RECN-IQ: A Cost-Effective Input-Queued Switch Architecture with Congestion Management*

Gaspar Mora† Pedro J. Garcia‡ José Flich† José Duato†
†Dept. of Computer Engineering
Universidad Politécnica de Valencia, Spain
{gmora, jflich, jduato}@gap.upv.es
‡ Dept. of Computing Systems
Universidad de Castilla-La Mancha, Spain
pgarcia@dsi.uclm.es

Abstract

As the number of computing and storage nodes keeps increasing, the interconnection network is becoming a key element of many computing and communication systems, where the overall performance directly depends on network performance. This performance may dramatically drop during congestion situations. Although congestion may be avoided by overdimensioning the network, the current trend is to reduce overall cost and power consumption by reducing the number of network components. Thus, the network will be prone to congestion, thereby becoming mandatory the use of congestion management techniques.

In that sense, the technique known as Regional Explicit Congestion Notification (RECN) completely eliminates the Head-of-Line (HOL) blocking produced by congested packets, turning congestion harmless. However, RECN has been designed for switches with queues at input and output ports (CIOQ switches), thus it can not be directly applied to other types of switches. Additionally, the method RECN uses for detecting congestion requires several detection queues that increase the memory requirements and thus switch cost.

Thus, we completely redefine the RECN mechanism in order to achieve different goals. First, we adapt RECN to a switch organization with queues only at input ports (IQ switches). These switches are simpler and cheaper to produce than CIOQ ones. Second, we propose a new method for detecting congestion that does not require several detection queues, thereby reducing RECN memory requirements.

These improvements lead to achieve a cost-effective switch organization that derive maximum performance even in the presence of congestion. Also, we present in detail a realistic switch architecture supporting the new mechanism.

Results demonstrate that the new RECN version in an IQ switch achieves maximum network performance in all the analyzed situations. These results have been obtained with a reduction factor of data memory requirements of 5 with respect to the previous RECN mechanism in CIOQ switches.

1. Introduction

High-performance interconnection networks are nowadays present in a wide variety of computing and communication systems: massive parallel processors, local and system area networks, clusters of PCs and workstations, IP routers, and, recently, inside the chips (Networks on Chip). In such environments, as the number of processing and storage nodes increases, the interconnection network plays a prominent role in the performance achieved by the whole system.

One of the main concerns interconnect designers faced during the last years has been network congestion. Congestion occurs when several flows of packets simultaneously and persistently request the access to the same network resources (typically, a switch output port). In these cases, any packet not granted will block, and will remain stored[1] in a queue until its request is attended. This may cause the appearance of the phenomenon known as Head-Of-Line (HOL) blocking, that occurs when a packet at the head of a queue blocks, preventing the rest of packets in the same queue from advancing, even if they request available resources. When congested packets block and produce HOL blocking to non-congested ones (those packets belonging to flows that do not contribute to congestion), non-congested flows advance at the same speed as congested flows, thereby severely degrading network performance and eventually collapsing the network (the effect is rapidly spread over the entire network).

Although the negative consequences of congestion have been always evident, congestion has not been considered a critical problem until recently, due to several reasons. For

*This work was supported by Spanish MEC under Grant TIN2006-15516-C04, by Junta de Comunidades de Castilla-La Mancha under Grant PBC-05-005 and by CONSOLIDER-INGENIO 2010 under Grant CSD2006-00046.

[1] We assume lossless networks, where blocked packets are never discarded. Note that most current interconnects (Myrinet 2000, Quadrics, InfiniBand, Advanced Switching, etc.) are lossless.

2007 International Conference on Parallel Processing (ICPP 2007)
0-7695-2933-X/07 $25.00 © 2007 IEEE

instance, networks were traditionally overdimensioned (using more network components than strictly needed), and this leads to a very low link utilization, thereby reducing contention and congestion probability. Moreover, different queue organizations at switches reduce or eliminate the HOL blocking effect. This is the case of VOQ where a queue is used per each output port of the switch (VOQ at the switch level)[2] or per each possible destination end node (VOQ at the network level)[3]. Also, the traditional architecture for switches in communication networks used queues only at their output ports (Output Queuing, OQ switches), so HOL blocking (at switch level) was eliminated in this case.

On the one hand, the high cost and power consumption of current network components discourages to overdimension the network. In fact, it is more appropriate to use a lower number of network components for interconnecting the terminals, thus reducing cost and power consumption. However, link utilization increases and subsequently congestion probability. On the other hand, the OQ scheme has become unfeasible because it requires the memories to operate at a much faster speed than the links, and link speed in current high-speed interconnects is on the order of Gbps. Thus, most current switches use either queues only at input ports (Input Queuing, IQ switches), or queues at both input and output ports (Combined Input and Output Queuing, CIOQ switches)[2]. However, IQ and CIOQ switches may be affected by HOL blocking. In fact, this problem may limit the throughput of the switch to about 58% of its peak value [6]. Although VOQ schemes may be used, they become very expensive as the memory is the component that drives the final cost of the switch.

Taking all this into account, the use of an efficient congestion management technique is becoming mandatory in modern interconnects, in order to keep network performance at maximum levels while using a reduced number of network components. Although many techniques have been proposed in this sense, none of them has been completely satisfactory until the proposal of Regional Explicit Congestion Notification (RECN) [4, 5]. RECN focuses on detecting and eliminating the HOL blocking produced by congested packets. In order to achieve this, RECN identifies congested packets and stores them in special, dynamically-assigned set aside queues (SAQs). RECN completely eliminates HOL blocking while requiring a small number of resources (queues) per port. In fact, RECN is the first truly efficient and scalable HOL blocking elimination technique.

