A First Approach to Provide QoS in Advanced Switching*

R. Martínez, F.J. Alfaro, J.L. Sánchez
Dept. de Sistemas Informáticos
Escuela Politécnica Superior
Universidad de Castilla-La Mancha
02071 - Albacete, Spain
{raulmm, falfaro, jsanchez}@info-ab.uclm.es

Tor Skeie
Simula Research Laboratory
P.O.Box 134, N-1325 Lysaker, Norway
tskeie@simula.no

Abstract

Nowadays, the use of multimedia applications that present QoS requirements is increasing rapidly. Advanced Switching (AS) is a new interconnection network technology that expands the capabilities of PCI Express. AS provides mechanisms that can be used to support QoS. Specifically, an AS fabric permits us to employ virtual channels, egress link scheduling, and an admission control mechanism to differentiate between traffic flows. In this paper we examine these mechanisms and show how to provide QoS based on bandwidth and latency requirements. Furthermore, a new algorithm, Weighted Fair Queuing Credit Aware, is proposed as a specific implementation of one of the schedulers suggested by the AS specification.

1 Introduction

Advanced Switching (AS) is a new open-standard fabric-interconnect technology for communications, storage, and embedded environments built on the same physical and link layers as PCI Express technology. The physical layer consists in a dual-simplex channel, which is implemented as a transmit pair and a receive pair. The link layer is responsible for data integrity. A credit-based flow control protocol ensures that packets are only transmitted when there is enough buffer space at the other end to store them. Moreover, AS includes an optimized transaction layer to enable essential communication capabilities, including protocol encapsulation, enhanced fail-over, high availability, congestion and system management, and mechanisms for providing QoS. AS specification v1.1 was published in March 2005 [1].

AS supports up to 16 unicast VCs, where one of them is the Fabric Management Channel (FMC). AS defines two egress link schedulers to resolve between these VCs: The VC arbitration table scheduler and the minimum bandwidth egress link (MinBW) scheduler. When implementing the egress link scheduler the interaction with the credit-based flow control must be taken into account. Thus, if the credits for a given VC have been exhausted, the VC scheduler must treat the corresponding queue as if it were empty.

The table scheduler provides an implementation of the Weighted Round Robin (WRR) algorithm [7]. Each VC arbitration table entry corresponds to a slot of a WRR arbitration period. Each entry contains a VC identifier value. When arbitration is needed, the table is cycled through sequentially and a whole packet is transmitted from the VC indicated in the current table entry, regardless of the packet's size.

The MinBW scheduler consists of two parts. The first is a mechanism to provide the FMC with absolute priority. The second is a mechanism to distribute bandwidth amongst the rest of the VCs according to a configured set of weights. AS does not specify an algorithm or implementation for the MinBW scheduler, but it must respect certain properties [1]: *Work conserving*, *bandwidth metering*, *no packet metering*, *minimum bandwidth guarantee*, *fair redistribution of unused bandwidth*, and *memoryless*.

Moreover, fabric management software may regulate access to the AS fabric, allowing new packet flows entry to the fabric only when sufficient resources are available.

*This work was partly supported by the Spanish CICYT under Grant TIC2003-08154-C06, by the Junta de Comunidades de Castilla-La Mancha under Grant PBC-05-005-1, and by the Spanish State Secretariat of Education and Universities under FPU grant.

2 Providing QoS over AS

As was stated in the previous section, AS provides several mechanisms that can be used to provide flows with QoS. However, AS specification does not indicate how to use these mechanisms. In this section, we propose 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. To define the SCs we are going to use a traffic classification based on two network parameters: Mean bandwidth and maximum latency. We distinguish between three broad categories of traffic types:

- Network Control traffic: High-priority traffic to maintain and support the network infrastructure. One SC will be dedicated to this kind of traffic.
- QoS traffic: This traffic has explicit minimum bandwidth and/or maximum latency requirements. Various QoS SCs can be defined with different bandwidth and latency requirements.
- Best-effort traffic: This traffic accounts for the majority of traffic handled by data communication networks today. Best-effort SCs are only characterized by the differing priority among each other.

