DISTRIBUTED CELL SCHEDULING ALGORITHMS FOR VIRTUAL-OUTPUT-QUEUED SWITCHES

Rainer Schoenen and Roman Hying
Institute for Integrated Signal Processing Systems, Aachen University of Technology
schoenen@ert.rwth-aachen.de, http://www.iss.rwth-aachen.de/

Abstract

Input buffered switches most efficiently use memory and crossbar bandwidth. Virtual Output Queueing (VOQ) is required to circumvent the head-of-line blocking limiting the throughput to 58.6%. For the slotted access control to the switch fabric, weighted arbitration algorithms achieve 100% throughput with lowest delays under all admissible traffic. Following the arbitration decision, distributed QoS-aware cell schedulers decide locally in each input port upon the next cell to forward. In this paper we treat cell scheduling algorithms employed in VOQ switches, in contrast to output queueing (OQ). We show that typical scheduling properties also hold under the VOQ architecture and give representative quantitative performance results.

1 Introduction

Very high speed switches are needed for future ATM and IPv6 networks. Input queued switches are most powerful because the access rate of crossbar and buffer memory is not higher than the line rate of the connected links. For the classical FIFO queue organization it is known that due to head-of-line blocking the maximum throughput is approximately 58.6% [4]. The Virtual Output Queueing (VOQ) can avoid blocking by bypassing cells destined for free output ports [15]. Each input manages a separate queue for each output (fig. 1). It has been shown that a throughput of 100% can then be achieved [13, 14, 16].

Arbitration algorithms are used to control the access of queues to the switch fabric by resolving the contention for the same output ports in each time slot. The achievable throughput and delay performance depends on the arbitration algorithm. Weighted algorithms [20] offer the best delay performance and they are required to operate well in other than symmetric load configurations. A global priority scheme is also managed in the arbiter [19].

After the arbitration decision in each input port a proper cell must be selected for transmission to the given output port. Unlike other papers assuming FCFS service it is necessary to provide cell scheduling in the input ports within each priority class. By doing so problems of QoS, flow separation and fairness can be addressed.

In this paper some popular cell scheduling algorithms are analysed and compared in VOQ and OQ configuration. We show that similar performance in both configurations can be expected for (weighted) round-robin and earliest-deadline-first schedulers. It is shown that moments and quantiles of the VOQ waiting time distribution are higher than in an OQ configuration, but the principal scheduling properties of QoS distinction work as well.

The paper is organized as follows. Section 2 discusses VOQ and arbitration algorithms. Selected cell scheduling algorithms are treated in section 3. In section 4 results for VOQ schedulers are given.

2 Arbitration in VOQ Switches

Figure 1 shows the VOQ configuration [15] which consists of M ports for input and output, a nonblocking switch fabric and an arbitration unit. Arriving cells on input port i are placed into the corresponding queue for their destination port o. Its current queue size is $q_{i,o}$. The process of arrivals to this queue is characterized by a mean rate $\lambda_{i,o}$. Let the input and output loads be

$$\rho_o^{out} = \sum_{i=0}^{M-1} \lambda_{i,o} T_{slot}; \qquad \rho_i^{in} = \sum_{o=0}^{M-1} \lambda_{i,o} T_{slot} \tag{1}$$

The load is admissible [15] if $\forall o : \rho_o^{out} < 1$ and $\forall i : \rho_i^{in} < 1$. In each time slot the arbiter selects unique pairs of input and output ports (a "match" (i,o)) based on weight information sent to it from the input ports.

For this bipartite graph matching problem [15] (fig 2) the optimum maximum size matching (MSM) or a maximum weight matching (MWM) [18] algorithms exist. With a weight chosen as $w_{i,o} = q_{i,o}$ the algorithm is called "longest queue first" (LQF) in [14], and it has been shown that 100% throughput can be achieved for all admissible i.i.d. arrivals [14] with MWM. Without weight information, as with MSM, instability and unfairness are likely and cell delays become very large for bursty or asymmetric load [20].