However, RECN has been proposed and designed assuming CIOQ switches. This narrows the scope of RECN, whose application is limited to switches with this type of switch organization, while IQ switches are currently preferred since their implementation is usually simpler and cheaper. Note that the cost of a switch is mainly driven by the memory used, and most of the switch power and area is consumed by the memory. This fact suggests that switch designs with fewer and smaller memories are preferable, thus the IQ organization being more attractive than CIOQ. Taking all this into account, one of the novelties we present in this paper is a new RECN version whose main aim is to adapt to IQ switches, thereby allowing any switch model with this organization to get efficient and scalable congestion management. The new version will be referred to as RECN-IQ. In this sense, RECN will be compatible with both CIOQ and IQ architectures, on which most of the current switches are based.

Additionally, we have reduced the memory requirements of RECN at the input ports of a switch. Specifically, the previous RECN version required several detection queues (one per output port in the switch) in order to detect congestion. In detail, whenever a detection queue fills over a given threshold, the output port associated to the detection queue is considered as a congested point. Although this method works accurately, it is expensive as it increases the minimum silicon area required at input ports (even if memory is dynamically managed). In order to avoid all the aforementioned problems, the new RECN version includes a new method for detecting congestion at input ports that requires a single detection queue per input port.

Note that, since the new RECN proposal removes data memories at output ports and detection queues at input ports, the power and area required for switches will be significantly reduced, thereby allowing us to build cheaper networks as switches will have smaller memories. Of course, this will lead to a reduction in the cost and power consumption of the overall system. For instance, in a 16-port switch, the previous RECN mechanism would require 256 detection queues in the switch, whereas the new RECN version will require only 16 queues (a reduction factor of 16).

Also, in this paper, we propose an efficient and realistic switch architecture suitable for the new RECN mechanism, describing in detail the structure and behavior of each functional unit of this architecture. All the memory sizes, signals and components required for a real implementation are defined. Note that this is the first time a switch architecture implementing RECN is described at this level.

Summing up, the main contributions of this proposal are the following:

- It adapts the RECN strategy to IQ switches, thereby allowing efficient congestion management in this type of switches.
- It is based on a perfectly defined, efficient and feasible switch architecture that favours a real implementation.
- It requires fewer resources at input ports than former RECN versions, thereby being an even more cost-effective technique.

[2] Although other schemes, like BC (Buffer Crossbar) switches, have been proposed, they are not so popular.

2007 International Conference on Parallel Processing (ICPP 2007)
0-7695-2933-X/07 $25.00 © 2007 IEEE

The rest of the paper is organized as follows. Section 2 describes the basics of the new proposal (RECN-IQ). Next, in Section 3, the IQ switch architecture with the new RECN-IQ mechanism is described. In Section 4 the performance of the new proposal is evaluated. Finally, in Section 5, some conclusions are drawn.

2. RECN-IQ: Functional Description

In this Section we will describe the new version of RECN for IQ switches. For this, we will describe the main modifications introduced in the previous RECN mechanism [4, 5] (suitable for CIOQ switches).

The switch architecture assumes a routing scheme similar to the one used in AS interconnects [1]. Thus, it is assumed that each packet includes in its packet header a turnpool and a turn pointer. The turnpool is made of 31 bits and contains a turn (offset from an input port to the requested output port) for each switch along the packet's path. The turn pointer, made of 5 bits, points to the set of bits of the turnpool encoding the turn for the current switch. Also, the switch architecture assumes ViCT switching.

For the sake of an easier understanding, the following subsections describe in detail and separately each different functional aspect of RECN-IQ: memory management, congestion detection, allocation of queues, processing of packets, and flow control. Later, when describing the switch architecture including RECN-IQ, we will implement each aspect in a separate functional unit.

2.1. Memory Management and Requirements

The RECN-IQ mechanism has been designed assuming data memories only at the input ports of a switch. At each input port a data memory will be used for allocating a normal (Cold) queue and a set of SAQs. For controlling the SAQs, an associated CAM (Content-Addressable Memory) will be required at each input port. Additionally, a CAM structure will be required also at output ports.

In the previous RECN mechanism, some detection queues were used at each input port. However, in the RECN-IQ mechanism only a single queue, the Cold Queue (CQ), will be used for storing all the incoming packets. Therefore, the new RECN mechanism will require at each switch fewer memory resources, both at the output ports (where only the CAM is required) and at the input ports (where detection queues are replaced by only one queue).

As an example, assuming a detection threshold of 2 slots (a slot will store one packet) and an Xoff threshold (required for activating the "stop" function in the flow control between SAQs) of 2 slots for SAQs, the RECN mechanism would require $N \times (2N + 8) + N \times (2 + 8)$ slots in a $N \times N$ switch with 4 SAQs per port. That is, at each input port (N) a memory with $2N + 8$ slots (N detection queues and 4 SAQs) is required, and at each output port (N) a memory with $2 + 8$ slots (1 standard queue and 4 SAQs) is required. All this sums 800 slots for a 16×16 switch. However, with the RECN-IQ mechanism, in a $N \times N$ switch with 4 SAQs per input port, the memory requirements are only 10 slots per input port. Since only memories are located at input ports, memory requirements are $10N$ slots. Thus, for a 16×16 switch, RECN-IQ would require 160 slots. A reduction factor of 5. Notice that these numbers have been obtained assuming short cables where Round-Trip-Time (RTT) is lower than a packet/slot.

2.2. Congestion Detection

The new RECN-IQ mechanism detects congestion only at input ports. In particular, whenever the number of packets in the CQ queue exceeds a given threshold (the *RECN-congestion* threshold), congestion is detected. Once a congested situation is detected, the congested point must be identified.

