Available online at www.sciencedirect.com
ScienceDirect
Journal of Systems Architecture 53 (2007) 355–368
JOURNAL OF SYSTEMS ARCHITECTURE
www.elsevier.com/locate/sysarc

A low-cost strategy to provide full QoS support in Advanced Switching networks ☆

Alejandro Martínez *, Raúl Martínez, Francisco J. Alfaro, José L. Sánchez

Departamento de Sistemas Informáticos, Escuela Politécnica Superior, Universidad de Castilla-La Mancha, 02071 Albacete, Spain

Received 28 June 2005; received in revised form 4 July 2006; accepted 23 October 2006
Available online 8 December 2006

Abstract

Advanced Switching (AS) is an open-standard fabric-interconnect technology that is built over the same physical and link layers as PCI Express technology. Moreover, it includes an optimized transaction layer to enable essential communication capabilities, including protocol encapsulation, peer-to-peer communications, mechanisms to provide quality of service (QoS), enhanced fail-over, high availability, multicast communications, and congestion and system management.

In this paper, we propose a strategy to use the AS resources that provides a good performance and QoS support at a low cost. When the system is considered as a whole rather than each element being taken separately, it is possible to use only two virtual channels (VCs) at the switches to provide a service like that with many more VCs. As a result, we obtain a noticeable reduction of silicon area and arbitration time. Our proposal is fully compatible with the AS specification and permits us to provide an adequate performance both for typical multimedia applications and for best-effort traffic.

Keywords: PCI Express Advanced Switching; QoS; Network architecture; Arbitration

1. Introduction

The PCI bus has served industry well for the last 10 years and is currently used extensively. However, the processors and I/O devices of today and tomorrow demand much higher I/O bandwidth than PCI bus or PCI-X can deliver. The reason for this limited bandwidth is the parallel bus implementation. PCI Express [21] eliminates the legacy shared bus-based architecture of PCI and introduces an improved and dedicated point-to-point interconnect. Advanced Switching (AS) is a new open-standard fabric-interconnect technology for communications, storage and embedded environments based on PCI Express.

AS provides mechanisms that correctly used permit quality of service (QoS) to be supported. Specifically, an AS fabric permits the employment of virtual channels (VCs), traffic classes, egress link scheduling, and an admission control mechanism to provide a different treatment for the traffic of the various service classes.

☆ This work was partly supported by the Spanish CICYT under grant TIC2003-08154-C06, by Junta de Comunidades de Castilla-La Mancha under grant PBC-05-005, and by the Spanish State Secretariat of Education and Universities under FPU grants.
* Corresponding author. Tel.: +34 967 599200.
E-mail addresses: alejandro@dsi.uclm.es (A. Martínez), raulmm@dsi.uclm.es (R. Martínez), falfaro@dsi.uclm.es (F.J. Alfaro), jsanchez@dsi.uclm.es (J.L. Sánchez).

1383-7621/$ - see front matter
doi:10.1016/j.sysarc.2006.10.013

The AS specification supports up to 16 unicast VCs and up to 4 multicast VCs. However, seeing so many VCs in a final commercial implementation of any technology is unusual. In fact, it seems that, when the technology allows it, the trend is to increase the number of ports instead of increasing the number of VCs per port [18].

In most of the recent switch designs, the buffers at the ports are implemented with a memory organized in logical queues. These queues consist of linked lists of packets, with pointers to manage them. The complexity and cost of the switch and the scheduling delays heavily depend on the number of queues at the ports (see [24] for a detailed design). VCs, which can be used for many purposes, are implemented as queues of this kind. Thus, a reduction of the number of VCs necessary to support QoS can be very helpful in the switch design and implementation.

In this paper, we show that it is enough to use two VCs at each switch port for the provision of QoS. One of these VCs is used for QoS traffic and the other for best-effort traffic. Although this is not a new idea, the novelty of our proposal lies in the way in which we use those two VCs, which allows a network behavior very similar to that of a network with many more VCs. We achieve it reusing at the switches some of the scheduling decisions made at network interfaces.

Our objective is to apply this strategy in an AS environment to provide applications with an adequate QoS performance, but using fewer VCs, which would result in less silicon area and a reduced processing delay. We will compare the performance of our proposal with those obtained using the two normative arbiters defined in the AS specification: The VC arbitration table scheduler and the minimum bandwidth egress link scheduler.

The remainder of this paper is structured as follows. In the next section, we review the AS specification, with an emphasis on QoS support. In Section 3, we propose how to use the mechanisms included in the AS specification for QoS support. This is followed by a description of our strategy for full QoS support with just two VCs in Section 4. Details on the experimental platform and the performance evaluation are presented in Section 5. Finally, Section 6 summarizes the results of this study.

2. Advanced Switching review

AS is a multipoint, peer-to-peer switched-fabric architecture designed to provide, in an open standard, the functionality of the proprietary interconnects that have been at the core of storage, communications, and embedded computing systems. Recently, the AS Interconnect Special Interest Group [1] has published v1.1 of the AS specification [2].

AS architecture is built upon the data link and physical layers established by the PCI Express architecture [21] to achieve widespread interoperability and cost-effective reuse of technology. The physical layer consists in a dual-simplex channel that is implemented as a transmit pair and a receive pair. A data clock is embedded using the 8b/10b encoding scheme. The initial frequency is 2.5 Gb/s, but the bandwidth of a link may be linearly scaled by adding signal pairs to form multiple lanes. In AS, the maximum packet size is 2176 bytes. A credit-based flow control protocol ensures that packets are only transmitted when the buffer at the other end is able to receive those packets. Finally, virtual cut-through switching is used in AS.

AS supports unicast and multicast traffic. For unicast traffic the AS transaction layer provides source-based path routing versus the memory-mapped routing of PCI Express. By eliminating the hierarchical structure of memory-mapped routing, flexible topologies can be constructed such as star, dual-star, full mesh, or multi-stage networks.

AS encapsulates data packets and attaches to them a header that routes the packets through the fabric, regardless of the original packet format. This header contains a protocol interface field that is used at the packet destination to determine the packet's format. Thus, nearly any transport, network, or link layer protocol can be routed through an AS network.

2.1. AS Support for QoS

AS provides mechanisms that permit QoS to be supported. Specifically, an AS fabric permits us to employ traffic classes (TCs), VCs, egress link scheduling, and an admission control mechanism to provide a different treatment to the traffic.