Because of the MWM and MSM complexity ($O(M^3 logM)$) a number of approximations have been

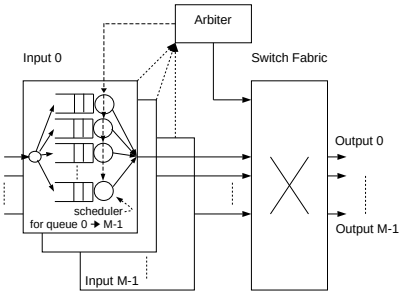

Figure 1: virtual output queueing

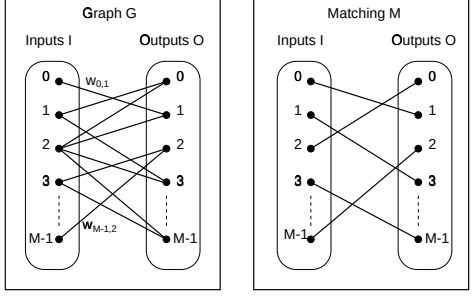

Figure 2: Bipartite graph matching

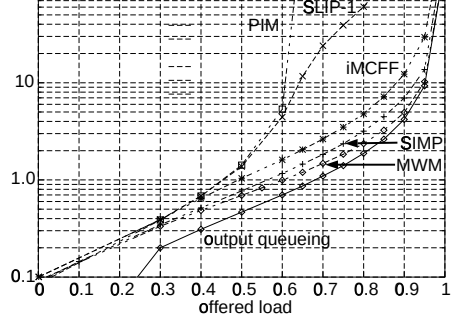

Figure 3: $E\{d\} = f(\rho)$ for VOQ

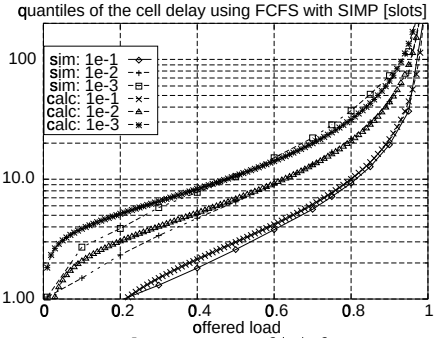

Figure 4: $d_{quantile(x)} = f(\rho)$ for SIMP

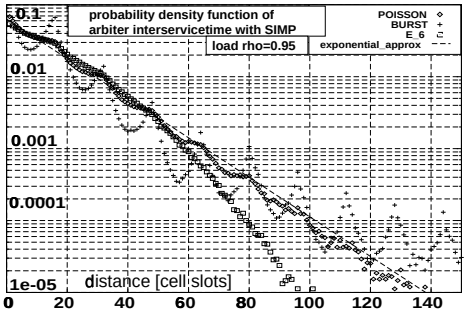

Figure 5: interservice time PDF for SIMP

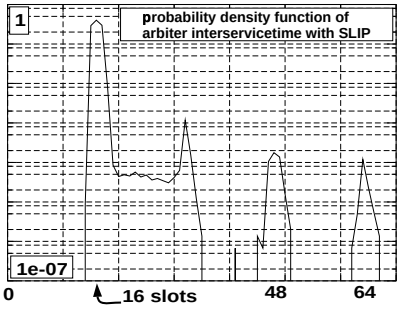

Figure 6: SLIP interservice PDF

proposed, such as iMCFF [16], iLQF [13] and SIMP [20]. MSM approximations exist with PIM [1], iSLIP [13] or WFA [22]. For our architecture, the SIMP algorithm is used, because its delay[1] performance is between the optimum MWM and iMCFF [16], as shown in figure 3. The algorithm is outlined here shortly:

	Algorithm „SIMP"
1	let I be an ordered list of all input ports and O the list of all output ports
2	let $I' \leftarrow I$ and $O' \leftarrow O$
3	choose the first output port o_c out of the ordered list O'
4	choose the input port i_c to match as one with $w_{i_c} = \max_{i \in I'}(w'_i)$, resolve ambiguities (same weights) in round-robin fashion
5	if $w_{i_c} > 0$ match i_c with o_c and let $I' \leftarrow I' \setminus i_c$ (set minus)
6	reduce the match space by letting $O' \leftarrow O' \setminus o_c$
7	repeat steps 3-6 until $O' = \emptyset$ (M repititions)
8	shift the list O cyclically before the next slot to achieve round-robin fairness between outputs
9	start the next time slot at step 2

Because of the complexity of the M^2-dimensional state space an analytical treatment is difficult. In the literature only unweighted algorithms are treated [13, 17], or the assumptions simplify it to an OQ model [6]. So far there has been no treatment of separate VOQ cell scheduling.