The RECN-IQ mechanism assumes that the origin of congestion is the output port requested by the first packet at the CQ queue. This assumption is based on the fact that, in a congestion situation, it is very likely that the first packet in a congested queue is blocked because it requests a congested output port. Therefore, in these situations, the detection mechanism will hit the congested output port.

Indeed, in a non-congested situation, the rate at which packets arrive to a given input port will be same at which packets leave the input port, thus the queue's occupancy will be low. Thus, if the CQ increases in size is because the packet at the head is blocked.

On the other hand, it may happen that the packet at the head of the queue is not being addressed to a congested output port. In that situation, the detection mechanism will fail, but the Post-Processing unit and the deallocation policy of SAQs will minimize the impact of the false detections (as explained below).

2.2.1 SAQ Allocation and Deallocation

Once congestion is detected, the input port allocates a new SAQ for the congested point (output port requested by the packet at the head of the CQ).

The associated CAM line will include all the routing information (turnpool + bit mask) and the status (Valid, Xoff, and SentXoff bits) of the SAQ. An active Valid bit indicates that the SAQ is assigned to a congestion point; an active Xoff bit indicates that the SAQ is stopped by the flow control between SAQs, and an active SentXoff bit indicates that the SAQ sent an Xoff signal to an upstream SAQ. In this case, when a new SAQ is allocated, it will set the "Valid" bit, and will reset the "Xoff" and "SentXoff" bits. As in

2007 International Conference on Parallel Processing (ICPP 2007)
0-7695-2933-X/07 $25.00 © 2007 IEEE

the previous mechanism, if there exists already in the port an associated SAQ for the detected congested point, then no new SAQ will be allocated. Thus, before allocating a new SAQ, a search in the CAM structure is performed.

Whenever a SAQ empties and is not blocked by the Xon/Xoff flow control (Xoff bit not set), then the SAQ is deallocated. So the RECN-IQ mechanism allows the deallocation of SAQs in a distributed way as it was done in the previous RECN mechanism.

2.3. Packet Processing

Whenever a new packet arrives to the switch, it is stored in the CQ queue, regardless whether it will pass through a detected congested point or not. A mechanism (Post-Processing mechanism) at each input port will decide later if the packet goes to any of the different SAQs allocated at this port. The Post-Processing mechanism cyclically inspects all the queues (CQ and SAQs), one at a time. At each queue, it will process the packet at the head of the queue. In particular, the routing information of that packet will be compared to the routing information of each CAM line. Thus, it will be detected whether or not the packet is going through a congested point in the network. In the case of a match, the packet will be moved to the corresponding SAQ. Otherwise, the packet will be set as ready for the switch arbiter. In this way, congested packets will not be blocked at the head of the queue, thereby avoiding HOL blocking.

Notice that the routing info of a packet may match at the same time the routing info of several active CAM lines in an input port. Thus, different situations may arise depending on the queue where the packet is initially mapped into. The first situation occurs when the packet is initially in the CQ and there are two matches. In order to avoid cycles among different SAQs when mapping packets, the shortest match will be selected. Thus, in a first post-processing step, the packet will be stored in the SAQ with the "less-specific" (shortest) associated turnpool. In a second situation, a packet stored in a SAQ matches (when it is post-processed) two CAM lines, one of them being less-specific than the one associated to the current SAQ. In that situation, the post-processing mechanism will select the less-specific match, but larger than the routing info for the current SAQ. This is done to preserve in-order delivery of the packets. To sum up, packets move initially from CQ to one SAQ, then from that SAQ to another in increasing matching size, until the packet reaches the SAQ with the largest match. Once a packet is post-processed and there is no match with any CAM line, then the packet is set as ready for the arbiter.

2.4. Congestion Information Propagation

The detection of congestion propagates between switches in the following manner. Whenever the number of packets on a SAQ exceeds the Xoff RECN Threshold, then the input port sends backwards an Xoff signal (containing the routing information that points to the associated congested point) to the corresponding output port at the upstream switch. Upon reception, a new CAM line is allocated at the output port of the switch pointing to the congested point (if there is not already a SAQ allocated for the notified congested point). The Valid and Xoff bits are set.

Whenever a packet passes through the output port[3], the routing information is inspected and compared to the routing information stored in the active CAM lines at this output port. If there is a match and the matching CAM line has its Xoff bit set, an internal Xoff notification is sent to the input port that sent the packet. Upon reception of that Xoff signal, the input port allocates a new SAQ+CAM line, with routing information pointing to the congested point (note that the routing information from the output port CAM line is updated at the input port CAM line for including another significant turn). The Xoff bit for that newly allocated SAQ is set. However, if a SAQ already existed for the congested point, then the Xoff bit of its associated CAM line is set.

2.5. Flow Control

A SAQ with its associated CAM line having the Xoff bit set cannot send packets through the switch. Whenever the number of packets on a SAQ goes below the RECN Xon threshold, an Xon signal is sent backwards to the connected output port at the upstream switch. Then, the corresponding CAM line at the output port resets the Xoff bit and broadcasts an internal Xon signal to all the input ports of the switch. The input ports with an allocated SAQ for the congested point reset the Xoff bit on the corresponding CAM line, thus allowing the packets stored in the SAQ to move forward.

2.6. RECN-IQ: Procedure Example

In order to introduce the basics of RECN-IQ, a graphical example showing the basic procedure is shown in Figure 1. Step 1 shows how congestion is detected. The link connected to output port P5 of Switch 2 is oversubscribed, therefore packets begin to accumulate on the Cold Queues (CQs) at several input ports of Switch 2. When the number of packets stored in the CQ in P2 of Switch 2 exceeds the RECN-IQ threshold, congestion is detected at the output port requested by the packet at the head of the CQ. Thus, output port P5 of Switch 2 is considered a congested point (congested point Z). Then, a new SAQ+CAM line is allocated at P2 of Switch 2, pointing to the detected congested point (as can be seen in Step 2 of Figure 1). Congested packets at the head of the CQ in P2 of Switch 2 are moved to

[3]Notice that queues are not implemented at output ports.