VCs provide a means of supporting multiple independent logical data flows over a given common physical channel. AS supports up to 20 VCs of three different types: Up to 8 bypassable unicast VCs, up to 8 ordered-only unicast VCs, and up to 4 multicast VCs. The bypassable VC with the highest number in each network element (usually VC 7) is called the fabric management channel.

The AS packet header contains a field with a TC identifier. This field permits us to specify one of eight possible TCs. Moreover, the AS packet header specifies the type of VC that the packet employs. The packet's TC identifier and the VC type are transmitted unmodified from source to destination through an AS fabric. At each hop within an AS fabric, the TC identifier is used to apply VC selection of the appropriate type. Each VC type (bypassable or ordered) is governed by a distinct TC/VC mapping.

AS defines two egress link schedulers to resolve between the up to 20 VCs competing for bandwidth on the egress link: The VC arbitration table scheduler and the minimum bandwidth egress link scheduler. A given implementation may choose either of them or may implement its own proprietary mechanism.

The VC arbitration table scheduler, or just table scheduler, provides an implementation of the Weighted Round Robin (WRR) algorithm, proposed by Katevenis et al. [13]. The VC arbitration table is a register array with fixed-size entries of 8 bits. Each table entry, which contains a VC identifier value, corresponds to a slot of a WRR arbitration period. When arbitration is needed, the table is cycled through sequentially and a packet is transmitted from the VC indicated in the current table entry. If the current entry points to an empty VC, that entry is skipped. The number of entries of the VC arbitration table may be 32, 64, 128, 256, 512, or 1024.

The minimum bandwidth egress link scheduler, or just MinBW scheduler, is intended for a more precise allocation of bandwidth regardless of packet size. This scheduler consists of two parts. The first is a mechanism to provide the fabric management channel with an absolute priority over the other VCs. However, it has its bandwidth limited by a token bucket. The second part of the MinBW egress link scheduler is a mechanism to distribute bandwidth amongst the rest of the VCs according to a specification of relative weights.

AS does not specify an algorithm or implementation for the MinBW scheduler, but only its behavior. However, according to the specification, several well-known scheduling algorithms exhibit the desired properties of the MinBW scheduler. Examples include variants of Weighted Fair Queuing (WFQ) [6] such as Self-Clocked WFQ [9], and variants of WRR such as Deficit WRR [23].

A connection admission control implemented in the fabric management software may regulate the access to the AS fabric. It would allow new packet flows entry to the fabric only when sufficient resources were available. Fabric management software may track resource availability by monitoring AS fabric congestion and tracking active packet flows and their bandwidth. This is very useful when traffic flows are predominately connection-oriented and carefully rate-limited.

3. Providing QoS over AS using normative mechanisms

In [17], we proposed a way of using some of the above-presented AS mechanisms in order to provide QoS. First of all, a set of service classes (SCs) with different requirements must be specified. When a flow accesses the AS fabric, it is aggregated into a SC depending on its characteristics. We propose to employ a SC for network control traffic, some for traffic with QoS requirements and the rest for best-effort traffic. If there are enough VCs we would devote a different VC to each existing SC. However, if there are not enough implemented VCs along the whole path of a connection, more than one SC should be assigned to the same VC.

The schedulers must be properly configured at the network elements to provide a differentiated treatment to the VCs. In order to do this, we consider two kinds of QoS requirements: maximum delay and minimum bandwidth.

The network control traffic will be assigned to the fabric management channel in order to get the maximum priority when using the MinBW scheduler. In case of using the table scheduler, the fabric management channel is processed in the same way as the other VCs, so we will consider it as a VC with high latency requirements. Best-effort SCs are only characterized by the differing priority among them. We will assign them a small amount of bandwidth proportional to their relative priority.

In [3] we explained how to configure an arbitration table (in that case for InfiniBand), similar to the AS arbitration table, to provide bandwidth and latency guarantees. In order to provide traffic of a given VC with a minimum bandwidth, the number of table entries assigned to that VC must be proportional to the desired bandwidth. In order to provide maximum delay requirements to a VC, the maximum separation between two consecutive table entries devoted to that VC must be fixed to an appropriate value. This allows us to control the

maximum latency to cross each network element and, therefore, the global delay.

Providing minimum bandwidth requirements to a VC with the MinBW scheduler is as easy as assigning to the VC in question a weight equal to the proportion of the egress link bandwidth that it needs. AS specification states that some implementations of WFQ exhibit the desired properties of the MinBW scheduler. Parekh and Gallager [19,20] analyzed the performance of WFQ from the standpoint of worst-case packet delay. Based on this study we will assign a higher amount of bandwidth than is needed to VCs with high latency requirements, in order to obtain an appropriate average and maximum delay performance.

Finally, to provide QoS guarantees a connection admission control (CAC) must be used. Without the CAC, it is only possible to obtain a scheme of priorities where some SCs would have a higher priority than others, but no guarantee could be given. In any case, the admission control would not be used over network control and best-effort traffic. The AS specification states the possibility of employing a CAC mechanism.

4. Providing full QoS support with only two VCs

The buffers at the ports of the switches are usually implemented with a memory organized in logical queues. These queues need control data structures in order to be managed properly. As a consequence of this, the complexity and cost of the switch and the scheduling delays heavily depend on the number of queues at the ports.

In this paper, we propose a switch architecture for AS with only two VCs at each switch port for QoS support. One of these VCs would be used for QoS and network control packets, and the other VC for best-effort packets.

4.1. Related work

During the last decade several switch designs with QoS support have been proposed. All of them incorporate VCs in order to provide QoS support.

The multimedia router (MMR) [7] is a hybrid router. It uses pipelined circuit switching for multimedia traffic and virtual cut-through for best-effort traffic. Pipelined circuit switching is connection-oriented and needs one VC per connection. This is the main drawback of the proposal because the number of VCs per physical link is limited by the available buffer size and there may not be enough VCs for all the possible existing connections (in the order of hundreds). Therefore, the number of multimedia flows allowed is limited by the number of VCs. Moreover, the scheduling among hundreds of VCs is a complex task.

MediaWorm [26] was proposed to provide QoS in a wormhole router. It uses a refined version of the virtual clock algorithm [28] to schedule the existing VCs. These VCs are divided into two groups: One for best-effort traffic and the other for real-time traffic. Several flows can share a VC, but 16 VCs are still needed to provide QoS. Besides, it is well known that wormhole is more likely to produce congestion than virtual cut-through. In [27], the authors propose a preemption mechanism to enhance MediaWorm performance, but in our view that is a rather complex solution.

