Journal of Advance Computational Research
Vol. 3 No. 2 (July-December, 2018)

Dynamic allocation & Deallocation of buffering in High Speed Switch

Vishnu Priya. A[1] and Senthil Kumar. P[2]
[1*]Research Scholar, Anna University of Technology, Chennai, India
[2]Professor, Saveetha Engineering College, Chennai, India

RECEIVED: January 29,2017. Revised March 06, 2017

Abstract: The evolution of interconnection network represents the dependency of the total system cost and power consumption. To achieve a better performance in High-speed network design, the number of components or resources must be reduced. This leads to a necessity of efficient congestion management technique. RECN (an efficient Head-of-Line blocking elimination technique) is a cost effective Switching architecture to face the challenges of congestion management, it has been recently proposed for Advanced Switching (AS). RECN detects the formation of congestion trees, dynamically allocates queues for storing congested packets, and thus eliminates the HOL blocking introduced by congestion trees. These queues are deallocated when congestion vanishes. In this paper, an enhanced RECN version, specifically distributed queue deallocation mechanism for RECN-IQ (Input queue switch architecture: only queues at input port of a switch) that reduces the number of required resources (Queues), reduces the memory requirement and does not require the use of control packets for the deallocation of SAQ's (Set Aside Queues-dynamically assigned for congested points) are proposed. Here a new congestion notification mechanism, where flow control packets are used to notify congestion, is used. This simplifies the implementation of RECN-IQ deallocation strategies in AS. Regarding the performance it leads to a significant reduction of the data memory area required at each port in the reduction factor of 5 times than RECN-CIOQ (Combined Input Output Queue - have queues at both Input port & Output port of a switch) and avoids the use of explicit congestion notifications and token-exchanging packets.

Key words: High-speed interconnection networks, Congestion management, Switching, Queuing

I. INTRODUCTION

Amount of traffic on the Internet is continuously growing at a high rate. Every day we hear of new businesses on the Internet, new services that are offered online, and new ways that the Internet can be utilized. Regarding the growth in interconnection network, problem arrived are the power consumption and cost utilization is increased. For this problem the only solution is to reduce the no of components or resources, because of this reduction link utilization between the components are increased. So there is a need of congestion management, in now a day's Congestion Management in interconnection networks is a widely known problem. Related to this the some of Congestion eliminating techniques are researed. Data passes through communication

Networks, such as the Internet, in the form of packets. These packets of information have a source and a destination, and travel in the network through intermediate nodes. When different packets request the same resource, typically output congestion occurs. The network grants access to only one packet, while the rest wait. When contention persists, the buffers containing the blocked packets fill, and in lossless networks, which do not allow dropped packets, flow control prevents other switches from sending packets to the congested ports. Although flow control is essential to avoid dropping packets, it rapidly propagates congestion to other switches, because it will also block packets stored at some of their ports. Through these aggregation effects, congestion can spread progressively through the network, forming *congestion trees*, whose branches match the paths followed by data flows contributing to the congestion (hot flows).Packets belonging to hot flows prevent other packets from advancing, even those belonging to non-congested flows (cold flows). This phenomenon, known as head-of line (HOL) blocking, occurs when a packet at the head of a first - in, first - Out (FIFO) queue becomes blocked, prevents the remaining packets in the same queue from advancing. The impact of HOL blocking on network performance

Correspondence email address: **a.vishnupriya2010@gmail.com**

can be very serious: Network throughput can degrade and packet latency can increase dramatically. Network throughput not only diminishes severely (by approximately 70 per cent) when congestion appears, it also remains low well beyond the end of the hot-spot traffic generation. This degradation is the result of HOL blocking produced by the congestion tree affecting other (non-congested) packets. Latency increases from a few hundred nanoseconds (before congestion) to several hundred microseconds, an increase of three orders of magnitude. Packet latencies eventually return to the low values, but only after several milliseconds. This data also confirms that packets experience massive HOL blocking during and after the hot-spot injection. Once this kind of blocking has occurred, even under relatively benign traffic patterns, congestion trees can remain in the network and keeps it operating at reduced efficiency for extended periods of time. Taking all this into account, most designers today choose to reduce system cost and power consumption by reducing the number of network components. If we are to maintain a system's computational power, there are only two ways to reduce the number of network components. If we are to maintain a system's computational power, there are only two ways to reduce the number of network components: increasing the number of end nodes attached to each switch, or using a more suitable fabric topology. However, each of these solutions leads to a higher link use level, thereby driving the network closer to its saturation point and increasing the likelihood of congestion but only the Regional Explicit Congestion Notification (RECN) mechanism achieves both the required efficiency and the scalability that emerging systems demand.

II. APPROACHES TO CONGESTION MANAGEMENT

The problems associated with congestion are more difficult to solve in lossless networks than in lossy networks such as the Internet, because lossless networks cannot drop packets when congestion occurs. However, researchers have proposed many techniques, representing two basic approaches. The first approach, described in the "Classical congestion elimination or reduction" tries to eliminate the congestion either by preventing its, Appearance (proactive techniques) or by acting immediately upon detection of congestion (reactive techniques).However, in general, neither of these techniques can effectively eliminate congestion. Proactive congestion management requires knowledge of the network status and application requirements that is not always available. The larger the network, the greater the difficulty in knowing network status, and the more conservative the behavior of the mechanism would have to be. The second, and more recent, congestion management approach takes the view that congestion itself is not a problem. Techniques that follow this approach try to eliminate congestion's negative effects mainly HOL blocking rather than eliminating the congestion trees. The idea is that if the technique can completely eliminate HOL blocking, congestion can exist harmlessly. The complete elimination of HOL blocking has been possible, In general, the most effective HOL-blocking elimination techniques use separate queues to store packets belonging to different flows. Some techniques distinguish data flows on the basis of their final destination (HOL blocking elimination at the network level) and use this criterion for mapping packets to queues. For instance, if a network uses virtual output queues (VOQs) at the network level (VOQnet), each port has as many queues as end points in the network, and every incoming packet is stored in the queue assigned to its destination. This most techniques for eliminating network-level HOL blocking are very efficient, but they require heavy use of resources (queues) for networks with many end points. Thus, these techniques don't scale well; their implementation on large networks is very expensive in terms of silicon area. Other techniques try to solve this problem by eliminating HOL blocking at the switch level. In this case, both resource allocation and packet storing in a switch port depend on the switch output port that the packet requests. In this case, both resource allocation and packet storing in a switch port depend on the switch output port that the packet requests. For example, if a network uses VOQs at the switch level (VOQsw), each switch port has as many queues as output ports in the switch, and a packet is stored in the queue assigned to its switch output port. Therefore, the number of queues at each port depends on the number of switch output ports, but not on the number of network endpoints. This confirms that VOQsw only partially eliminates HOL blocking but this is scalable. On the other hand, VOQnet maximizes network throughput and minimizes packet latency. This situation changed recently, with the proposal of RECN. This technique for eliminating HOL blocking is fully efficient and scalable.