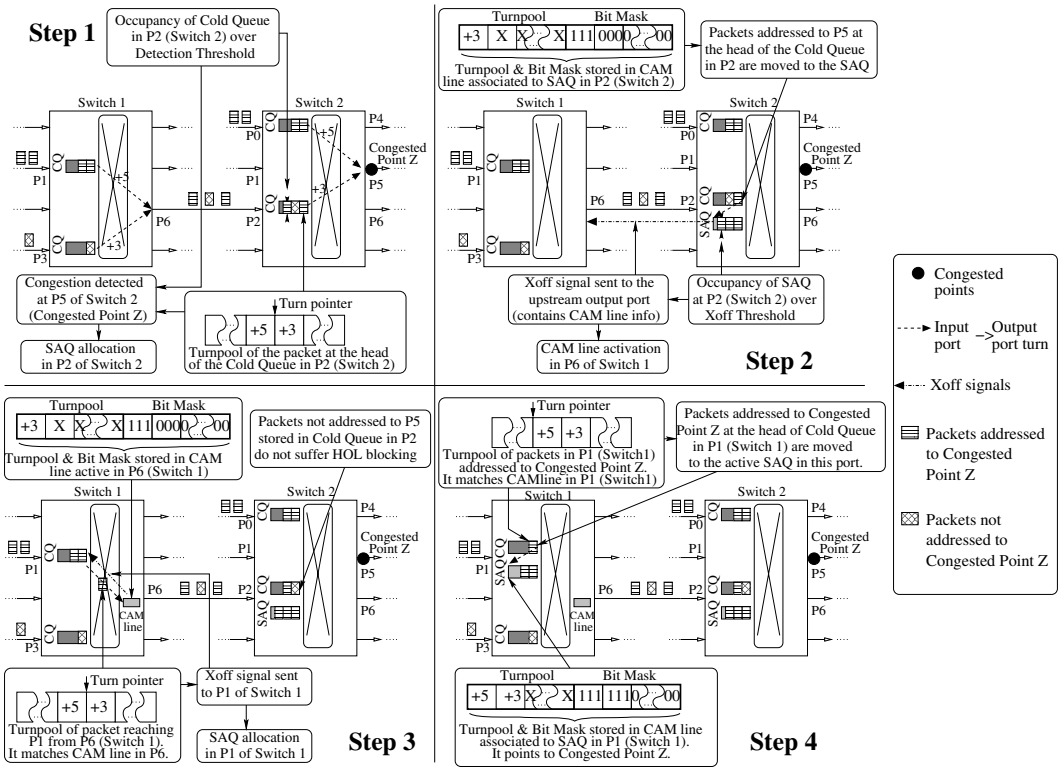


Figure 1. Example of RECN-IQ basic procedure.

the SAQ by the Post-Processing (PP) mechanism, so avoiding the HOL blocking these packets may produce. When the number of packets at the SAQ exceeds the Xoff threshold, an Xoff signal is sent backwards to the upstream output port (P6 in Switch 1). As shown in Step 3 of Figure 1, upon reception of the Xoff signal, a new CAM line pointing to the congested point is activated at P6 in Switch 1. The contents of this CAM line are copied from the downstream CAM line. When any input port of Switch 1 sends a packet through P6, the turnpool of the packet is compared to the turnpool stored in the CAM line. In the case of a match, the output port sends an Xoff signal to the input port (in the example, P1 of Switch 1). Upon reception of such signal, the input port allocates a new SAQ+CAM line pointing to the detected congested point Z, as can be seen in Step 4 of Figure 1. After this, the congested packets at the head of the CQ in P1 of Switch 1 are moved to the SAQ by the PP mechanism. If the number of packets at this SAQ exceeds the Xoff threshold, a Xoff signal would be sent backwards, and so on.

3. Input Queued Switch Architecture with RECN-IQ

In the previous Section we have described the new RECN-IQ mechanism. Of course, this mechanism would influence many aspects (i.e. memory management, flow control issues, scheduler) of any switch supporting it. Therefore, we detail in this Section the entire switch organization required for implementing the RECN-IQ mechanism.

Figure 2 shows a general overview of the switch architecture with RECN-IQ, depicting an input port, the crossbar and an output port. For the sake of an easier understanding, the switch has been divided into six functional units: Memory management (MMU), Mapping (MU), Post-Processing (PPU), Routing (RU), RECN Flow Control (RFCU), and Congestion Detection (CDU). In order to focus on the RECN-IQ mechanism and to avoid introducing graphical complexity, we do not include the arbiter in this Figure. Further in this Section we will discuss about the arbiter.

Also, the memory structures required by the mechanism and the switch architecture are depicted in Figure 2. In addition to the SRAM memory, an additional memory of 2 Kb is required at each input port. This memory will store the routing headers of each packet. Also, two register files will be required: the Pointer Registers File (PRF) and the Requests Registers File (RRF). The PRF will store the pointers among different packets, in order to keep the logical structure of the queues. The RRF will store the requested output ports of the packets.