InfiniBand was proposed in 1999 by the most important IT companies to provide present and future server systems with the required levels of reliability, availability, performance, scalability and QoS [11]. Specifically, the InfiniBand Architecture (IBA) proposes three main mechanisms to provide the applications with QoS. These are traffic segregation with service levels, the use of VCs (IBA ports can have up to 16 VCs) and the arbitration at output ports according to an arbitration table. Although IBA does not specify how these mechanisms should be used, some proposals have been made to provide applications with QoS in InfiniBand networks [3].

These proposals, therefore, use a significant number of VCs to provide QoS support. However, if a great number of VCs are implemented, it would require a significant fraction of silicon area and would make packet processing slower. Note that this paper deals with single-chip switches, where the buffers, the crossbar, and the scheduler are inside the same chip. This is necessary in order to offer the low cut-through latencies demanded by current applications.

Traditional two VC proposals distinguish between just two broad categories (regular and premium) [5,14]. In contrast, the novelty of our proposal lies in the fact that, although we use only two VCs at the switches, the global behavior of the network is very similar as if the switches were using many more VCs. This is because we are reusing at the switch ports the scheduling decisions taken at the network interfaces, which have as many VCs as traffic classes. In the end, the network

provides a differentiated service to all the traffic classes considered.

To the best of our knowledge, only Katevenis and his group [4] have proposed something similar before. However, their proposal is aimed at a single-stage router based on a single buffered crossbar. This crossbar has small buffers at the crosspoints that the authors split into two VCs. In contrast, our proposal is a simpler and more general technique, as we will see in the next section.

4.2. Our proposal

The basic idea of our proposal, deeply explained in [15], consists in using full VC support at the network interfaces, but only two VCs at the switch ports. One of these VCs would be used for QoS and network control packets and the other for best-effort packets. We reuse at switches the scheduling decisions performed at network interfaces. This allows us to achieve a performance similar to that obtained by systems with many more VCs.

Fig. 1 shows an example of a network interface that is connected to a switch. Note that at both the network interface and the switch input port, there are several VCs. When a packet arrives at the switch, the header is analyzed and the packet is then usually stored in a VC according to the flow or class to which it belongs. However, packets arriving at the switch have been previously sorted by the network interface according to certain criteria.

If we separated the packets in different VCs, we would lose this order, which may contain enough information to simplify the scheduling at the switch. However, it is not enough to put all the packets in the same VC to reuse the scheduling decisions. It is also necessary that the crossbar scheduler considers the global priority of the packets. This is the main difference from a traditional 2 VC design, which would only consider two categories in both the network interfaces and the switches.

In order for our proposal to be effective we need to make two assumptions. The first one is that a static priority criterion exists to order packets. In this way, every packet would be stamped with a priority level. The TC identifier at the AS packet header serves this purpose perfectly. This requirement is necessary because we will maintain the incoming order along the whole network. This, however, is not a great problem because queuing delays for QoS traffic will be short and therefore, the packet ordering established at network interfaces does not need to be changed at any switch in the path.

The second assumption is that there must be a connection admission control (CAC) for the traffic with QoS requirements, so that no link is oversubscribed by QoS traffic. This requirement is needed to provide bandwidth guarantees and to avoid starvation of the QoS traffic. It is also necessary to assure that this kind of traffic will flow with short delays. For that purpose, we would use the CAC considered in the AS specification.

Note that although we assume that QoS traffic does not oversubscribe any link, no assumption is made about best-effort traffic. Thus, if we did not separate QoS traffic and best-effort traffic, the total bandwidth demand for a given output link could exceed the available bandwidth. For this reason,

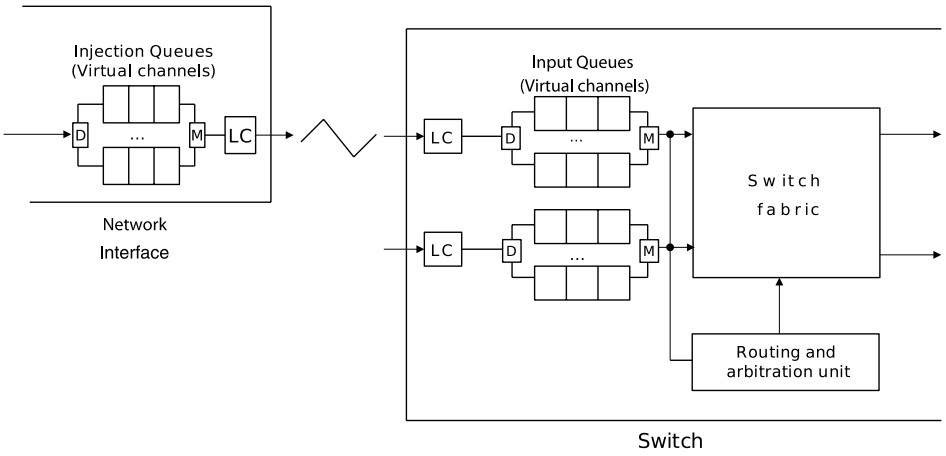

Fig. 1. QoS support at the network interface and the switch.

we cannot use just one VC, and hence our proposal to use two VCs at the switches.

It is important to note that the order of the different best-effort SCs is also kept with this design. Although we use only one VC for best-effort traffic, we also consider the different priorities of packets belonging to this group. This means that the switch will also differentiate between these kinds of packets, as we will see in the next section.

Now, we will proceed to describe in depth how our proposal works. Let us suppose that several packets arrive at a switch from a network interface. Taking into account that the interface implements a priority-based arbiter, the first packet should be the one with the highest priority. So, instead of separating the packets among several VCs according to their SCs, we put them all in the same queue in the arrival order. Later, when the switch must decide which packets should be transmitted, it will seek in the input queues. Note that it is only necessary to look at the first packet in each queue, because its position at the front of the queue indicates that it had a higher priority when it left the network interface. However, this crossbar scheduler must take into account the global priority of the packets, which is present in the AS header, and not only the VC where they are stored.

Obviously, the network interface can only arbitrate among the packets it holds at a given moment. Therefore, when no more high-priority packets are available, a low-priority QoS packet can be transmitted. If this packet has to wait at a switch input queue, and other packets with higher priority are transmitted later from the network interface, they would be stored in the same VC as the low-priority packet, and be placed after it in the queue. Thus, the arbiter would penalize the high-priority packets, because they would have to wait until the low-priority packet is transmitted.