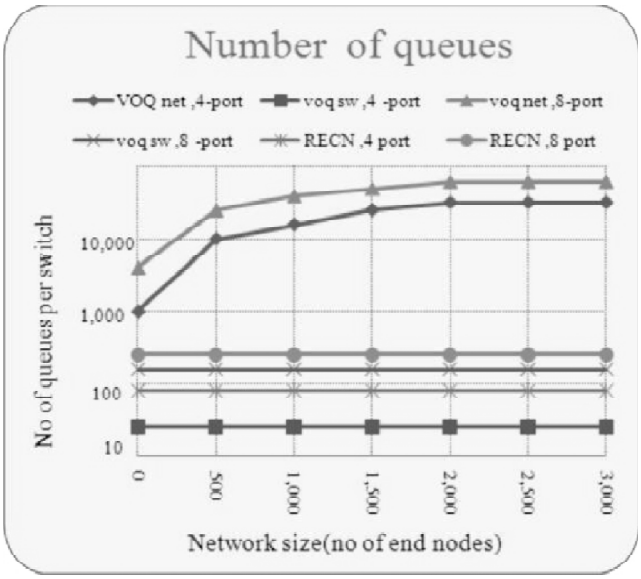

Figure 1: No of Queues per switch VOQsw, VOQnet, RECN

In RECN, the first key difference is that, to store congested flows separately, the previous techniques allocated queues statically, whereas RECN allocates and deallocates queues dynamically. Thus, RECN uses queues only as necessary. Another difference is that RECN can address congestion at any network point, not only endpoints. This lets RECN accurately identify congested flows and non-congested ones, so it can separate them. Finally, RECN recognizes that packets belonging to non-congested flows can be stored in the same queue without producing significant HOL blocking, even if they follow different routes. This allows a great reduction in the number of queues that the mechanism requires. Fig 1 shows a comparative study of number of queues required for each switch architecture.

(A) RECN Implementation

RECN is implemented in both input port (fig. 3) and output port (fig. 4) of the switching architecture. This strategies use queuing (buffering) of packets on both side i.e. on both sender and receiver. Figs 3 and 4 show, for the two sides, the main structures and events involved in the basic RECN procedure.

(1) Routing Requirements

Like other techniques for eliminating HOL blocking, RECN is based on storing the packets belonging to congestion trees in separate queues. More precisely RECN detects and Notifies the position of the roots of the different congestion Trees present in the network and later checks at any notified Port whether an incoming packet will pass through one of these roots separate queue. Therefore, RECN requires the ability to inspect packet routes in advance at any port. In addition, to achieve maximum accuracy and efficiency, RECN Must also detect congestion within the network, not only at the endpoint. Therefore, it requires a method of identifying any network port.

The use of source deterministic routing helps RECN meets both requirements. Specifically, RECN assumes the routing properties of Advanced Switching (AS), an open standard for fabric-interconnection technologies developed by the ASI. RECN use protocol interface -7 (PI-7) Format in Advanced Switching Interconnect (ASI is a switching interconnect technology, that combines the advanced features of existing proprietary fabrics with industry standard technology and design practices developed by the Advanced Switching Interconnect Special Interest Group (ASI SIG™). ASI is uniquely positioned within the industry to draft off the industries tremendous adoption of PCI e while providing the critical features required in demanding switch fabric architectures.) Technology. Fig. 2 shows the advanced switching packet header format PI-7. Packet headers in AS contain a 31 bit field that encodes all the turns that a packet must take along its route (a turn is a shift from a switch input port to an output port).This field is known as the *turnpool.* The header also includes a