Additionally, besides some small registers, for each im-

2007 International Conference on Parallel Processing (ICPP 2007)
0-7695-2933-X/07 $25.00 © 2007 IEEE

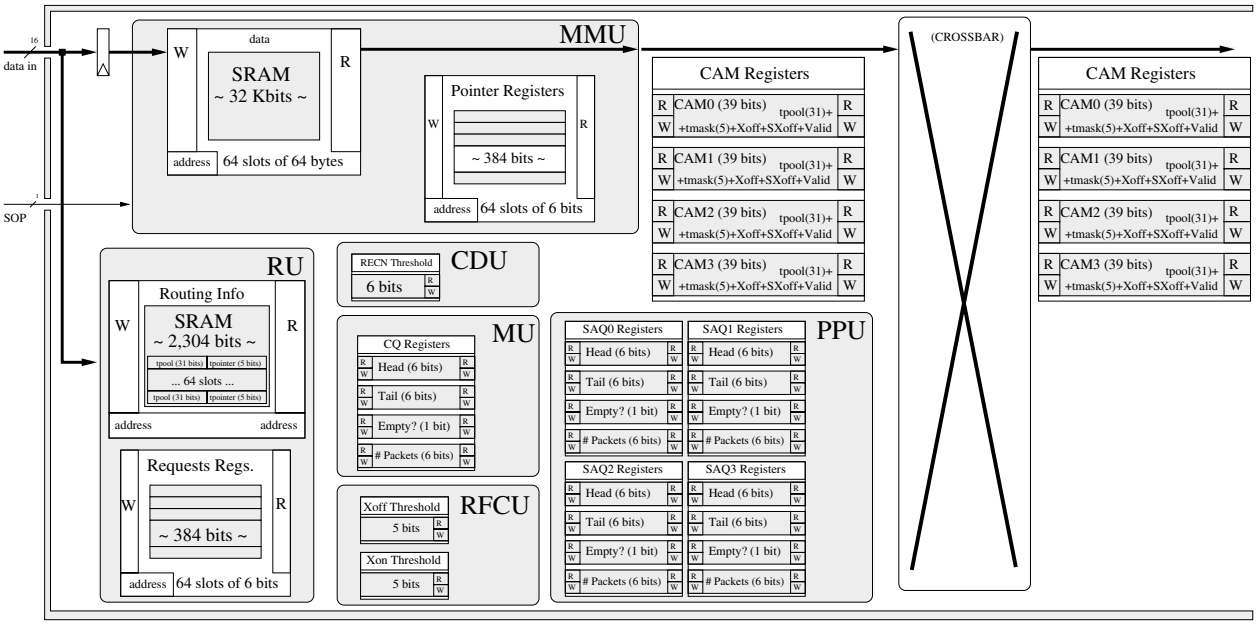
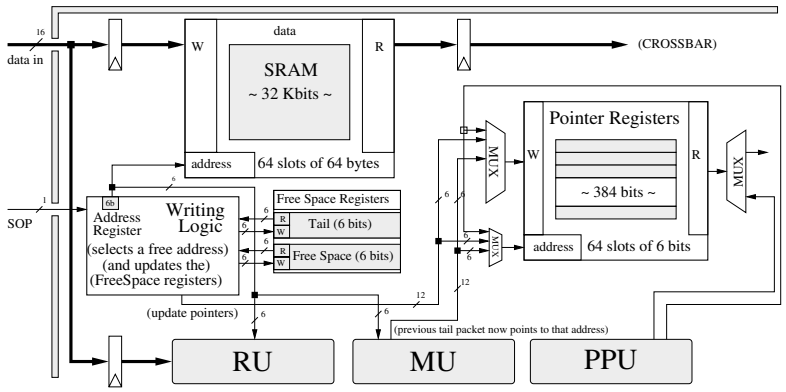
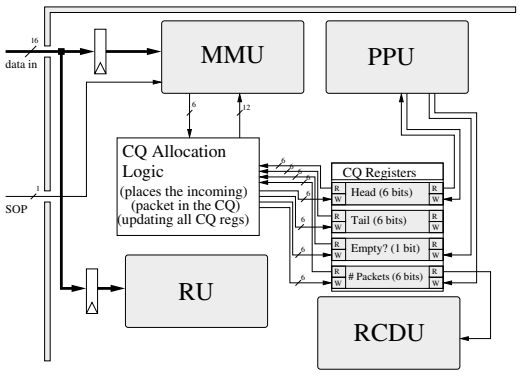


Figure 2. General overview of the proposed switch architecture.



(a) MMU (Memory Management Unit) detailed.



(b) MU (Mapping Unit) detailed.

Figure 3. MMU (Memory Management Unit) and MU (Mapping Unit) units in detail.

plemented queue (CQ or SAQs) a set of four registers will be managed to keep the queue structure and to keep track of the queue occupancy level: head, tail, empty and number-of-packets registers. Finally, also a CAM structure is required, but in this case at both ports, input and output.

3.1. Memory Management Unit

In the proposed switch architecture packets are stored only at input ports. Thus, the Memory Management Unit (MMU) is located at the input side of the switch. The MMU is in charge of mapping all the incoming packets to the corresponding queue and to keep track of the allocated queues within the memory. Figure 3.(a) shows the scheme for the MMU. This element works in a Dynamically Allocate Multi-Queue buffer (DAMQ) [7] fashion and it consists of a SRAM memory of 32 Kbits and an associated logic. The memory is divided into slots of 64 bytes (64 chunks of 64 bytes each). Slots are used atomically.

