Full QoS Support with 2 VCs for Single-chip Switches*

Alejandro Martínez, Francisco J. Alfaro, José L. Sánchez
Escuela Politécnica Superior, Univ. Castilla-La Mancha
{alejandro,falfaro,jsanchez}@info-ab.uclm.es

José Duato
DISCA, Univ. Politécnica Valencia
jduato@disca.upv.es

Abstract

Current interconnection standards providing hardware support for quality of service (QoS) consider up to 16 virtual channels (VCs) for this purpose. However, most implementations do not offer so many because VCs increase the complexity of the switch and the scheduling delays. We have shown that this number of VCs can be significantly reduced, because it is enough to use two VCs for QoS purposes at each switch port. In this paper, we explore two alternative switch designs that take advantage of this reduction.

1 Introduction

Clusters of PCs have emerged as a cost-effective platform to implement Internet services and applications. These clusters provide service to thousands or tens of thousands of concurrent users. Many of these Internet applications are multimedia applications, which usually present bandwidth and/or latency requirements [10]. These are known as QoS requirements.

In the next section, we will be looking at some of the proposals to provide QoS in clusters. Most of them incorporate 16 or even more VCs, devoting a different VC to each traffic class. This increases the switch complexity and required silicon area and also prevents the use of these VCs for other purposes. Moreover, it seems that, when the technology enables it, the trend is to increase the number of ports instead of increasing the number of VCs per port [9]. This, in fact, is the trend followed nowadays by companies in their new products launched to the market.

In [8], we have proposed a strategy to use just two VCs at each switch port for the provision of QoS that obtains similar results as if we were using many more VCs. This is achieved by respecting at the switches some of the scheduling decisions made at network interfaces. In this paper, we thoroughly explore two alternative switch designs that take advantage of this reduction and we evaluate them with different traffic models. Simulation results show that our proposal provides a very similar performance compared with a traditional architecture with many more VCs both for the QoS traffic and for the best-effort traffic.

The remainder of this paper is structured as follows. In the following section, the related work is presented. In Section 3 we review our strategy to reduce the number of VCs required for QoS support. Details on the experimental platform and the performance evaluation are presented in Section 4. Finally, Section 5 summarizes this study.

2 Related work

During the last decade, several cluster switches with QoS support have been proposed. All of them incorporate VCs in order to provide QoS support. Among the most recent ones are the industry standards InfiniBand and PCI Express Advanced Switching (AS). InfiniBand [6] can support up to 16 VCs. On the other hand, AS architecture [1] incorporates up to 20 VCs. If a great number of VCs is implemented, it would require a significant fraction of silicon area and would make packet processing a more time-consuming task.

On the other hand, there have been proposals which use only two VCs. For instance, the Avici TSR [2] is a well-known example of this. It is able to segregate premium traffic from regular traffic. However, it is limited to this classification and cannot differentiate among more categories. In the recent IEEE standards, it is recommended to consider seven traffic classes [5]. So, although being able to differentiate two categories is a big improvement, it could be insufficient.

In contrast, the novelty of our proposal lies in that although we use only two VCs at switches, the global behavior of the network is very similar as if the switches were using much more VCs. This is because we are reusing at the switch ports the scheduling decisions performed at the network interfaces, which have as many VCs as traffic classes (8 VCs in our performance evaluation).

*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 grant.

3 Full QoS Support with only two VCs

In [8], we have proposed a new strategy to use only two VCs at each switch port to provide QoS which achieves similar performance results to those using many more VCs. We review this proposal in this section.

The key idea of our proposal is based on this observation: Assuming that the links are not oversubscribed, all the traffic flows through the switches seamlessly. Therefore, the basic idea of our proposal consists in using only two VCs at the switch ports. One of these VCs is used for QoS packets and the other for best-effort packets. Moreover, we propose to use a connection admission control (CAC) to guarantee that QoS traffic will not oversubscribe the links and we give QoS traffic absolute priority over best-effort traffic, which is not subject to the CAC.

The scheduler we propose considers the original priority of the packets at the head of the queues, instead of just whether they are regulated traffic or not. This is not very complex because very efficient priority encoder circuits have been proposed [4]. Note that this cannot lead to starvation on the regulated traffic because the CAC assures that there is enough bandwidth for all the regulated flows. Thereby, by using this scheduler, the switches achieve some reutilization of the scheduling decisions made at network interfaces. This is because the order of the incoming messages is respected, but at the same time, the switches merge the flows to produce a correct order at the output ports.