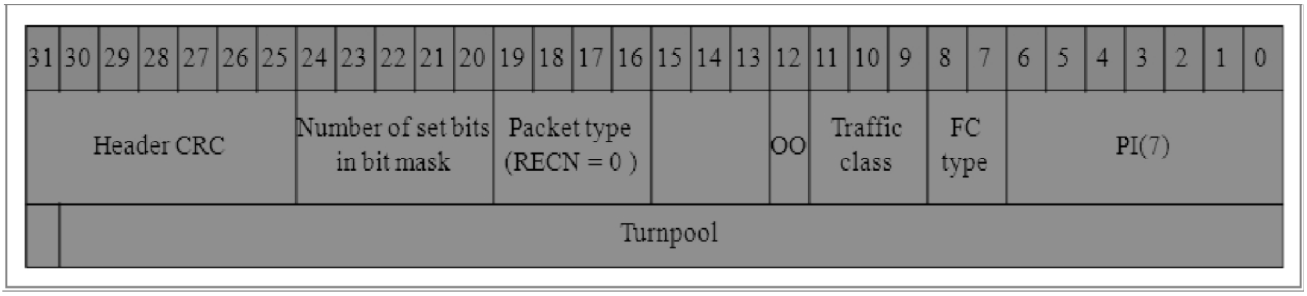

Figure 2: Advanced Switching PI -7 packet Header Format for RECN

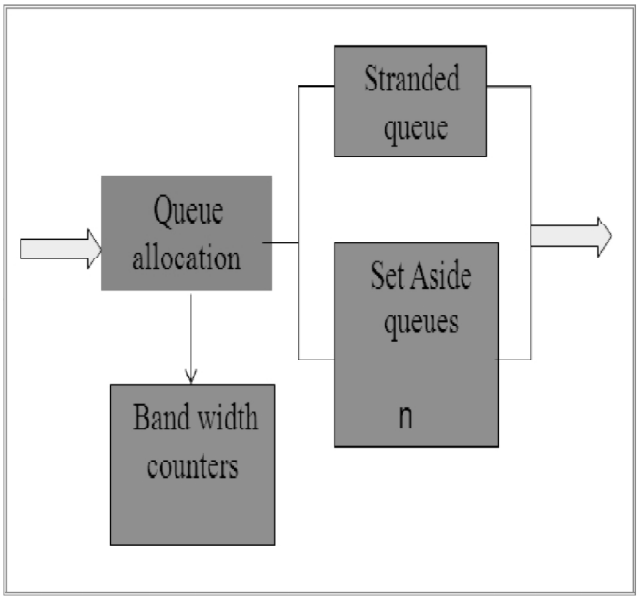

Figure 3: RECN Logical Input Port Organization

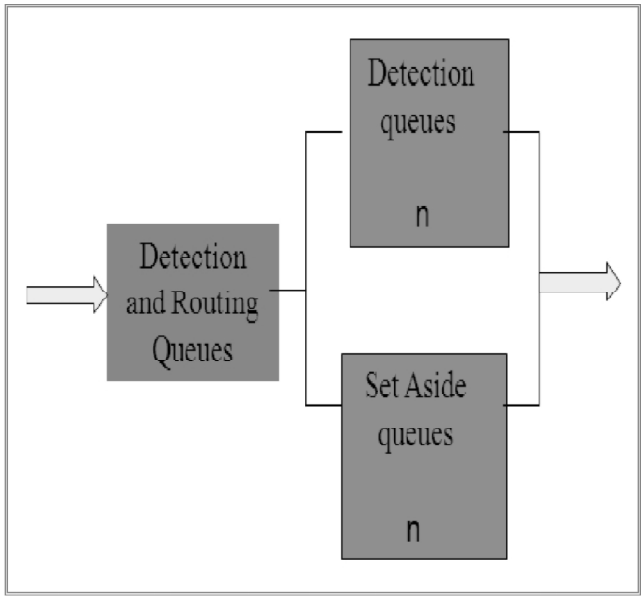

Figure 4: RECN Logical Output Port Organization

pointer (turn pointer) that indicates the next turn. This routing mechanism lets each of the different network ports address the root of a congestion tree using a specific sequence of turns. Moreover, the routing mechanism also lets RECN determine whether a packet in a port will pass through a congestion tree root simply by inspecting a sequence of turns from its turnpool.

(A) Required Resources

RECN requires additional resources to standard queues in both at input (fig. 3) and output (fig. 4) switch ports. In particular, besides the standard queue, at each port RECN uses a set of additional queues, called set-aside queues (SAQs). A port dynamically allocates a SAQ to store packets belonging to a specific congestion tree once the port detects, or is notified of, the congestion tree's existence. When congestion clears, the port deallocates the corresponding SAQ, making it available to store packets belonging to other congestion trees. Each switch port stores packets belonging to non-congested flows in its standard queue. Thus, the normal queue and the set of SAQs share the data RAM at each port. Although the number of SAQs per port could be varied, the dynamic allocation and deallocation arrangement means that fewer SAQs can efficiently handle congested situations. Moreover, the number of SAQs is independent of network size. In fact, RECN can eliminate HOL blocking in a network using just few numbers of SAQs per port. In short; [2] the RECN mechanism is fully scalable and efficient in congestion avoidance. Fig 1 compares the number of queues per switch required for implementing VOQnet, VOQsw, and RECN. RECN uses a content addressable memory (CAM) to manage the SAQs. Each CAM line contains the control information associated with a specific SAQ. Fig 5 shows CAM line used for RECN-CIOQ without dynamic deallocation schemes. This Information consists of a valid bit (V), indicating whether the SAQ is active; a turnpool and a bit mask, identifying the root of the congestion tree for which this SAQ is allocated; a blocked bit (B) for the *X*on/*X*off flow control (where *X*on and *X*off are the thresholds for resuming and stopping the incoming flow); and a ready bit (R), which indicates whether the SAQ may send packets to the link controller. Only 65 bits of control information are required per SAQ: 62 bits for the turnpool and bit mask, and the 3 control bits. SAQs (those with bit V active).

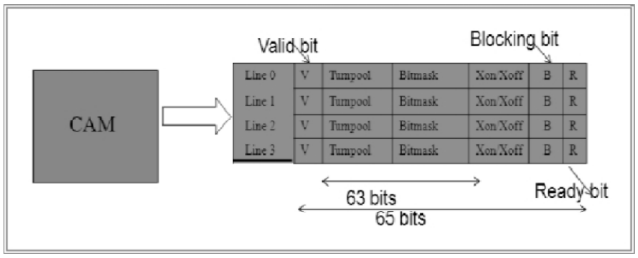

Figure 5: RECN CAM Organization for ASI Networks

If the packet turnpool matches several routes, the longest matching one is selected. When the packet's turnpool fails to match, the packet goes to the standard queue (actually, a detection queue). This turnpool comparison takes place in the same way at output ports (Fig. 4).