Whenever a new packet arrives (Start Of Packet signal, SOP) the writing logic selects a free slot in the memory for mapping the packet. In order to do this, the MMU incorporates two registers to keep track of the list of free slots, managed in a LIFO manner. Additionally, in order to keep track of the different queues implemented in the memory, the MMU incorporates a list of pointers, one for each pos-

2007 International Conference on Parallel Processing (ICPP 2007)
0-7695-2933-X/07 $25.00 © 2007 IEEE

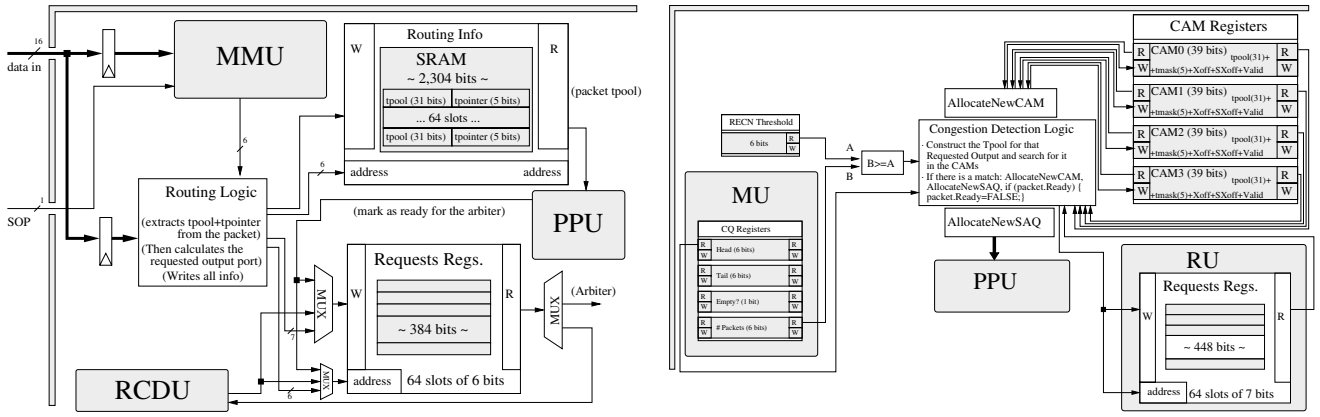


Figure 4. RU (Routing Unit) and CDU (Congestion Detection Unit) units in detail.

sible slot. Thus, it incorporates 64 registers. Each register corresponds to a slot and indicates which is the next slot in the queue's order.

3.2. Mapping Unit

As already explained, in the RECN-IQ mechanism, the memory of each input port is logically divided into a single (Cold) queue, and a set of SAQs for storing congested packets. In the proposed architecture, we assume four SAQs per set (this number of SAQs is enough for handling effectively congestion situations, as it will be demonstrated in the evaluation section). SAQs are dynamically allocated when required.Whenever a new packet arrives to an input port, it is stored always in the CQ queue. To keep track of the CQ, the Mapping Unit (MU) is used.

Figure 3.(b) shows the scheme for the MU. It consists of four registers and an associated logic. The logic places the incoming packet in the CQ and updates the registers. The Head and Tail registers keep track of the structure of the CQ whereas the remaining two registers keep track of the queue's occupancy.

3.3. Routing Unit

At the same time packets are being stored into the CQ, the switch routes the packet in order to decide which output port the packet requests. For this, the Routing Unit (RU) shown in Figure 4.(a) places the packet header in a separate SRAM memory of 2 Kbits. This memory contains the header of each packet stored in the port, so it is divided into 64 slots. Each slot contains a turnpool and a turnpointer following the AS packet header format. Thus, each slot will be 36 bits wide.

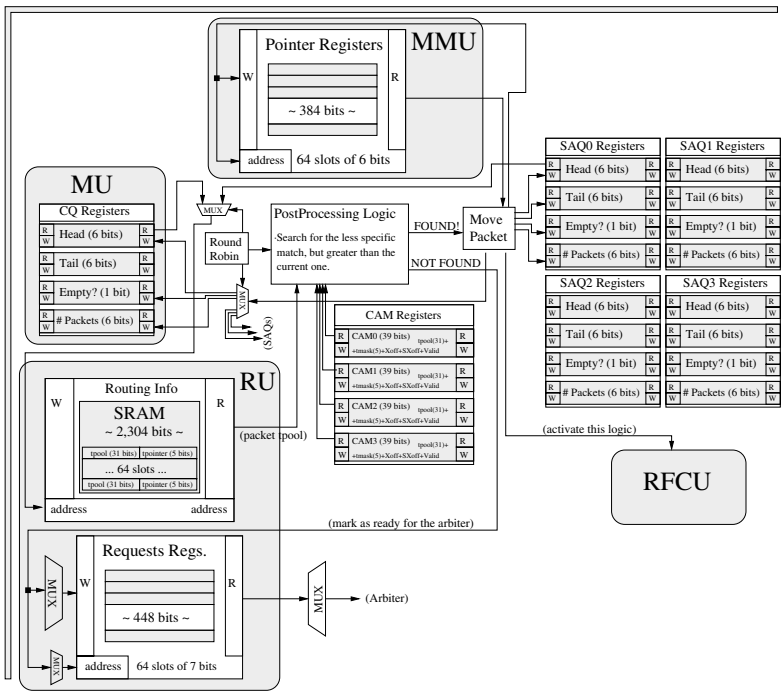
Additionally, the RU will extract from the header of the incoming packet the output port requested, and will store that information into the request register file (RRF), made up of 64 registers 7 bits wide each. The RRF will be inspected by the scheduler in order to decide when to forward the packet through the crossbar. However, one of these bits (the Ready-For-Scheduler, RFS bit) will be used in order to enable/disable the forwarding of the packet. This bit will be switched on by the PPU unit (see below). Anyway, whenever a new packet arrives the input port and is routed, the RFS bit is reset, thus the packet is disabled for being forwarded. Later, the PPU unit will set this bit, thus enabling the packet for the scheduler.

3.4. Congestion Detection Unit

The Congestion Detection Unit (CDU) shown in Figure 4.(b) is in charge of detecting congestion, computing the output port congested, and, if required, allocating a new SAQ for the congested point. First, it compares the current queue's occupancy of the CQ against a given RECN threshold (measured in number of packets or slots). If the threshold is reached, then the congestion detection logic is triggered. This logic will extract from the Request Registers File (RRF) the output port requested by the packet at the header of the CQ. Then, it will build the turnpool and the bit mask addressing that output port and will compare this information against all the routing info stored in the CAM line of each active SAQ. In order to implement these CAM lines, the CDU incorporates a register for each possible SAQ. If there is no match, a new SAQ for the congested output port will be allocated.