When various flows obtain access to the AS fabric they will be classified into the various SCs depending on their characteristics. If there is a sufficient number of VCs we will devote a separate VC to each existing SC. However, if this is not the case, some SCs must share the same VC. The schedulers must be properly configured in the different network elements to provide a differentiated treatment for the different SCs.

The network control SC will be assigned to the FMC in order to achieve the maximum priority when using the MinBW scheduler. In the case of the table scheduler, the FMC is processed in the same way as the other VCs so we will consider the control SC as a QoS SC with high latency requirements.

Best-effort SCs will be assigned only with a small amount of bandwidth proportional to their relative priority. The rest of the bandwidth to be dedicated to best-effort traffic will be assigned to the control SC. This SC will be assigned a bandwidth equal at least to the sum of this best-effort bandwidth, the expected amount of control traffic, and the expected amount of bandwidth that the network is not able to provide (it is not usually possible to achieve a 100% of global throughput). Note that the bandwidth left by the control traffic itself would be redistributed among the other VCs, specifically, best-effort SCs. The bandwidth assigned to the control SC in the case of the table will be left unassigned when using the MinBW scheduler. The remaining resources will be distributed among QoS SCs in accordance with their requirements.

2.1 Providing QoS requirements with the table scheduler

In [3] we explained how to configure this kind of arbitration table (in that case for InfiniBand) 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 egress link bandwidth. To provide maximum latency requirements to the traffic of a VC, the maximum time must be studied that a packet can spend crossing a network element as well as the time it takes to be transmitted to the next element once it has been chosen by the scheduler. With this information it is possible to control the maximum latency of a network element crossing, by fixing the maximum separation between two consecutive table entries devoted to the VC in question.

2.2 Providing QoS requirements with the MinBW scheduler

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 [4] exhibit the desired properties of the MinBW scheduler. Parekh and Gallager [8] analyzed the performance of WFQ from the standpoint of worst-case packet delay. On the basis of this study, we assign a higher amount of bandwidth than is needed to those VCs with high latency requirements, in order to obtain an appropriate average and maximum latency performance.

2.2.1 A new proposal for the MinBW scheduler algorithm: WFQCA

In accordance with the AS specification [1], several well-known scheduling algorithms exhibit the desired properties of the MinBW scheduler. Examples include variants of Weighted Fair Queuing (WFQ) [4] such as Self-Clocked WFQ [5], and variants of Weighted Round Robin (WRR) [7] such as Deficit Round Robin [10]. 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 [11]. Most WFQ variants, such as Self-Clocked WFQ, are different approaches to WFQ in order to reduce its complexity. Therefore, we have chosen to use the original WFQ algorithm, because in our view it is the best option, among those proposed by the specification, to make performance comparisons.

WFQ [4] is a work-conserving algorithm that distributes the bandwidth among various flows according to a configurable set of weights. The original WFQ algorithm tracks the set of flows, in our case VCs, which are active in each instant. A VC is considered active if it has a packet to be transmitted. The set of active VCs is used to compute the *virtual finishing time* in which a packet would have been completely transmitted in the corresponding GPS [8] system. When a packet arrives at the output queues it is stamped with its virtual finishing time. Packets are transmitted in an increasing order of timestamp.

The use of this algorithm in the AS environment faces two problems. The first problem is that the amount of flow control credits is not considered to determine the active set of VCs. The second 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 (WFQCA). The WFQCA 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, therefore, can be implemented in this new technology.

2.3 Final considerations

To provide QoS guarantee, an admission control (AC) must be used. Without AC 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, no admission control would be implemented for network control traffic and best-effort traffic.

There are two possible ways of configuring the schedulers. The first possibility is to configure the schedulers in advance, defining a set of SCs with a different minimum bandwidth and maximum latency reservation [9]. This distribution would be made taking into account the expected use of each SC. The second possibility is to configure the schedulers in accordance with the connection requirements in a dynamic way. With this approach the scheduler configuration may be modified both when a new connection is accepted and when a previously established connection ends [2]. This allows us more flexibility and a more accurate use of the resources. Note that the second possibility is only feasible when a AC is used.