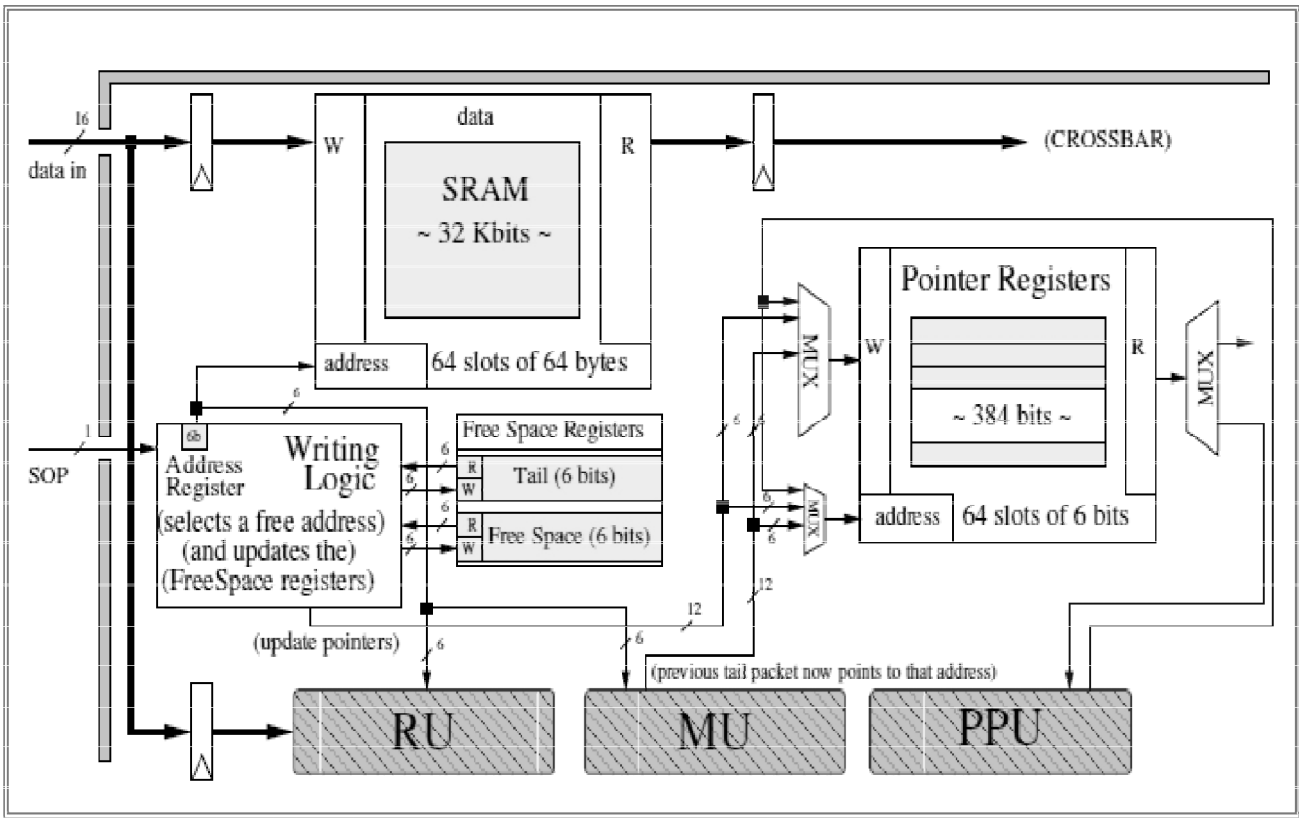

Figure 6: RECN Mechanism of Input Queued Switch Architecture

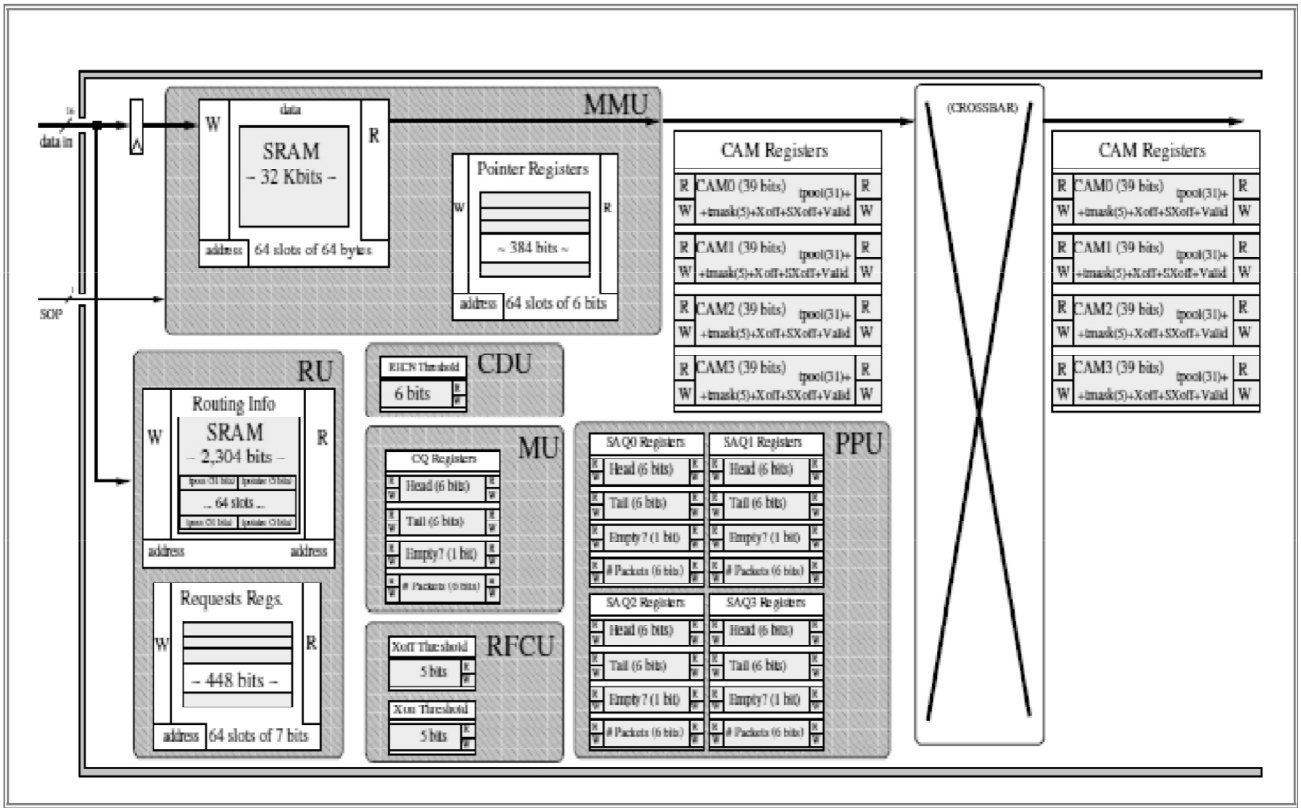

Figure 7: RECN Memory Mechanism at Switch Input Port

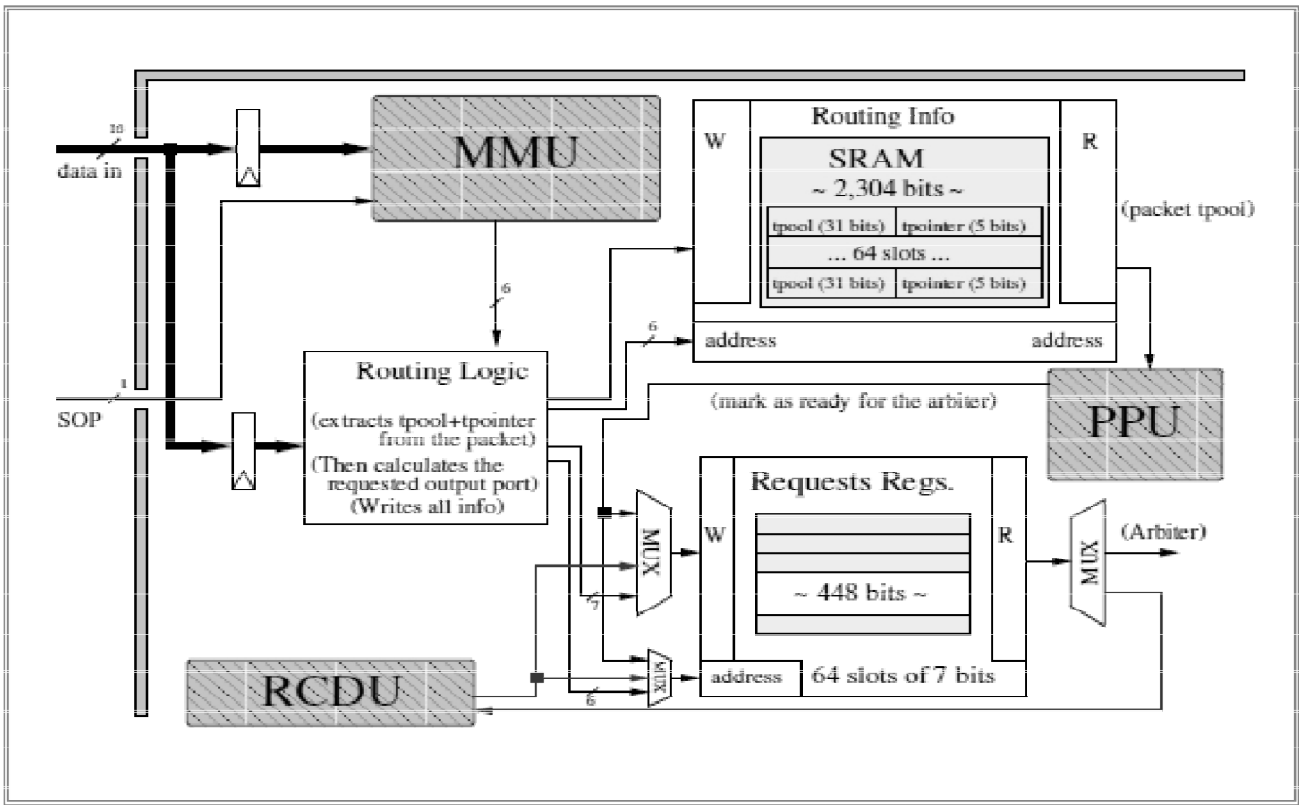

Figure 8: RECN Congestion Detection Mechanism

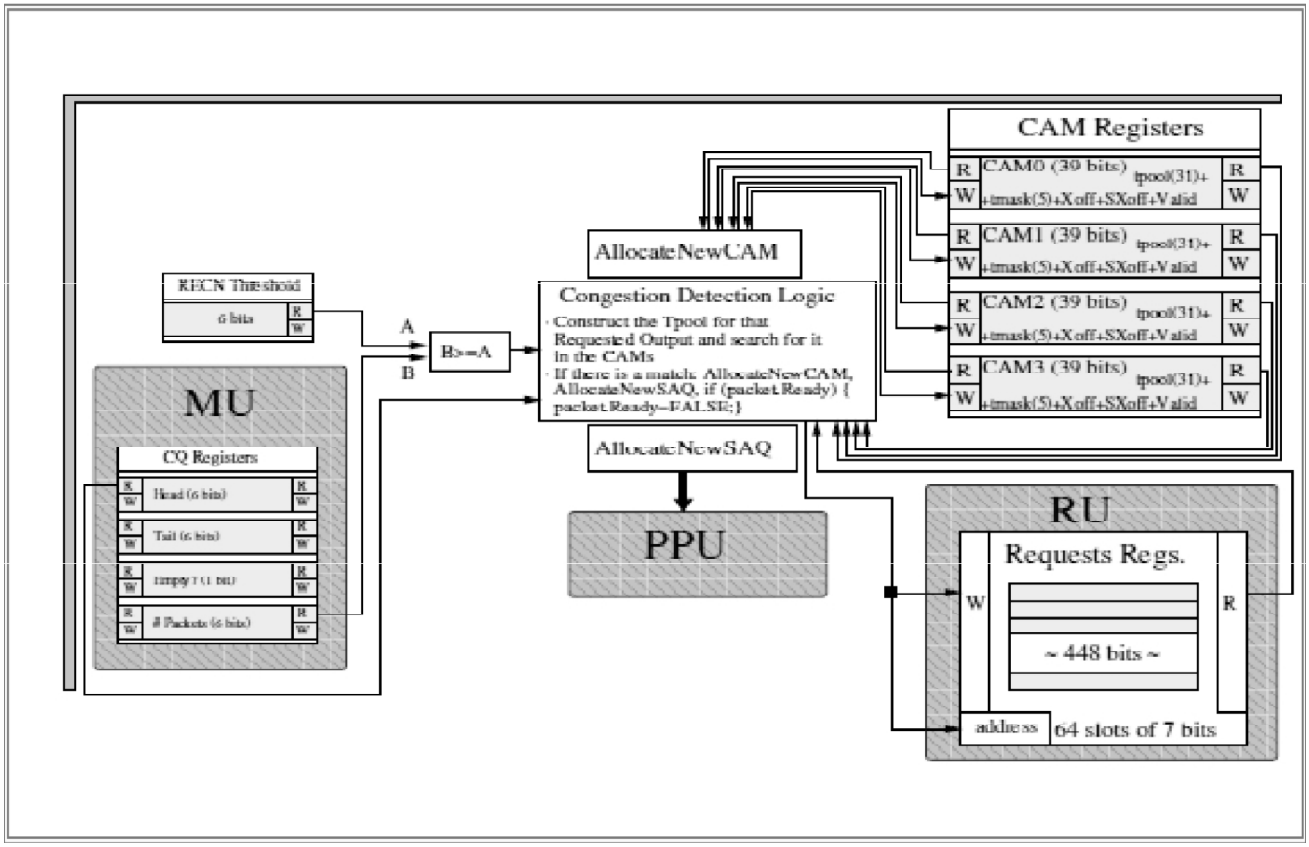

Figure 9: RECN Routing Unit

(1) Congestion Detection and Notification

At input ports (Fig. 3), the standard queue in the RECN mechanism is divided into several detection queues, each assigned to a specific output port. Each non-congested packet is stored in the detection queue associated with its requested output port. (Without the SAQs, the memory arrangement at the input ports would follow a virtual output queuing scheme at the switch level). However, RECN does not use the detection queues to eliminate HOL blocking introduced by congested packets. Instead, the SAQs perform whenever the occupancy of a detection queue reaches a certain input detection threshold that input port triggers a new SAQ allocation. Each such allocation involves the valid bit being set in the corresponding CAM line, and the turnpool and bit mask fields being filled to address the output port associated with the detection queue. Additionally, SAQ allocation resets the blocked bit (B). At the output port , whenever a new packet arrives and the standard queue occupancy is beyond a given output detection threshold, that port sends an input SAQ allocation request to the input port that sent the arriving packet. This internal notification includes the turnpool and bit mask that identifies the output port from the notified input port (fig. 3). This information goes into the CAM line associated with the input SAQ that the input port allocates upon receiving the notification. Whenever a SAQ at the input port (Fig. 3) reaches the *X*off threshold, it raises the sending of a *X*off flow control packet to the upstream switch. This control packet is sent through the adjacent output port (the output port connected to the same bidirectional link as the input port), and it contains the turnpool and the bit mask associated with the input SAQ. Similarly, once the *X*on threshold is reached