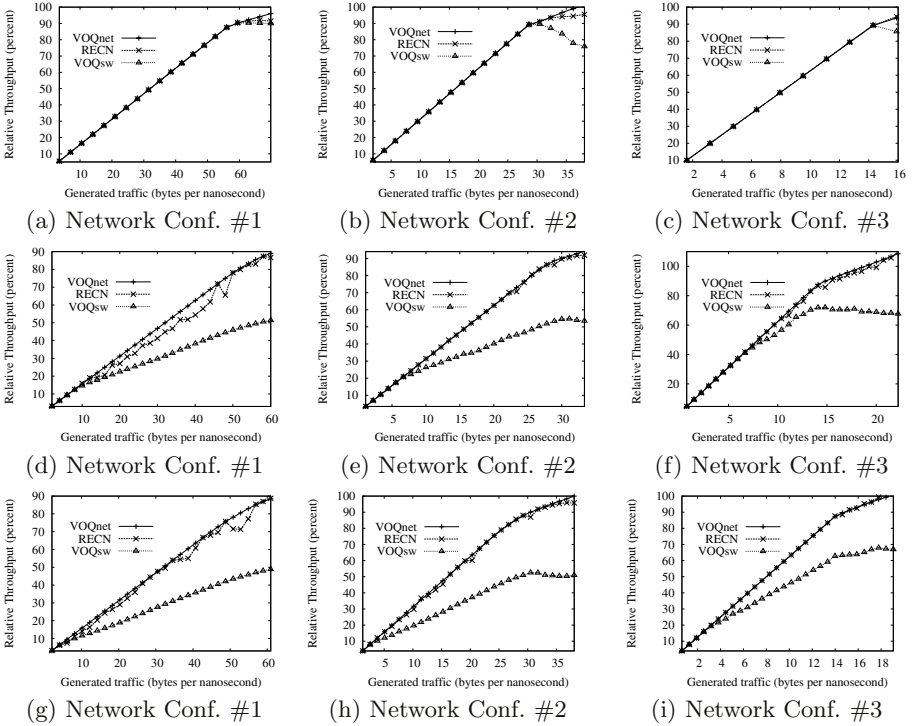


Fig. 1. Relative throughput for network configurations #1, #2 and #3, traffic case (a,b,c) #1 (uniform), (d,e,f) #2 (light hot-spot), and (g,h,i) #3 (heavy hot-spot).

To sum up, RECN allows to achieve the maximum bandwidth offered by the network by virtually eliminating the HOL blocking introduced by the higher use of links when network is downsized. VOQsw is far from achieving the maximum offered network bandwidth as it does not handle properly HOL blocking.

Increasing Network Size and Bandwidth. Now, we will evaluate how VOQsw and RECN behave when they are used as a technique to achieve maximum throughput when overall network bandwidth is increased by upsizing the network. For all the network configurations evaluated in this section, one endnode is attached to each switch. Thus, as the network size increases, the number of endnodes also increases, and so does the average length of routes (potentially increasing HOL blocking).

Figure 2 shows the performance for different network configurations (#1, #4, and #5) and different traffic patterns. For uniform traffic pattern (traffic case #1, Figures 2.a, 2.b, and 2.c) it can be deduced that the VOQsw solution behaves roughly as well as RECN and VOQnet. From this fact, it could be deduced also that VOQsw is a good solution in order to efficiently upsize the network. However, again, this deduction is not valid when a congestion spot is present in the network (modeling bursty traffic). For a light congestion tree (traffic case

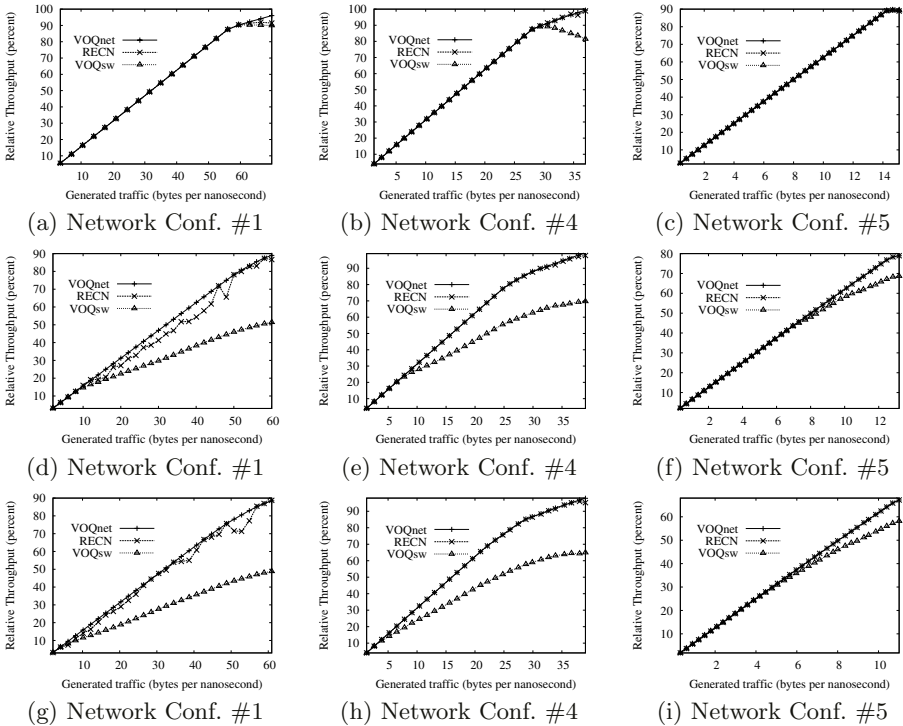


Fig. 2. Relative throughput for network configurations #1, #4 and #5, traffic case (a,b,c) #1 (uniform), (d,e,f) #2 (light hot-spot), and (g,h,i) #3 (heavy hot-spot).