Notice that the logic associated to the CDU is enabled only when the detection threshold is reached. Thus, in the absence of congestion, most of the logic will be disabled.

2007 International Conference on Parallel Processing (ICPP 2007)
0-7695-2933-X/07 $25.00 © 2007 IEEE



(a) PPU (Post-Processing Unit) detailed.

(b) RFCU (RECN Flow Control Unit) detailed.

Figure 5. PPU (Post-Processing Management Unit) and RFCU (RECN Flow Control Unit) units in detail.

3.5. Post-Processing Unit

The Post-Processing unit (PPU) shown in Figure 5.(a) is in charge of separating the congested flows from the non-congested ones. Also, it will separate every congested flow from each other.

For keeping the logical structure of SAQs, the PPU requires some registers per SAQ (the same required for the CQ at the MU unit): Head and Tail registers for keeping queue structure and two registers to keep queue occupancy level.

The PPU works as a background task managing stored packets. Its main purpose is to classify packets according to the congested points already identified. To do this, the PPU continuously inspects the routing header (this info is located at the routing info memory at the RU) of each packet located at the head of any queue (CQ, SAQ0, SAQ1, SAQ2, and SAQ3). With this info, it checks if there is a match with any of the identified congested points. If so, the packet is moved to the SAQ associated to the congested point (it should be noted that the packet is not moved at all, only the pointers are updated).

Notice that the PPU may move packets from the cold queue (CQ) to a SAQ. In this case, the packet has been identified as passing through the congested point for which the SAQ has been allocated. The packet is simply moved to the tail of the SAQ. This is done by just adjusting the pointers of the CQ and the SAQ. Also, the Pointer Registers File (PRF) is updated accordingly.

However, note also that a packet stored at the head of a SAQ can be moved by the PPU unit. In this case, the unit moves the packet only if there is a match of the turnpool of the packet with the routing info associated to other SAQ allocated for a more specific congested point in the network (with a longer turnpool match).

Whenever a packet gets postprocessed (it is treated by the PPU unit), the RFS bit of the packet (located at the RRF file) is written. If the packet is moved (either from the CQ or from a SAQ) to a new queue, then, the bit is reset. On the contrary, if the packet is not moved by the PPU unit, then the RFS bit is set, thus allowing the scheduler to forward the packet through the crossbar.

3.6. Flow Control Unit

The RECN-IQ mechanism implements Xon/Xoff (Stop & Go) flow control for the SAQs. As seen in Section 2, these flow control signals are used as notifications of con-

2007 International Conference on Parallel Processing (ICPP 2007)
0-7695-2933-X/07 $25.00 © 2007 IEEE

gestion (propagation or vanishing) within the network as well. The RECN flow control unit (RFCU) is shown in Figure 5.(b).

Each time a packet is moved to a SAQ by the Post-processing unit, the RFCU logic is activated in order to check if the occupancy of the receiving SAQ goes over the Xoff threshold, and also to check if the occupancy of the sending SAQ (in the case of packets being moved from a SAQ) goes below the Xon threshold. Therefore, the RFCU compares the number of packets of each SAQ against the Xoff (Xon) threshold. If the SAQ occupancy has crossed the Xoff (Xon) threshold, and the SentXoff bit is unset (set), an Xoff (Xon) signal is sent backwards to the output port in the switch upstream. Then, the SentXoff bit for that SAQ is set (unset).

3.7. Scheduler and Global Flow Control

Although SAQs are individually flow controlled, the switch implements also a general (memory level) flow control mechanism. That is, each input port memory will have a number of credits available for packets. The number of credits will be the number of slots available in the memory, regardless of the receiving queue. Therefore, a packet will be transmitted over a link if the receiving memory has an available slot (a credit) and the transmitting queue is not blocked (in the case the transmitting queue is a SAQ and its associated CAM has the "Xoff" bit set). Once a packet is transmitted the switch decrements the number of available credits at the downstream memory. Therefore, at each output port the switch implements a counter of the number of credits available.

Therefore, the scheduler must take the credit counters at each output port into account when scheduling packets for transmission. Besides the credit counters, also, the scheduler will take into account the RFT bit of each packet and the Xoff bits of each CAM line associated to an active SAQ.

Regarding the arbiter, the switch architecture presented up to now is independent of the algorithm used to implement the arbiter. Anyway, we propose (and evaluate in the next Section) a round-robin arbiter with two phases. At the first phase an arbiter at each input port selects a queue with an available packet for transmission at its head. This arbiter performs a round-robin selection. At the second phase, an arbiter at each output port selects, in a round-robin fashion, one among all the possible requests of all the input ports.

4. RECN-IQ Evaluation

In this section, the RECN-IQ mechanism is evaluated by means of simulation results. Specifically, we show the network throughput and network latency 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-IQ configurations and also no SAQs for simulating switches not using RECN-IQ. Two different synthetic traffic patterns are used: uniform and hot-spot.

Regarding network size and topology, the following two Multi-stage Interconnection Network (MIN) configurations have been analyzed:

- Configuration 1: 64 end nodes, MIN made of 8x8 switches (48 switches in 3 stages, perfect shuffle as interconnection pattern).
- Configuration 2: 256 end nodes, MIN made of 8x8 switches (256 switches in 4 stages, perfect shuffle as interconnection pattern).

The organization of all the switches in those configurations is the one described in Section 3, so they are IQ switches in which RECN-IQ can be enabled/disabled. 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.