The main advantage of our proposal is the reduction of VCs needed at the switches. Usually, when a port has several VCs, its memory is statically partioned among the VCs. Therefore, if several traffic classes share the same VC, a more efficient use of the buffer space can be achieved, because this space can be shared. This has two potential benefits: Allowing a reduction in the total buffer space implemented or providing the same space but being more flexible in its assignation (no static partition among traffic classes), leading to more tolerance of bursty traffic.

A drawback of our technique is the switches not being able to reschedule the traffic as freely as if a different VC for each traffic class were implemented. In that case, the switches have the ability to push forward high-priority packets in detriment of packets with less priority at the same input port. Using our technique, it is only possible to do so with packets from different input ports.

Let us analyze the consequences of this handicap: It may happen that when no more high-priority packets are available, a low-priority packet is transmitted. We will assume that both packets are regulated traffic and, thus, will share the same VC. If the low-priority packet has to wait in a switch input queue, and other packets with higher priority are transmitted 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 has a small impact on performance because there is bandwidth reservation for these packets. This means that all the QoS packets will flow with short delay.

On the other hand, the best-effort traffic classes only receive coarse-grain QoS, since they are not regulated. However, the interfaces are still able to assign the available bandwidth to the highest priority best-effort traffic classes and, therefore, some differentiation is achieved among them. If stricter guarantees were needed by a particular flow, it should be classified as QoS traffic. Therefore, although best-effort traffic can obtain a better performance using more VCs, the results do not justify the higher expenses.

Summing up, our proposal consists in reducing the number of VCs at each switch port needed to provide flows with QoS. Instead of having a VC per traffic class, we propose to use only two VCs at switches: One for QoS packets and another for best-effort packets. In order for this strategy to work, we guarantee that there is no link oversubscription for QoS traffic by using a CAC strategy.

4 Performance Evaluation

In this section, we show the behavior of our proposal. We have performed the tests considering four cases. First, we have tested the performance of our proposal, which uses 2 VCs at each switch port. It is referred to in the figures as *New 2 VCs*. Note that, with our proposal, the network interfaces still use 8 VCs. We have considered two variants of this proposal, which are noted *New 2 VCs-P* (16 ports) and *New 2 VCs-B* (8 ports, but larger buffers per VC).

We have also performed tests with switches using 8 VCs. In this case, it is referred to in the figures as *Traditional 8 VCs*. Finally, we have also tested a traditional approach with 2 VCs, noted in the figures as *Traditional 2 VCs*. In this case, the network interfaces also use 2 VCs. *Traditional 2 VCs* switches have the same number of ports than *Traditional 8 VCs*, but they have 4 times more buffer space per VC, allowing a good performance with bursty traffic. The four switch models studied take equivallent sillicon area.

The network used to test our proposal is a butterfly multistage with 128 end-points. The switches use a combined input and output buffer architecture, with a crossbar to connect the buffers. The CAC we have implemented is a simple one, based on average bandwidth. Each connection is assigned a path where enough resources are assured. No packets are dropped because we use credit-based flow control. Finally, there is a speed-up of 2.0 in the crossbar of the switch. Note that this is internal speed-up (the external links keep their throughput) and is achieved by doubling the word size in the access to the memories and at the crossbar.

In Table 1, the characteristics of the modeled traffic are included. We have considered the traffic classes (TCs) defined by the IEEE standard 802.1D-2004 [5] at the Annex G, which are generally accepted for interconnection networks. However, we have added an eighth TC, *Preferential Best-effort*, with a priority between *Excellent-effort* and *Best-effort*. In this way, the workload is composed of 8 different TCs: Four QoS TCs and four best-effort TCs. Each TC has decreasing priority, such that TC 7 has the highest priority and TC 0 has the lowest.

Table 1. Traffic injected per host.

TC	Name	% BW	Packet size	Notes
7	Network Control	1	[64,512] B	self-similar
6	Audio	16.333	128 B	64 KB/s conns.
5	Video	16.333	[64,2048] B	MPEG-4 traces
4	Controlled Load	16.333	[64,2048] B	1 MB/s conns.
3	Excellent-effort	12.5	[64,2048] B	self-similar
2	Pref. Best-effort	12.5	[64,2048] B	self-similar
1	Best-effort	12.5	[64,2048] B	self-similar
0	Background	12.5	[64,2048] B	self-similar