This situation, which we call *order error*, has little impact on performance because there is bandwidth reservation for QoS packets and, therefore, the possible congestion would only happen for short periods of time. This means that all the QoS packets will flow with short delay, as we will see in the performance evaluation section.

Summing up, our proposal consists in reducing the number of VCs at each switch port that are needed to provide flows with QoS. Instead of having a VC per SC, we propose to use only two VCs: One for QoS packets and another for best-effort packets. The scheduling decisions performed at network interfaces are reused at switches and it is possible, therefore, to achieve a performance similar to that obtained by systems with many more VCs. For this strategy to work, we must guarantee that there is no link oversubscription for QoS traffic by using a suitable CAC strategy. It is important to note that this proposal does not aim at achieving a higher performance but, instead, at drastically reducing the switch complexity while keeping the performance and behavior of systems with many more VCs.

5. Performance evaluation

In this section, we evaluate three different scenarios, using three schedulers for the AS switches: The two defined in the AS specification and our own proposal. First, we will explain the simulated AS architecture. Next, we will give details on the network parameters and the load used for the evaluation. Finally, we present and comment on the results.

5.1. Simulated architecture

The network used to test the proposals is a perfect-shuffle multi-stage interconnection network (MIN) with 64 end-points (Fig. 2). In AS, any topology is possible, but we have chosen a MIN because it is a usual topology for high-performance interconnects. The switches use a combined input and output buffer architecture, with a crossbar to connect the buffers. We are assuming some internal speed-up (×1.5), as is usually the case in most commercial switches. We use virtual output queuing (VOQ) at the switch level, which is the usual solution to avoid head-of-line blocking.

In Table 1, there is a summary of the characteristics of the three architectures we are evaluating. The *Table 8 VCs* and *WFQ 8 VCs* proposals are equal, except for the scheduling applied at the output ports. The switches have 8 ports, each one implementing 8 VCs. Each VC is further divided in the input ports with a queue per output port to implement VOQ, giving a grand total of 64 queues per port. In that cases, the scheduler has to decide over up to 512 packets. On the other hand, each VC has 16 kbytes of available buffer, which is dynamically shared among the virtual output queues. This makes a total of 256 kbytes per port.

The switch architecture we propose has also eight ports. However, there are only 2 VCs per port and, therefore, 16 queues per port. Moreover, the

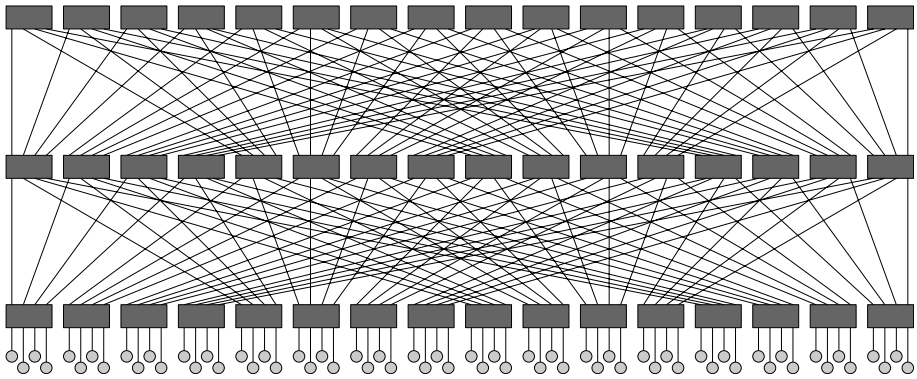

Fig. 2. 64 port folded perfect-shuffle MIN with 8 port switches.

Table 1
Simulated architectures

Architecture	Ports	VCs	Queues/port	VC memory	Port memory
Table 8 VCs	8	8	64	16 kbytes	256 kbytes
WFQ 8 VCs	8	8	64	16 kbytes	256 kbytes
New 2 VCs	8	2	16	16 kbytes	32 kbytes

amount of buffer space is again 16 kbytes per port, but the total per port is only 32 kbytes. Since the scheduler has to consider one fourth of the number of queues, it should be possible to reduce its complexity and the length of scheduling cycles. Considering the research in the area, namely Peh's work [22], our proposal should provide a speed-up of 2.0 in this delay. However, for the sake of clarity, the same scheduling time has been considered in the three cases.

In our tests, the link bandwidth is 2.5 Gb/s but, with the 8b/10b encoding scheme, the effective maximum bandwidth for data traffic is only 2 Gb/s. As mentioned before, AS defines a credit-based flow control to avoid dropping packets. The remaining parameter values are picked from the AS specification and are given in Table 2.

AS gives freedom to use any algorithm to schedule the crossbar. For the two arbiters proposed in the specification, we have implemented a First Come, First Served scheduler. For our proposal, we have used an arbiter as described at Section 4.

We have modeled an entire network, including the network interfaces at the end-points. These interfaces must implement an 8 VC priority arbiter in order for our proposal to work. This is necessary because our switches reduce their complexity by reusing part of the scheduling decisions made at the network interfaces. For the other two architectures, we have provided the end-points with the same egress schedulers as the switches and 8 VCs.

The CAC we have implemented is a simple one, based on average bandwidth. Each connection is assigned a path where enough resources are assured. We also use a load-balancing mechanism, which consists in assigning the least occupied route among those possible.

Table 2
Simulation parameters

Data packet size	64–2176 bytes
Packet header size	8 bytes
Credits message size	48 bits
Channel bandwidth	2 Gb/s
Crossbar bandwidth	3 Gb/s
Network interfaces	64

5.2. Traffic model

The IEEE standard 802.1D-2004 [10] defines seven SCs at the Annex G, which are particularly appropriate for this study. Table 3 shows each SC and its requirements. In this way, the workload is composed of seven different SCs. For simplicity, we will only use the bypassable VCs, assigning each SC a different TC identifier. Note that each TC will be assigned to a different VC, except when using our proposal. Each TC has increasing priority, such that

Table 3
Service classes suggested by the standard IEEE 802.1D-2004

Type	SC	Description
Best-effort	Background (BK)	Bulk transfers and other activities that should not impact the use of the network by other applications
Best-effort	Best-effort (BE)	LAN traffic as we know it today
Best-effort	Excellent-effort (EE)	The best-effort type services that an information services organization would deliver to its most important customers
QoS	Controlled load (CL)	Traffic from applications subject to some form of admission control based on bandwidth
QoS	Video (VI)	Traffic with a limit of 100 ms for latency and jitter
QoS	Voice (VO)	Traffic with a limit of 10 ms for latency and jitter
Control	Network control (NC)	Traffic to maintain and support the network infrastructure characterized by a "must get there" requirement

TC 7 has the highest priority and TC 0 has the lowest.