3 Performance Evaluation

In this section we evaluate the behavior of our proposals by simulation. We have used a perfect-shuffle multi-stage interconnection network (MIN) with 64 end-points. The switch model has 8 ports and uses a

combined input and output buffer architecture, with a crossbar to connect the buffers. In our tests, the link bandwidth is 2.5 Gb/s but, with the 8b/10b encoding scheme, the maximum effective bandwidth for data traffic is only 2 Gb/s.

The IEEE standard 802.1D-2004 [6] defines seven traffic types at the Annex G, which are particularly appropriate for this study. We will consider each traffic type as a SC. Table 1 shows each SC and its requirements. In this way, the workload is composed of 7 SCs and each one of them will be assigned to a different VC. The NC SC is assigned to the FMC.

Table 1: SCs suggested by the standard IEEE 802.1D-2004.

Type	**SC**	**Description**
Control	Network control (NC)	Traffic to support the network infrastructure.
QoS	Voice (VO)	Traffic with a limit of 10 ms for latency and jitter.
QoS	Video (VI)	Traffic with a limit of 100 ms for latency and jitter.
QoS	Controlled load (CL)	Traffic with explicit bandwidth requirements.
Best-effort	Excellent-effort (EE)	Preferential best-effort traffic.
Best-effort	Best-effort (BE)	LAN traffic as we know it today.
Best-effort	Background (BK)	Traffic that should not impact other flows.

Our intention is to show that, with a AC that controls the QoS traffic workload, the QoS requirements of the different SCs are met, whatever the load of best-effort traffic. Table 2 shows the proportion of traffic that each node injects regarding the link bandwidth (2 Gb/s). In these simulations we have used a fixed packet size of 128 bytes due to the problems of the table scheduler to deal with variable packet size. The destination pattern is uniform in order to fully load the network.

We have performed the tests considering two cases: The table scheduler and the MinBW scheduler implemented using the WFQCA. Table 2 shows the two scheduler configurations.

Table 2: Injected traffic and scheduler configuration.

	Injected traffic			**Table Conf.**		**MinBW Conf.**
SC	**Min. %**	**Max. %**	**Traffic pattern**	**# entr.**	**Dist.**	**Weight**
NC	1	1	self-similar	16	4	-
VO	18.75	18.75	64Kb/s CBR connections	16	4	0.25
VI	18.75	18.75	750 Kb/s MPEG-4 traces	12	6	0.1875
CL	18.75	18.75	750 Kb/s CBR connections	12	-	0.1875
EE	0	29.25	self-similar	5	-	0.078125
BE	0	29.25	self-similar	2	-	0.03125
BK	0	29.25	self-similar	1	-	0.015625
	57.25	145		64		0.75

3.1 Simulation results

Figures 1 and 2 show the performance of the table and the MinBW schedulers, respectively. For a given input load, several simulations have been conducted, and the average values and the confidence intervals at a 90% confidence level are shown in the figures.

Results show that the MinBW scheduler using the WFQCA provides a slightly better performance than the table scheduler, but the table scheduler also provides a good performance. In both cases, control and QoS SCs obtain all the bandwidth that they inject. However, when the network load is very high the best-effort SCs do not yield a corresponding result. These SCs obtain a bandwidth proportional to their priority. Moreover, we can see that the maximum throughput of the network is around 90%.

Results also show that, in both cases, control and QoS SCs obtain a low average latency regardless of network load. The average latency of best-effort SCs grows with the load. Furthermore, it can be seen that best-effort SCs obtain different average latency according to their different priority. Both schedulers fulfill the maximum latency requirements recommended by the IEEE [6], 10 ms for voice and 100 ms for video.