In figures 5 and 6 the time between successive service events is shown for different traffic and for SIMP or SLIP. Observe the periodicity in multiples of $M \cdot T_{slot}$, which is most dominant for SLIP. An $M/D/1$ model per port with an M-slow server ($\mu = T_{slot}^{-1} M^{-1}$) slightly underestimates the delay for SLIP. For lower load or smooth traffic with SIMP an exponential distribution looks very similar. However it is wrong to assume that an $M/M/1$ queueing system can describe any of the results. In the VOQ system there is no statistical independence between arrivals and service, thus it would be an overestimation by around a factor M. The dependence is exactly what we want weighted matching algorithms to achieve. There is less than a factor of two between OQ and VOQ cell delays, as well for mean (fig. 3) and for quantiles[2] (fig. 4). In section 4 an approximative analytic model is presented.

3 Cell Scheduling Algorithms

The reason for sophisticated cell scheduling algorithms is the global provision of an individual per-stream QoS and the decision which concrete cell to send in a slot-by-slot timescale. In traditional output-queued (OQ) switches all cells for a specific egress link are handled by one cell scheduler in each output port and there is almost no difference between the multiplexer being fed by M input links with a rate of λ/M each or one input with a rate λ. For $M \to \infty$ an FCFS[3] scheduler with Poisson traffic is appropiately modelled by an $M/D/1$ queueing system. A lot of analytical results and approximations are available for simple $G/D/1$ systems [8, 21], e.g. under bursty workload ($MMPP/D/1$). In many cases the performance can be numerically calculated as the cell delay distribution $PDF(d)$ in stationarity.

[1] in this paper we use the term delay as a synonym for waiting time; in the graphs all constant offsets are omitted

[2] for delay quantile d_ε holds: $Pr\{delay > d_\varepsilon\} = \varepsilon$

[3] FCFS=first come first served [8], WFQ=weighted fair queueing, (W)RR=(weighted) round robin, EDF=earliest deadline first