In Table 4, we show the characteristics of the traffic injected to the network from each node. This situation responds to the use of a CAC for the traffic with QoS requirements. The amount of QoS traffic that is going to be injected is the maximum allowed by the CAC. However, in a real situation, the amount of best-effort traffic is uncontrolled. We simulate this by increasing the amount of injected best-effort traffic. Our intention is to put the network in a high load and see if the QoS SCs are able to obtain their requirements.

We follow the recommendations of the network processing forum switch fabric benchmark specifications [8]. We use a uniform distribution of destinations to fully utilize the network capacity. In our tests, the packets are generated according to different distributions, as can be seen in Table 4. Audio, video, and controlled load traffic are composed of point-to-point connections of the given bandwidth. Note that audio traffic models both the audio part of the video transmissions and plain audio connections.

The self-similar traffic is composed of bursts of packets heading to the same destination. The packet size is governed by a Pareto distribution, as recommended in [12]. In this way, many small size packets are generated, with an occasional large size packet. The periods between bursts are modelled with a Poisson distribution. With this traffic model, if the burst size is long, there is a lot of temporal and spatial locality and should show worst-case behavior because at a given moment, many packets are grouped going to the same destination. Therefore, we use a long burst value of 30 packets for best-effort traffic.

5.3. Scheduler configuration

The simulations compare our AS proposal with the two based on the arbiters defined in the AS specification: The VC arbitration table scheduler and the MinBW egress link scheduler. Table 5 shows the VC arbitration table and the MinBW egress link scheduler configuration. Note that the NC SC is assigned to the FMC VC.

For the sake of simplicity, a table of 64 entries has been used in the simulations for the table arbiter (*Table 8 VCs* in the figures). The VC that accommodates the control traffic has been given a maximum separation between table entries of 4, which is a bal-

Table 4
Injected traffic

TC	SC	Min. %	Max. %	Packet size (bytes)	Traffic pattern
7	NC	1	1	64	Self-similar
6	VO	18.75	18.75	128	64 Kb/s CBR connections
5	VI	18.75	18.75	Up to 2176	3.3 MB/s MPEG-4 traces
4	CL	18.75	18.75	2176	750 Kb/s CBR connections
3	EE	5.25	17.58	Up to 2176	Self-similar
2	BE	5.25	17.58	Up to 2176	Self-similar
0	BK	5.25	17.58	Up to 2176	Self-similar
		73	110		

Table 5
Scheduler configuration

VC	SC	Table			MinBW
		% entries	# entries	Max. sep.	Asig. bandw.
7	NC	25	16	4	[c]
6	VO	25	16	4	0.25
5	VI	18.75	12	6	0.1875
4	CL	18.75	12	[b]	0.1875
3	EE	7.8125	5	[b]	0.078125
2	BE	3.125	2	[b]	0.03125
1[a]	–	0	0	–	0
0	BK	1.5625	1	[b]	0.015625
		100	64		0.75

[a] This VC is not used.
[b] Unspecified separation.
[c] This VC is not handled by the WFQ scheduler.

ance between a good latency and not too many expended entries. This number of table entries represents 25% of bandwidth, more than enough to fulfill the bandwidth requirements of the network control traffic. A small number of table entries has been assigned to best-effort traffic, in proportion to its importance. The remaining table entries have been assigned to QoS traffic (voice, video and controlled load). Maximum separations of 4 and 6 have been assigned to audio and video traffic, respectively. We have chosen these values because our tests have shown them as appropriate for those SCs. The controlled load traffic is assigned the same number of table entries as video traffic, but no restriction of maximum separation has been taken into account.

In order to configure the MinBW egress link scheduler, no weight must be assigned to control traffic because this traffic has been assigned to the FMC, and thus, it has strict priority over the rest, as specified in the AS standard. The remaining VCs have been assigned a weight equal to the proportion of traffic reserved in the table. Note that the voice VC is assigned a higher amount of bandwidth than it theoretically requires, but this is necessary to fulfill its latency requirements.

As discussed before, the AS specification states that several well-known scheduling algorithms exhibit the desired properties of the MinBW scheduler. In order to choose a specific implementation, we have discarded the variants of WRR because they generally produce worse latency and fairness properties compared to variants of WFQ [25]. Most WFQ variants, such as self-clocked WFQ, are different approaches of WFQ in order to reduce its complexity. Therefore, we have chosen to use the original WFQ algorithm, because we think that it is the best option among those proposed by the specification to make performance comparisons.

However, the use of this algorithm in the AS environment faces two problems. The first one is that the amount of flow control credits is not considered to determine the active set of VCs. The second problem is that this algorithm does not take into account the time used to transmit control packets, which are not controlled by the WFQ algorithm. In order to solve these problems we propose a new version of the WFQ algorithm, which we have called weighted fair queuing credit aware (WFQ-CA) [16]. The WFQ-CA works in the same way as the WFQ algorithm except in the following aspects:

- A VC is active only when it has a packet and there are enough credits to transmit the packet that is at the head of the VC queue.
- When a packet belonging to an active VC is received, it is stamped with its *virtual finishing time*. When a VC is inactive because of lack of credits and receives enough credits to be able to transmit again, the packets in that VC are restamped as if they had arrived in that instant. This permits us to implement the memoryless property that an AS scheduler must have.
- The value of the internal clock that the algorithm uses is not changed during the transmission of a control packet.

This new algorithm accomplishes all the properties that the AS MinBW scheduler must have and, thus, can be implemented in this new technology. Therefore, we employ this algorithm in the *WFQ 8 VCs* case.

The scheduler based on our proposal (*New 2 VCs* in the figures) uses the configuration described in Section 4. It uses a strict priority criterion, only 2 VCs at the switch ports, and 8 VCs at the network interfaces. It is important to note that our proposal does not imply any modification of the AS standard, and it is fully compatible with the specification.

5.4. Simulation results

We have considered three traditional QoS indices for this performance evaluation: Throughput, latency, and jitter. For a given network load level, several simulations have been conducted. In this way, we show the average values and the confidence intervals at 95% confidence level.