#2, Figures 2.d, 2.e, and 2.f) in the network, again, the VOQsw solution suffers HOL blocking that is not solved, and therefore, it does not achieve maximum offered bandwidth. Relative network throughput is always lower than 70%.

On the other hand, RECn is able to keep the maximum performance for all the network configurations. It has to be noted that RECn achieves its goal by using a maximum of 8 SAQs. Figure 3 shows, for traffic case #3 and network configurations #1, #4 and #5, the maximum SAQ utilization at ingress and egress sides. It can be seen that the maximum number of SAQs used is below 8 for most of the traffic loads.

4 Conclusions

We have shown the importance of using a suitable congestion control mechanism for virtually eliminating the HOL blocking that appears by dimensioning in several ways interconnection networks with mesh topology. From the results presented in this paper, we can deduce that network performance is affected by HOL blocking when the network is sized in certain ways and VOQsw is used. On the contrary, the RECn mechanism allows to dimension the network in any way while keeping network performance roughly at maximum, due to the

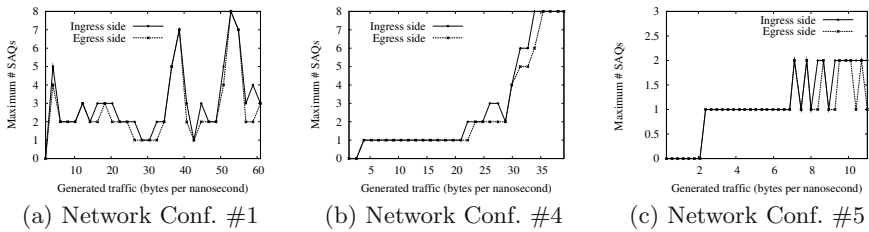


Fig. 3. Maximum number of SAQs used. Configurations #1, #4, #5, traffic case #3.

efficient handling of the HOL blocking problem. Moreover, this can be achieved in a scalable way. Therefore, RECN allows to reduce network size, cost and power consumption or to increase network size and overall bandwidth without degrading network performance.

References

1. Top 500 supercomputer list, <http://www.top500.org>.
2. N. J. Boden et al, "Myrinet - A gigabit per second local area network," *IEEE Micro*, pp. 29–36, February 1995.
3. InfiniBandTM Trade Association, <http://www.infinibandta.com>.
4. W. J. Dally and C. L. Seitz, "The Torus Routing Chip," *Journal of Distributed Computing*, vol. 1, no. 3, pp. 187–196, Oct. 1986.
5. L. Shang, L. S. Peh, and N. K. Jha, "Dynamic Voltage Scaling with Links for Power Optimization of Interconnection Networks", in *Proc. Int. Symp. on High-Performance Computer Architecture*, pp. 91–102, Feb. 2003.
6. J. M. Stine and N. P. Carter, "Comparing Adaptive Routing and Dynamic Voltage Scaling for Link Power Reduction", *Computer Architecture Letters*, vol. 3, June 2004.
7. T. Anderson et al, "High-Speed Switch Scheduling for Local-Area Networks", *ACM Transactions on Computer Systems*, vol. 11, no. 4, pp. 319–352, Nov. 1993.
8. Y. Tamir and G. L. Frazier, "Dynamically-Allocated Multi-Queue Buffers for VLSI Communication Switches", *IEEE Trans. on Computers*, vol. 41, no. 6, June 1992.
9. A. Smai and L. Thorelli, "Global Reactive Congestion Control in Multicomputer Networks", in *Proc. 5th Int. Conference on High Performance Computing*, 1998.
10. W. J. Dally, P. Carvey, and L. Dennison, "The Avici Terabit Switch/Router", in *Proc. Hot Interconnects 6*, Aug. 1998.
11. 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.
12. Quadrics QsNet. Available at <http://doc.quadrics.com>
13. J. Duato, I. Johnson, J. Flich, F. Naven, P.J. García, T. Nachiondo, "A New Scalable and Cost-Effective Congestion Management Strategy for Lossless Multistage Interconnection Networks", in *Proc. 11th Int. Symp. High-Performance Computer Architecture*, Feb. 2005.
14. "Advanced Switching for the PCI Express Architecture". White paper. Available at <http://www.intel.com/technology/pciexpress/devnet/AdvancedSwitching.pdf>
15. "Advanced Switching Core Architecture Specification". Available at <http://www.asi-sig.org/specifications> for ASI SIG.