in a SAQ, a *X*on flow control packet also containing the corresponding turnpool and bit mask is sent through the adjacent output port. Upon receiving the *X*off control packet, the output port at the upstream switch compares the received turnpool and bit mask to all the valid turnpools from the CAM. When the output port finds a match, it blocks the corresponding SAQ (by setting the B bit). When it does not find a match, it considers the *X*off control packet to be a congestion notification, and thus allocates a new SAQ for the network point indicated by the received turnpool and bit mask. Because RECN uses flow control messages to propagate congestion information between switches, the mechanism requires only one type of control message. Moreover, the information that must be included in this type of message fits into a certain control packet format defined in the Advanced Switching specifications. In terms of control packets, RECN is fully compatible with Advanced Switching. Along with other control information, the format of the notification packet includes all the information required for identifying the congested root associated with the SAQ whose occupancy has reached the *X*off threshold (taken from that SAQ's CAM line). Similarly, whenever a SAQ at the output port reaches the *X*off or *X*on threshold, it broadcasts an internal flow control notification to all the input ports of its switch. This notification includes the turnpool and bit mask associated with the output SAQ. Each input port updates this routing information to include a new turn: the one required for reaching the notifying output port from the notified input port. Each input port compares the modified turnpool and bit mask to all the valid turnpools in its CAM. (Fig. 5) An input port that finds a match handles the internal notification as a flow control notification, and sets the B bit of the corresponding CAM line accordingly. If the notification is a *X*off type and there is no match, the input port that sent the last packet to the notifying output port allocates a new SAQ for the network point encoded by the modified turnpool and bit mask. This newly allocated SAQ is blocked by setting its B bit.