4.1. Results for Uniform Traffic

For the two MIN configurations considered, throughput and latency results obtained using uniform traffic and different numbers of SAQs per port can be seen in Figure 6.

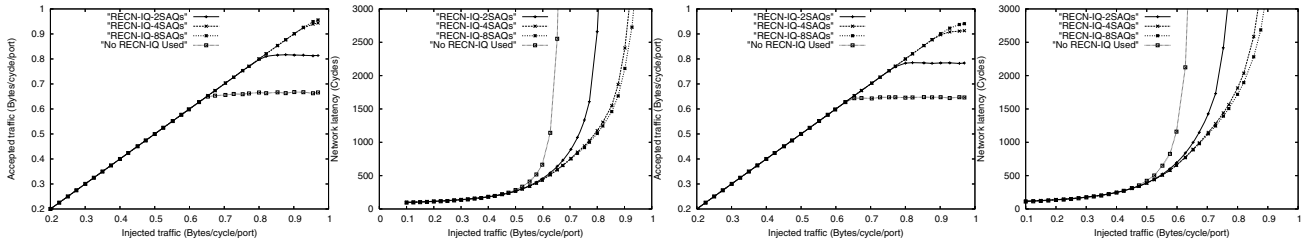
In the case of the 64-end node MIN (Figures 6.(a) and 6.(b)), the maximum efficiency for an IQ switch without RECN-IQ (no SAQs used) is about 65%. When using RECN-IQ and only 2 SAQs, the efficiency increases to above 80%. However, as can be seen in the figures, at least 4 SAQs are needed for achieving 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 efficiency can be increased from 65% to almost 80% when compared to the case of not using RECN-IQ (Figure 6.(c)).

4.2. Results for Hot-Spot Traffic

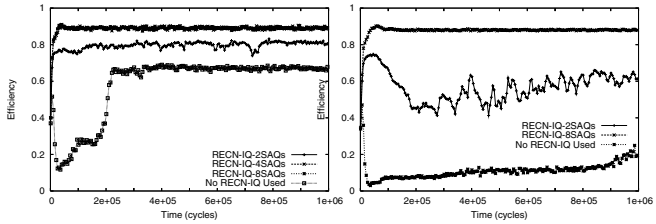
We have also made experiments using a hot-spot traffic pattern. Specifically, in this case all the end nodes inject 10% of their traffic to a single (hot-spot) end node (end node number 6 in this very case), whereas the rest of the traffic is randomly distributed among the rest of end nodes. Furthermore, the hot-spot is only injected during a small period of time at the start of the simulation (first 100,000 cycles).

2007 International Conference on Parallel Processing (ICPP 2007)
0-7695-2933-X/07 $25.00 © 2007 IEEE



(a) Network throughput. 64-end node network. (b) Network latency. 64-end node network. (c) Network throughput. 256-end node network. (d) Network latency. 256-end node network.

Figure 6. Network throughput and network latency. Uniform distribution of packet destinations.



(a) Switch efficiency. 64-end node network. (b) Switch efficiency. 256-end node network.

Figure 7. Switch efficiency versus time for Hot-Spot traffic.

Figure 7 depicts switch efficiency versus time for MIN configurations 1 and 2. As can be seen, when RECN-IQ is not used (no SAQs used), the switch efficiency drops to unaceptable levels during the hot-spot traffic injection. Moreover, in the case of the 256-end node MIN, the congested situation persists long time after the hot-spot injection ends.

The situation changes dramatically when using RECN-IQ. Figures 7.(a) and 7.(b) also depict the switch efficiency when using RECN-IQ with 2, 4 and 8 SAQs. For the 64-end node MIN, using 4 SAQs is enough for completely eliminate the negative side-effects of congestion, thereby achieving maximum switch efficiency. Anyway, note that with only 2 SAQs, 80% of switch efficiency can be achieved. For larger networks, like the 256-end node MIN, maximum switch efficiency is guaranteed by using 8 SAQs.

5. Conclusions

For modern interconnection networks, the use of an effective congestion management technique has become mandatory in order to keep network performance at maximum even in congestion situations. Although the formerly proposed RECN mechanism efficiently solves the problems related to congestion, its application is restricted to CIOQ switches, thereby not being suitable for the IQ switches.

In order to afford an effective congestion management technique to this type of switches, we have proposed in this paper an adaptation of RECN to IQ switches. The resulting mechanism, referred to as RECN-IQ, also introduces a new way for detecting congestion at input ports that significantly reduces the data memory area required at each port. From the evaluation results presented in this paper, we can deduce that RECN-IQ eliminates HOL blocking as well as RECN, while being an even more cost-effective technique.

Moreover, a feasible, realistic switch architecture implementing RECN-IQ has been proposed and described in detail in this paper. This architecture could be the base for switches which would allow to build cheaper networks featuring congestion management.

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 local-area networks. *ACM Transactions on Computer Systems*, 11(4):319–352, Nov 1993.

[3] W. J. Dally, P. Carvey, and L. Dennison. The avici terabit switch/router. *Proc. Hot Interconnects 6*, Aug 1998.

[4] J. Duato, I. Johnson, J. Flich, F. Naven, P. J. Garcia, and T. Nachiondo. A new scalable and cost-effective congestion management strategy for lossless multistage interconnection networks. *Proc. 11th International Symposium on High-Performance Computer Architecture (HPCA05)*, pages 108–119, Feb 2005.

[5] 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, Sep 2006.

[6] 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.

[7] Y. Tamir and G. L. Frazier. High-performance multi-queue buffers for vlsi communication switches. *ISCA '88: Proceedings of the 15th Annual International Symposium on Computer architecture*, 1988.

2007 International Conference on Parallel Processing (ICPP 2007)
0-7695-2933-X/07 $25.00 © 2007 IEEE