Maximum jitter determines the receiver's user space for audio and video packets. Inappropriate results of latency or jitter may lead to dropped packets at the application level. For that reason, we show the maximum values of latency and jitter. However, no packets are dropped at network level due to the flow control.

Fig. 3 shows the latency results for network control traffic. The *X* axis is the normalized load of the network. We can see that the three cases succeed in getting a reasonable average and maximum latency. However, the *Table 8 VCs* arbiter, even after devoting 25% of table entries to it, gets the worst performance. For the *WFQ 8 VCs* and *New 2 VCs* cases, the results are better. This is because, in the *WFQ 8 VCs* and *New 2 VCs* cases, this SC has a strict priority over the rest of the traffic. However, since network control traffic is mixed with other classes in the 2 VCs, the maximum latency is affected, but still in acceptable levels.

In Fig. 4, we show the performance of audio traffic. Remember that, according to the IEEE guidelines, this SC should achieve an average latency and jitter lower than 10 ms. For the *WFQ 8 VCs* and the *New 2 VCs* cases this is easily achieved.

The *Table 8 VCs* arbiter, however, yields a very bad performance. This arbiter cannot even guarantee the minimum bandwidth for audio traffic. The reason for this is that each entry in the AS table allows one packet to leave, whatever its size. For that reason, the arbiter penalizes audio traffic, since packet size for audio traffic is very small.

Fig. 5 shows the QoS results for video traffic. In this case, the *Table 8 VCs* arbiter performance is much better than in the audio case, due to the unfair advantage taken by video packets over the smaller audio ones. Again, the *WFQ 8 VCs* arbiter provides acceptable results, but the *New 2 VCs* achieves better performance.

We can find the performance of controlled load traffic in Fig. 6. All the arbiters provide an optimum throughput. Our proposal increases the average latency, but this is not a requirement for the controlled load traffic.

We can conclude at this point that the *WFQ 8 VCs* and the *New 2 VCs* cases provide a good QoS performance for the QoS traffic. However, the *Table 8 VCs* arbiter is not valid for this purpose due to its problems with variable packet size.

Fig. 7 shows the performance of the best-effort traffic in terms of throughput. We can see that the *WFQ 8 VCs* and the *Table 8 VCs* cases provide a throughput proportional to the assigned weights and number of table entries, respectively. In the *Table 8 VCs* case best-effort SCs obtain a higher throughput due to the unfair advantage that these SCs take over audio traffic.

The *New 2 VCs* case provide an absolute priority performance. The excellent-effort SC, which has the highest priority of the best-effort SCs, obtains the

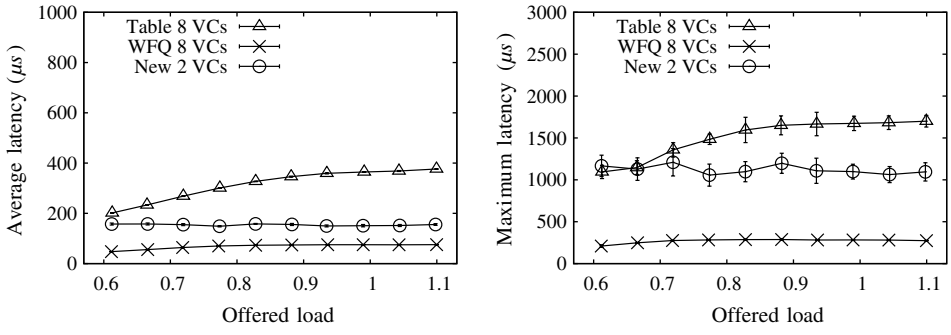

Fig. 3. Latency results for network control traffic.

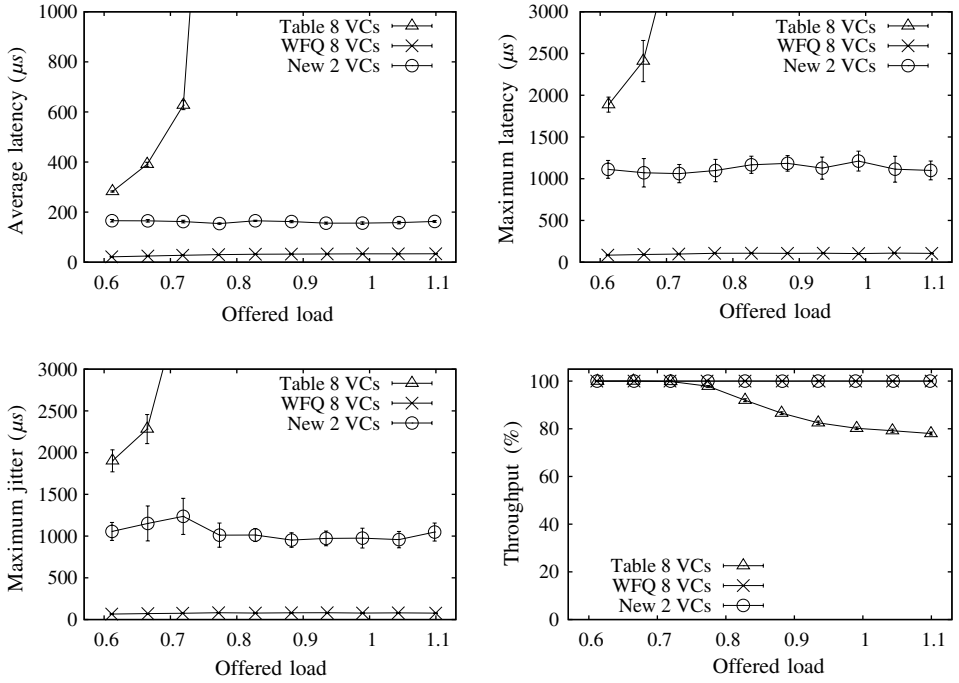

Fig. 4. Performance of audio traffic.