(2) Resource Deallocation

When a congestion tree vanishes, the SAQ associated with it must be deallocated. Of course, it is important to avoid early deallocation, which could introduce HOL blocking. RECN requires certain conditions before it detects that congestion is no longer present in a port, and only then does it deallocate the corresponding SAQ. To allow distributed SAQ deallocation, RECN requires the following conditions, which can be ascertained only from local information:

- The SAQ is empty.
- The SAQ is not blocked by the ready (R) Bit at the CAM line. Thus, there are no link pointers to this SAQ on any other queue. This condition avoids an empty SAQ being deallocated just after its allocation while the congestion tree is present.
- The SAQ is not blocked by *X*off flow control.

The SAQ must be in the *X*on state to be deallocated. This condition avoids SAQ deallocation while the congestion tree is near (the corresponding

downstream SAQ occupancy is over the *X*off threshold). A SAQ's deallocation does not depend on other SAQs. As soon as a SAQ is not necessary (because the congestion tree associated with it has disappeared), RECN deallocates it. Immediately after its deallocation, the SAQ can be allocated for another congestion tree. Thus, distributed SAQ deallocation allows efficient use of network resources. The deallocation mechanism does not require any specific control message between SAQs maintaining RECN's control packet compatibility with advanced Switching. Congestion Detection Unit (CDU- Fig. 8), Routing Unit(RU – Fig 9), Mapping Unit(MU), Packet Processing unit (PPU), Flow Control Unit (FCU), Scheduler And Global flow Control unit. MMU is located inside the Input port. MMU is in charge of mapping all the incoming packets to the corresponding queue and to keep track of the allocated queues within memory. It has 64 registers to indicate which the next slot in queues order is. MU is used to keep track of cold queues, which is used to store the newly arrived packet to the input port. RU will decide which output port of switch has request according to the header value output port and packet header. CDU is in charge of detecting the congestion, computing the output port congested, and if required allocating the new SAQs for the congested point. PPU have the authority of separating the congested flows from the non-congested one, and it will separate every congested flow from each other. This PPU will move the packet from CQ to a SAQ. FCU will supervise Xon/Xoff (Stop & Go) flow control for the SAQs. This strategy also has the same efficient features like

- Congestion Detection
- Congestion notification
- SAQ allocation
- SAQ Deallocation
- Packet processing
- Flow control

(1) Congestion Detection, Notification and SAQ Allocation

With RECN-DQ, congestion detection is made in the same way as in the original RECN. However, the propagation of congestion detection differs significantly, as congestion notifications are now associated with the Xon/Xoff flow control. Moreover, SAQs for the same congestion tree are not linked (there is one SL field in the CAM, used to store the service level of the corresponding SAQ is assigned). Fig 10 shows CAM structure. For the case of Xon/Xoff flow control between two switches, whenever the occupancy of the input SAQ reaches the Xoff threshold, a Xoff control packet will be sent upstream.

(B) RECN DD-IQ

RECNDD-IQ improves the previous RECN version in several ways. Firstly, SAQs can be deallocated independently of other SAQs allocated for the same congestion tree. Secondly, explicit congestion notification packets are no longer needed as congestion notifications are now attached to flow control packets. Both leads to lower requirements of queues and control information. The RECN –IQ mechanism has been designed to have data memory only at input port of a switch. Fig. 6 shows the complete architecture of Input Queued switch with RECNDD.RECN_IQ have some different functional units Memory Management Unit (Fig 7- MMU), This packet includes the turnpool and the bit mask associated to the SAQ. The upstream output port will compare the received turnpool and bit mask to those associated to all the allocated SAQs at the output port. When matching, the corresponding SAQ will be set at Xoff state (a Xoff control packet has been received), activating the corresponding Xoff bit.

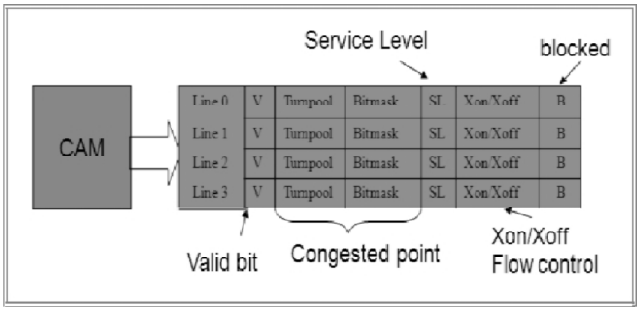

Figure 10: RECN CAM Organization

However, when failing to match, the switch will consider the Xoff control packet as a congestion notification. Thus, a new SAQ will be allocated for the received turnpool and bit mask. Furthermore, the new allocated SAQ will be set at Xoff state. For the case of Xon/Xoff flow control within a switch, whenever the occupancy of a SAQ at the output side reaches or is beyond the Xoff threshold, an internal Xoff notification is sent to all the input ports of the switch. This notification contains the turnpool and the

bit mask from the associated CAM line. The Xoff notification also includes the ID of the input port that sent the last packet to the SAQ. On every input port, upon reception of the internal Xoff notification, the turnpool and bit mask are compared to those associated to all the allocated SAQs. When matching, the corresponding SAQ is set at Xoff state (an internal Xoff notification has been received). If so, the input port must allocate a new SAQ for the congestion tree. Additionally, the newly allocated SAQ will be set at Xoff state.