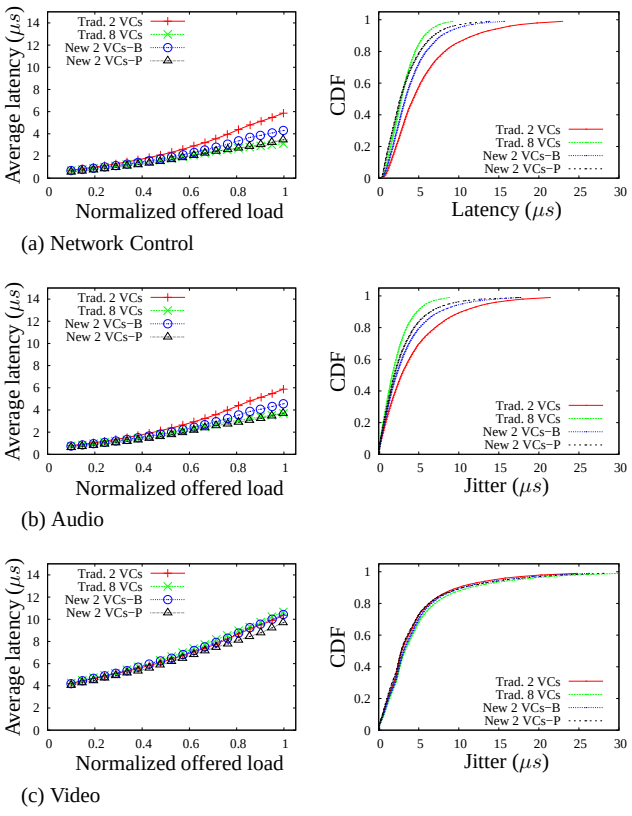
The destination pattern is based on Zipf's law, as recommended in [3]. In this way, the traffic is not uniformly distributed. The packets are generated according to different distributions, as can be seen in Table 1. Audio, video, and controlled load traffic are composed of point-to-point connections of the given bandwidth. The self-similar traffic is bursty traffic generated with on/off sources, governed by Pareto distributions, as recommended in [7].

4.1 Simulation results

In this section, the performance of our proposal will be shown. We will consider three traditional QoS indices for this performance evaluation: Throughput, latency, and jitter. Inappropriate results of latency or jitter may lead to dropped packets at the application level. For that reason, we also show the cumulative distribution function (CDF) of latency and jitter, which represents the probability of a packet achieving a latency or jitter equal to or lower than a certain value.

Figure 1 shows the performance of QoS traffic. In this case, the best average latency corresponds to our proposal *New 2 VCs-P*, mostly because the switches in this interconnection network have more ports, which in turn leads to a two-stage MIN, one stage less than the other alternatives.

On the other hand, in Figure 1 we can also see the CDF of latency and jitter at a normalized network load of 1.0. Jitter for Network Control traffic has not sense, while CDF of latency is not as interesting as jitter for Audio and Video traffic. The *Traditional 8 VCs* case offers the best results. With our proposal, a small portion of the packets see their



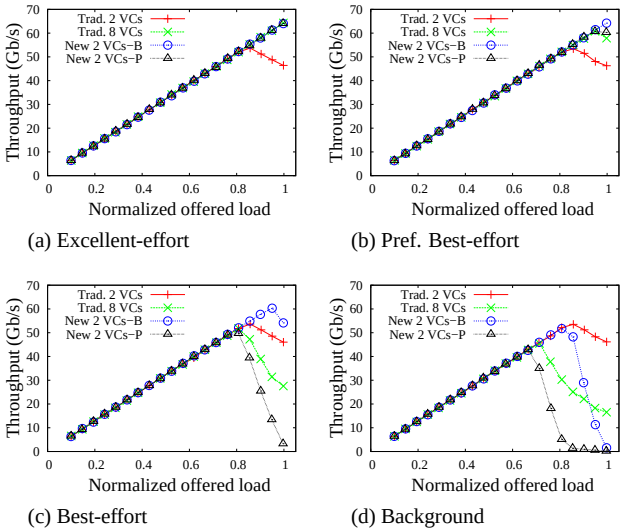
(a) Network Control

(b) Audio

(c) Video

Figure 1. Results for QoS traffic.

latency increased, and thus, the jitter. This is the price to pay for this reduction of VCs, although these maximum values are safely inside the allotted margins (10 and 100 ms for audio and video, respectively [5]).



(a) Excellent-effort

(b) Pref. Best-effort

(c) Best-effort

(d) Background

Figure 2. Throughput of BE traffic.