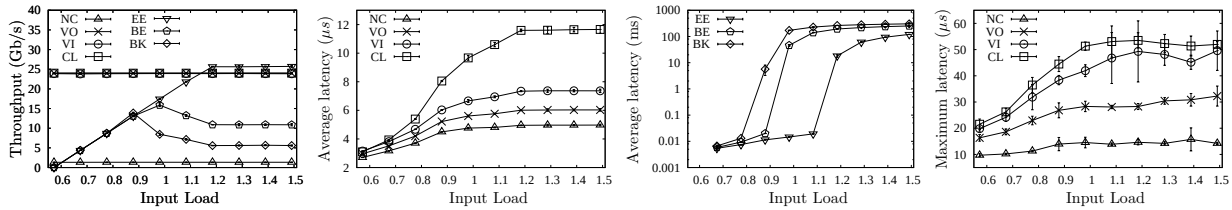

Figure 1: Table scheduler performance.

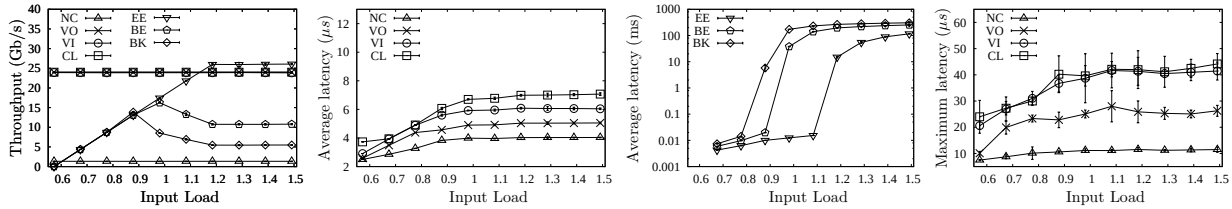

Figure 2: MinBW scheduler performance.

4 Conclusions

In this paper, we have proposed several methods of using the AS mechanisms to provide applications with QoS based on bandwidth and latency requirements. Specifically, we have proposed the WFQCA algorithm as a concrete implementation for the MinBW scheduler that accomplishes the AS specification. The results obtained show that, with fixed packet size, the two egress link schedulers defined by AS, the table-based and the MinBW schedulers, are able to provide the different traffic types with their requirements: Network control traffic obtains a good latency; the flows with bandwidth requirements obtain the amount that they need; the flows with latency requirements do not exceed the maximum allowed; finally, best-effort flows obtain a different amount of bandwidth and latency performance in accordance with their different priorities.

Finally, our view is that the table scheduler is a very simple mechanism, which has positive properties. For this reason, as future work we are focusing our attention upon several aspects to improve the AS arbitration table behavior, the first and foremost of which will be to solve the problem that arises when variable packet size is considered.

References

[1] Advanced Switching Interconnect Special Interest Group. *Advanced Switching core architecture specification. Revision 1.1*, March 2005.

[2] F. J. Alfaro, J. L. Sánchez, and J. Duato. A new proposal to fill in the InfiniBand arbitration tables. In *Proceedings of IEEE International Conference on Parallel Computing (ICPP'03)*, October 2003.

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

[4] A. Demers, S. Keshav, and S. Shenker. Analysis and simulations of a fair queuing algorithm. In *SIGCOMM*, 1989.

[5] S. J. Golestani. A self-clocked fair queueing scheme for broadband applications. In *INFOCOM*, 1994.

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

[7] M. Katevenis, S. Sidiropoulos, and C. Corcoubetis. Weighted round-robin cell multiplexing in a general-purpose ATM switch chip. *IEEE J. Select. Areas Commun.*, October 1991.

[8] A. K. Parekh and 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):344–357, 1993.

[9] S.A. Reinemo, F.O. Sem-Jacobsen, T. Skeie, and O. Lysne. Admission control for diffserv based quality of service in cut-through networks. In *In Proceedings of the 10th International Conference on High Performance Computing (HiPC 2003). Hyderabad, India*, pages 118–128, December 2003.

[10] M. Shreedhar and G. Varghese. Efficient fair queueing using deficit round robin. In *SIGCOMM*, 1995.

[11] D. Stiliadis and A. Varma. Latency-rate servers: a general model for analysis of traffic scheduling algorithms. *IEEE/ACM Trans. Netw.*, 6(5):611–624, 1998.