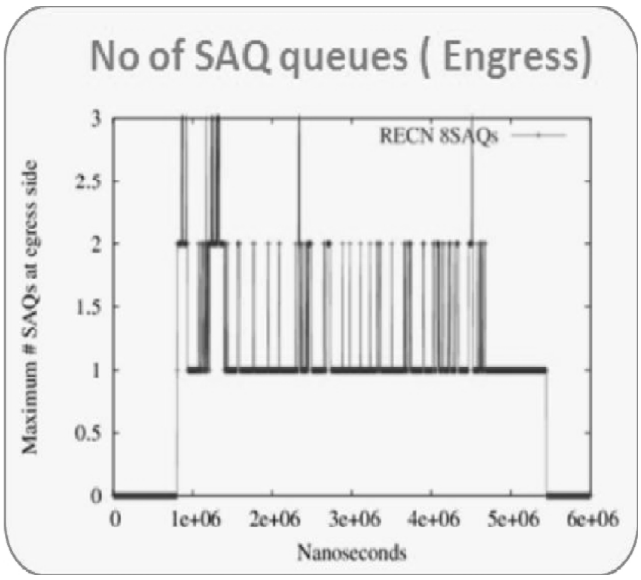

Figure 11: Number of SAQs Queues Needed at Egress of RECN Switch Mechanism

Note that Xon control packets (and Xon internal notifications) also contain a turnpool and a bit mask. This is needed in order to identify which SAQ must be set at Xon state.

(2) SAQ Deallocation Mechanism

With RECN-DQ, SAQs can be deallocated in a distributed way. Thus, the basic deallocation requirement is that the SAQ is empty. However, a new safety condition must be met in order to deallocate SAQs properly: the SAQ is neither blocked due to the blocking bit nor at Xoff state. Note that, since a SAQ is at Xoff state at the moment it is allocated, this condition avoids an empty SAQ being deallocated just after its allocation (premature deallocation). Thus, the timer is no longer needed. Moreover, as the SAQ must be at Xon state to be deallocated, SAQs are not deallocated while congestion is near (a SAQ at Xoff state means that the downstream SAQ is full of packets).

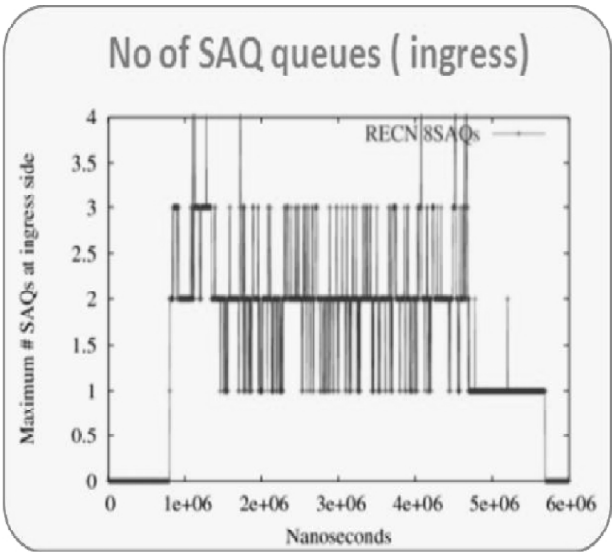

Figure 12: Number of SAQs Queues Needed at Ingress of RECN Switch Mechanism

Note that all of these conditions can be evaluated completely from local information, and all of them are independent of the state of other SAQs, with the only exception of flow control. Notice also that the external data required for evaluating the new conditions are just those contained in the Xon/Xoff flow control packets. Thus, RECN-DQ does not require specific RECN messages (detection notifications, tokens, etc.). As the Xon/Xoff flow control is considered in the AS specification, RECN-DQ does not require any special control message that is not compatible with AS.

III. SIMULATION RESULTS

In this section, the RECN-DQ mechanism is evaluated by means of simulation results. Specifically, we show the network throughput achieved when the injected traffic is varied. Also, results of switch efficiency as a function of time are presented. These metrics have been measured for different values of the maximum number of SAQs available at input ports. Specifically, we have considered 2, 4 and 8 SAQs for simulating different RECN-DQ configurations and also no SAQs for simulating switches not using RECN-DQ. Regarding network size and topology, the following two Bidirectional Multi-stage Interconnection Network (BMIN) configurations have been analyzed:

- Configuration 1: 64 end nodes, BMIN made of 8x8 switches (48 switches in 3 stages, perfect shuffle as interconnection pattern).

- Configuration 2: 256 end nodes, BMIN made of 8x8 switches (256 switches in 4 stages, perfect shuffle as interconnection pattern).

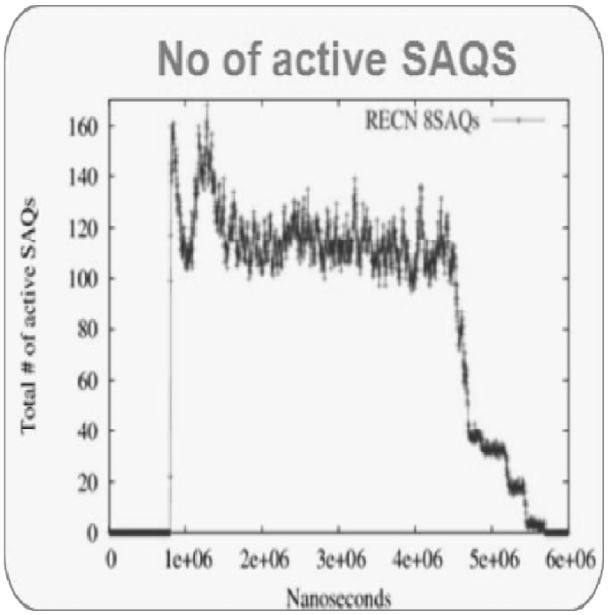

Figure 13: Number of SAQs Queues is Active Results for Uniform Traffic

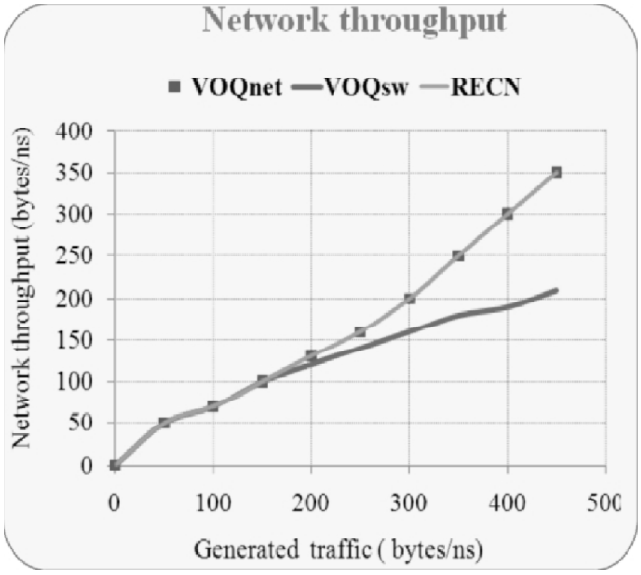

Figure 14: Network Throughput of 64*64 Networks, with Comparison of Different Switch Architecture

Other common parameters used in all the simulations are:

- *Input Memories Size=4KB (64 packets);*
- *Packet Length=64 bytes;*
- *Xon Threshold =5 packets;*
- *Xoff Threshold = 10 packets;*
- *Congestion detection Threshold= 5 packets;*

For the two BMIN configurations considered, throughput and latency results obtained using uniform traffic and different numbers of SAQs per port can be seen in Fig. 11 in egress point, Fig. 12 as in ingress point & fig. 13 as number of active SAQs. In the case of the 64-end node MIN, the maximum efficiency for an IQ switch without RECN-DD (no SAQs used) is about 65%. When using RECN-IQ and only 2 SAQs, the efficiency increases to above 80%. However, as can be seen in the figures, at least 4 SAQs are needed for achieving maximum switch efficiency.