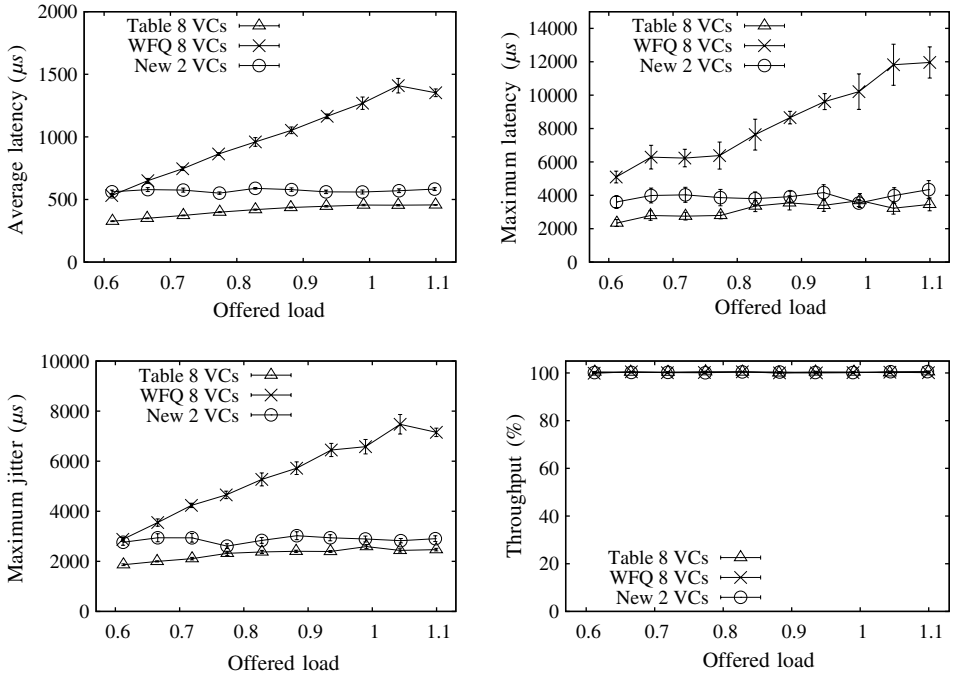

Fig. 5. Performance of video traffic.

highest throughput while the background SC is starved at high loads. This is not a handicap since the background SC should not impact the use of the network by other SCs.

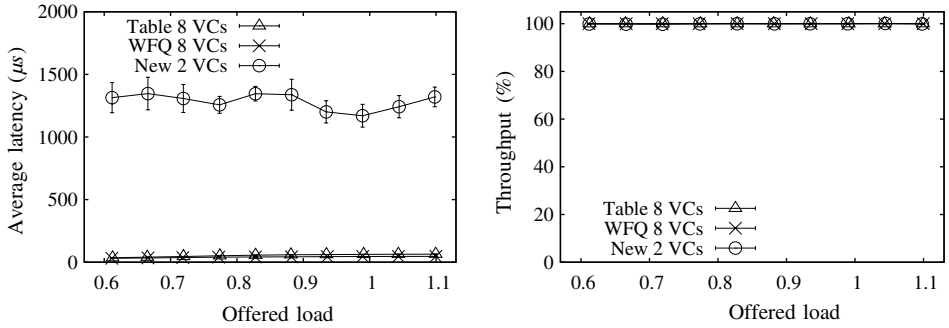

Fig. 6. Average latency and throughput for controlled load traffic.

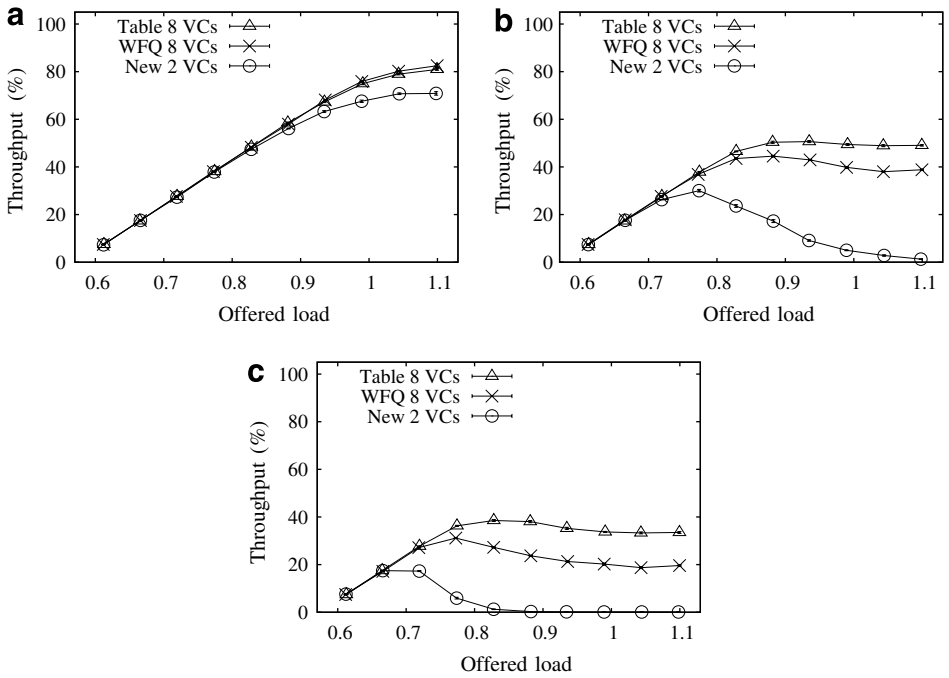

Fig. 7. Throughput for best-effort SCs. (a) Excellent-effort, (b) best-effort and (c) background.

We can conclude that the table arbiter is very limited and clearly inappropriate for providing QoS if variable packet size is used. In this case, this arbiter penalizes audio traffic, since packet size for audio traffic is usually very small.

The WFQ (MinBW) arbiter is able to provide a correct QoS, but needs more VCs and, thus, more buffer space to work. Finally, our proposal of using only two VCs at the switch ports can provide an adequate QoS both to multimedia traffic and to best-effort traffic. Our technique is also able to differentiate among the 7 SCs, although it only uses two VCs at the switches. The switch model we have proposed reduces the number of VCs and the associated memory.

6. Conclusions

In the Advanced Switching (AS) specification, three possible arbiters are proposed: Table-based, MinBW, and proprietary. In this paper, we have reviewed the two normative arbiters and we have proposed a fully compatible custom switch design with only two VCs.

We have presented a new proposal for QoS support at the AS switches. This consists in making the network elements co-operate, building together ordered flows of packets. Consequently, the switches try to respect the order in which packets arrive at the switch ports, which is probably correct. This allows a drastic reduction in the number of

VCs required for QoS purposes at each switch port, reducing both buffer space and switch complexity. We also expect a reduction in scheduling delays.

We have shown that the AS table-based arbiter is not appropriate for QoS. This is due to the fact that each table entry enables one packet to leave, whatever its size. In this way, the table arbiter penalizes traffic with small packet size (e.g. audio traffic).