In Figure 2, we evaluate the best-effort traffic. In this case, the *Traditional 2 VCs* approach produces the same performance for all the TCs, which is an inadequate behavior, because excellent-effort and preferential best-effort traffic classes should have better performance. The reason for this inadequate behavior is that all the best-effort classes look the same for the schedulers at both the network interfaces and switches in the *Traditional 2 VCs* model.

On the other hand, the arbiters using our technique take into account the priority of the packets, although they share the same VC. For that reason, our proposals, which devote a single VC at the switches for all the best-effort TCs, can provide a behavior similar to that of the *Traditional 8 VCs* approach, which uses 4 VCs for the best-effort TCs.

The *New 2 VCs-B* offers the best performance because it provides the maximum flexibility in the use of the buffer space, while in the *Traditional 8 VCs* case the same amount of memory is statically partitioned. On the other hand, the *New 2 VCs-P* case offers a worse performance because the buffer space is smaller.

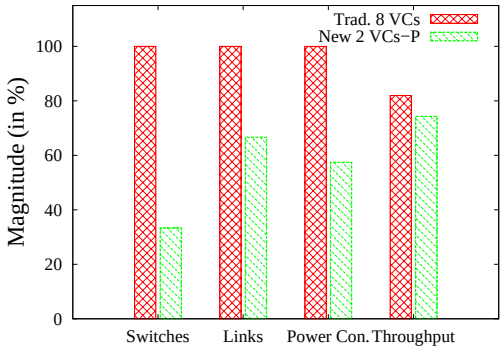


Figure 3. Summary of the trade-offs of the *New 2 VCs-P* proposal, compared with the *Traditional 8 VCs* case.

Figure 3 summarizes the trade-offs of using our *New 2 VCs-P* proposal. As can be seen, there is a very noticeable reduction in chip[1] and link counts and, therefore, in the associated power consumption of the interconnection network (calculated from estimates of the ASIC chip design). On the other hand, the global throughput achieved with bursty traffic is slightly lower (only a 10% reduction), but the reduction on the component count would lead to a much more cost-effective solution.

According to these results, we can conclude that our proposals can provide an adequate QoS performance. The reduction of the number of VCs to just two allows us to offer two alternative switch designs: The first keeps the expenses but produces a higher global throughput by using more flexible buffers, and the second greatly decreases the cost and power-consumption of the interconnection with a small degradation in performance (13% global throughput less than *Traditional 2 VCs* with non-uniform traffic).

[1]Remember that both *New 2 VCs-P* and *Traditional 8 VCs* switches take equivalent silicon area.

5 Conclusions

In [8], we presented a proposal to use only two VCs at each switch port to provide QoS support. This paper offers two novel designs for the switch that benefit from that proposal. The first design increases the global performance of the network against unbalanced traffic by using buffer space in a flexible way. The second design greatly reduces the component count, and thus, the cost and power-consumption of the interconnection, at the cost of a small degradation of performance.

References

[1] Advanced switching core architecture specification. Technical report, available online at `http://www.asi-sig.org/specifications` for ASI SIG members.

[2] W. Dally, P. Carvey, and L. Dennison. Architecture of the Avici terabit switch/router. In *Proceedings of the 6th Symposium on Hot Interconnects*, 1998.

[3] I. Elhanany, D. Chiou, V. Tabatabaee, R. Noro, and A. Poursepanj. The network processing forum switch fabric benchmark specifications: An overview. *IEEE Network*, Mar. 2005.

[4] C. Huang, J. Wang, and Y. Huang. Design of high-performance CMOS priority encoders and incrementer/decrementers using multilevel lookahead and multilevel folding techniques. *IEEE Journal of Solid-State Circuits*, 1(37):63–76, Jan. 2002.

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

[6] InfiniBand Trade Association. *InfiniBand architecture specification volume 1. Release 1.0*, Oct. 2000.

[7] R. Jain. *The art of computer system performance analysis: techniques for experimental design, measurement, simulation and modeling*. John Wiley and Sons, Inc., 1991.

[8] A. Martínez, F. J. Alfaro, J. L. Sánchez, and 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)*, Dec. 2005. Available at `http://www.i3a.uclm.es/documentos/2/congresos/Congreso2_134_HiPC05.pdf`.

[9] C. Minkenberg, F. Abel, M. Gusat, R. P. Luijten, and W. Denzel. Current issues in packet switch design. In *ACM SIGCOMM Computer Communication Review*, Jan. 2003.

[10] D. Miras. A survey on network QoS needs of advanced internet applications. Technical report, Internet2 - QoS Working Group, 2002.