The switch efficiency for the 256-end node MIN is slightly lower than the one for the 64-end node MIN, unless we use RECN-IQ and 8 SAQs. In this case, the switch efficiency is maximum. When using only 4 SAQs, the efficiency is above 90%. When using RECN-IQ with only 2 SAQs, switch

IV. CONCLUSION

This paper concludes a new RECN version, referred to as RECNDD-IQ. RECN dynamically separates congested packets from noncongested ones, thus eliminating the HOL blocking.

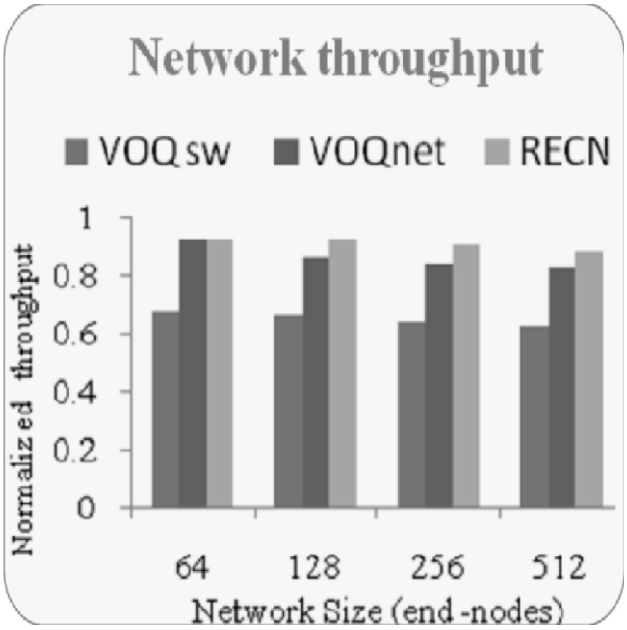

Figure 15: Network Throughput of Various Size Networks, with Comparison of Different Switch Architecture

This is achieved by dynamically allocating and deallocating SAQs (Set Aside Queues) for congested packets at switches.

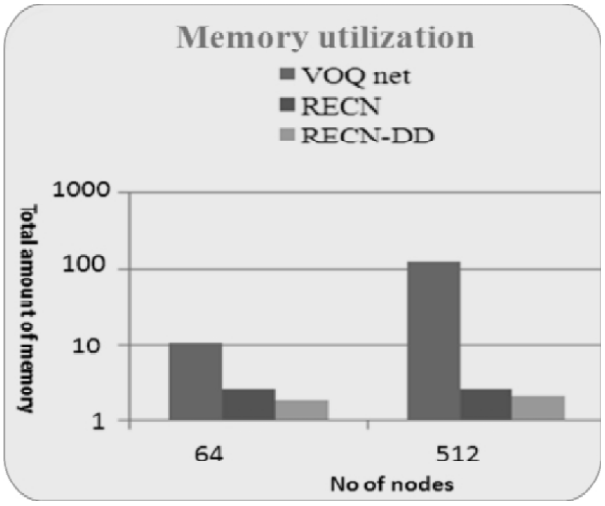

Figure 16: Memory Utilization for Various Sizes of Networks, with Comparison of Different Switch Architecture

This version introduces a new, completely distributed deallocation SAQ mechanism that does not impose any constraint on the order in which queues are deallocated. From the performance we can conclude that RECN and RECN-IQ require much less memory area at each port than VOQnet, but they also exhibit an excellent scalability. Note that VOQnet is not scalable at all. Furthermore, RECNDD-IQ reduces significantly the memory area required by RECN at each port, thus reducing the overall switch cost. Fig 14, 15, 16, shows the performance of RECN with compared to others.

References

[1] Advanced Switching Core Architecture Specification. Available at *http://www.asi-sig.org/specifications for ASI* SIG.

[2] T. Anderson, S. Owicki, J. Saxe, and C. Thacker. High-speed Switch Scheduling for Localarea Networks. *ACM Transactions on Computer Systems*, **11**(4): 319–352, 1993.

[3] Alejandro Martý´nez, Pedro J. Garcý´a J.Alfaro, Jose´ L.Sa´nchez, Jose´ Flich, Francisco J. Quiles, a Jose´Diatom "A Switch Architecture Guaranteeing QoS Provision and HOL Blocking Elimination "*Parallel and Distributed Systems*, **20**(1), 2009.

[4] W. J. Dally, P. Carvey, and L. Dennison. The vici Terabit switch/router. Proc. Hot Interconnects 6, 1998.

[5] J. Duato, I. Johnson, J. Flich F. Naven, P. J. Garcia, and T. Nachiondo. A New Scalable and Cost-effective Congestio Management Strategy for Lossless Multistage Interconnection Networks. Proc. 11th International Symposium on High Performance Computer Architecture (HPCA05), 108–119, 2005.

[6] P. J. Garcia, J. Flich, J. Duato, I. Johnson, F. Quiles, and F. Naven. Efficient Scalable Congestion Management for Interconnection Networks. *IEEE Micro*, **26**(5): 52–66, 2006.

[7] P. J. Garcia *et al.*, "Dynamic Evolution of Congestion Trees: Analysis and Impact on Switch Architecture," Proc. Int'l Conf. High Performance Embedded Architectures & Compilers (HiPEAC 2005), LNCS 3793, Springer, 2005, pp. 266-285.

[8] InfiniBand Architecture Specification Volume 1, Release 1.0, InfiniBand Trade Assoc., Oct. 2000.

[9] M. Karol, M. Hluchyj, and S. Morgen. Input versus Output Queueing on a Space Division Switch. *IEEE Transactions on Communications*, **35**(12): 1347–1356, 1987.

[10] Myrinet, Myricom Inc., *http://www. myrinet.com*

[11] QsNet Overview, white paper, quadrics Ltd., 2005, *http://www.quadrics.com*