On the other hand, AS proposes the MinBW arbiter. In this case, the AS specification only states what is the appropriate behavior for this kind of arbiter, but does not define any precise implementation. However, AS specification states that several scheduling algorithms exhibit the desired properties of the MinBW scheduler. We have used a modified version of the WFQ algorithm as an implementation of the MinBW scheduler because we think it is the best option among those subjected by the specification. We have shown that this scheduler is able to provide bandwidth guarantees even to the SCs with less priority. Furthermore, it is able to work perfectly well even with packets of different size.

Finally, the performance that our proposal achieves is very similar to the one obtained with the more complex MinBW option. It is adequate for both multimedia and best-effort traffic in terms of latency, jitter, and throughput. Specifically, in the scenario that we present in this paper, our proposal uses 1/4 the buffer space. Moreover, although in this paper we have not considered the simplification in the implementation of the arbiter, we expect that it would provide a speed-up of 2.0 for the scheduling delay, which would improve the results that we have shown.

References

[1] Advanced Switching Interconnect Special Interest Group. <http://www.asi-sig.org>.

[2] Advanced Switching Interconnect Special Interest Group. Advanced Switching Core Architecture Specification. Revision 1.1, March 2005.

[3] F.J. Alfaro, J.L. Sánchez, J. Duato, QoS in InfiniBand subnetworks, IEEE Transactions on Parallel Distributed Systems 15 (9) (2004) 810–823.

[4] N. Chrysos, M. Katevenis, Multiple priorities in a two-lane buffered crossbar, in: Proceedings of the IEEE Globecom 2004 Conference, November 2004, CR-ROM paper ID "GE15-3".

[5] W. Dally, P. Carvey, L. Dennison, Architecture of the Avici terabit switch/router, in: Proceedings of the 6th Symposium on Hot Interconnects, 1998, pp. 41–50.

[6] A. Demers, S. Keshav, S. Shenker, Analysis and simulation of a fair queueing algorithm, Journal of Internetworking Research and Experience (1990) 3–26.

[7] J. Duato, S. Yalamanchili, M.B. Caminero, D. Love, F.J. Quiles, MMR: a high-performance multimedia router, Architecture and design trade-offs, in: Proceedings of the 11th Symposium on High Performance Computer Architecture (HPCA), January 1999.

[8] I. Elhanany, D. Chiou, V. Tabatabaee, R. Noro, A. Poursepanj, The network processing forum switch fabric benchmark specifications: an overview, IEEE Network (2005) 5–9.

[9] S.J. Golestani, A self-clocked fair queueing scheme for broadband applications, in: Proceedings of IEEE INFOCOM, 1994, pp. 636–646.

[10] IEEE. 802.1D-2004: Standard for local and metropolitan area networks, 2004. <http://grouper.ieee.org/groups/802/1/>.

[11] InfiniBand Trade Association. InfiniBand architecture specification volume 1. Release 1.0, October 2000.

[12] R. Jain, The Art of Computer System Performance Analysis Techniques for Experimental Design Measurement, Simulation and Modeling, John Wiley and Sons, Inc., 1991.

[13] M. Katevenis, S. Sidiropoulos, C. Courcoubetis, Weighted round-robin cell multiplexing in a general-purpose ATM switch, IEEE Journal on Selected Areas in Communication 20 (1991) 1265–1279.

[14] M. Katevenis, P. Vatsolaki, D. Serpanos, E. Markatos, ATLAS I: a single-chip ATM switch for NOWs, in: Proceedings of the Workshop on Communication and Architectural Support for Network-based Parallel Computing (CANPC 97), San Antonio, Texas, USA, 1997.

[15] A. Martínez, F.J. Alfaro, J.L. Sánchez, J. Duato, Providing full QoS support in clusters using only two VCs at the switches, in: Proceedings of the 12th International Conference on High Performance Computing (HiPC), pp. 158–169.

[16] R. Martínez, F.J. Alfaro, J.L. Sánchez. Implementing the Advanced Switching minimum bandwidth egress link scheduler, to be presented in IEEE International Symposium on Network Computing and Applications (IEEE NCA06), July 2006.

[17] R. Martínez, F.J. Alfaro, J.L. Sánchez. Providing quality of service over Advanced Switching, to be presented in International Conference on Parallel and Distributed Systems (ICPADS), July 2006.

[18] C. Minkenberg, F. Abel, M. Gusat, R.P. Luijten, W. Denzel. Current issues in packet switch design, in: ACM SIGCOMM Computer Communication Review, vol. 33, January 2003, pp. 119–124,.

[19] A.K. Parekh, R.G. Gallager, A generalized processor sharing approach to flow control in integrated services networks the single-node case, IEEE/ACM Transactions on Networking 1 (3) (1993) 344–357.

[20] A.K. Parekh, R.G. Gallagher, A generalized processor sharing approach to flow control in integrated services networks: the multiple node case, IEEE/ACM Transactions on Networking 2 (2) (1994) 137–150.

[21] PCI Special Interest Group. PCI Express Base Architecture Specification. Revision 1.0a, April 2003.

[22] L. Peh, W. Dally, A delay model and speculative architecture for pipelined routers, in: Proceedings of the 7th International Symposium on High-Performance Computer Architecture, IEEE Computer Society, 2001, p. 255.

[23] M. Shreedhar, G. Varghese, Efficient fair queueing using deficit round robin, IEEE/ACM Transactions on Networking 4 (3) (1996) 375–385.

[24] D. Simos. Design of a 32×32 variable-packet-size buffered crossbar switch chip. Technical Report FORTH-ICS/TR-339, Inst. of Computer Science, Forth, July 2004.
[25] D. Stiliadis, A. Varma, Latency-rate servers: a general model for analysis of traffic scheduling algorithms, IEEE/ACM Transactions on Networking 6 (5) (1998) 611–624.
[26] K.H. Yum, E.J. Kim, C.R. Das, A.S. Vaidya, MediaWorm: a QoS capable router architecture for clusters, IEEE Transactions on Parallel Distributed Systems 13 (12) (2002) 1261–1274.
[27] K.H. Yum, E.J. Kim, C.R. Das. QoS provisioning in clusters: an investigation of router and NIC design, in: Proceedings of the 28th Annual International Symposium on Computer Architecture (ISCA), IEEE Computer Society, July 2001.
[28] L. Zhang, Virtual clock: a new traffic control algorithm for packet switched networks, ACM Transaction on Computer Systems 9 (2) (1991) 101–124.