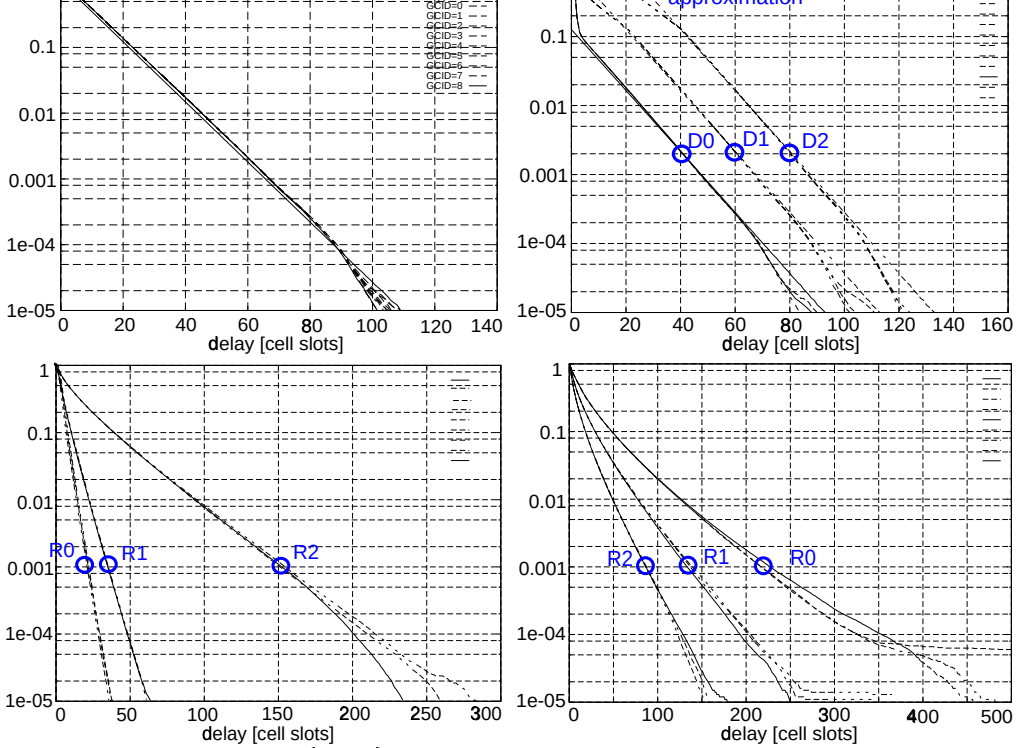

Figure 7: OQ switch: $Pr\{d > t\}$ for FCFS, EDF, RR and WRR (in Z-order) at $\rho = 0.95$

For realtime applications the QoS demand per switch can be expressed as $Pr\{d > d_{max}\} < \varepsilon$ where ε and d_{max} are derived from ATM traffic and QoS parameters [2].

More complex schedulers have evolved for the need to treat distinct traffic streams differently. The static priority (SP) system [21] separates traffic by priority classes such that any lower-priority traffic has no influence on any higher-priority stream. This is useful for separating realtime non-bursty, bursty and best effort traffic. It is not suited for supporting individual $d_{max}[VC]$ per connection (VC).

Two families of QoS-capable schedulers evolved in the past, each of them having a parameter per connection (VC), with which the performance can be controlled. The family of WFQ^3 schedulers uses a rate-proportional parameter $w[VC]$, which guarantees a certain share of the bandwidth during congestion and offers an average service rate proportional to $w[VC]$. The lower-complexity approximations of WFQ, WRR^3 [5] and RR^3 (unweighted) are assumed here.

The other family offers a parameter $D[VC]$ which has the meaning of deadline. The algorithm EDF^3 [3, 9] is known to be optimum in the sense that it minimizes the total probability of exceeding the deadlines for all streams. With a properly chosen parameter $D[VC] = d_{max}[VC] + c$ this very successfully guarantees individual delay bounds[4] statistically. The complexity of EDF due to its timestamp-sorter can be reduced by its approximation RPQ [11, 10] with any precision.

These schedulers have been evaluated and simulated in a traditional OQ environment for later comparison with their VOQ counterpart. For the graphs shown in this paper a 16*x*16 switch has been symmetrically loaded with Poisson traffic[5] at $\rho = 0.95$ using 2160 streams[6] with nine different characteristics in rate and deadline, as shown in fig. 8.

The complementary distribution functions (CDF) of the cell delay are shown in figure 7. Analytically this approximation for the waiting time distribution in $G/D/1$ systems [23] has been used:

$$Pr\{w \le t\} = 1 - ae^{-bt} \tag{2}$$

$$a = \frac{2E[w]^2}{E[w^2]} \qquad \text{and} \qquad b = \frac{2E[w]}{E[w^2]} \tag{3}$$

where the first and second moment are obtained for the $G/D/1/\infty/FCFS$ system. For EDF we observe the desired separation for streams with different delay requirements. A good approximation (seen as asymptotes in the figure) for the CDF is given by

$$Pr\{w \le t\} = 1 - ae^{-b(t+\bar{D}_0 - D_{0i})} \tag{4}$$

[4] these are much tighter than worst-case bounds and allow for the highest utilization [3]

[5] Bursty traffic has also been used but doesn't contribute much and is omitted due to space limitations

[6] 16 ports · 15 destinations each · 9 classes

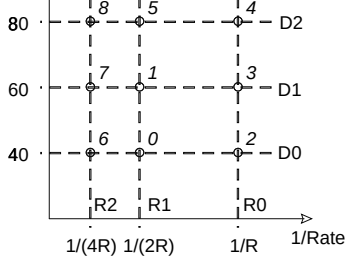

Figure 8: $rate[VC]$ and $d_{max}[VC]$

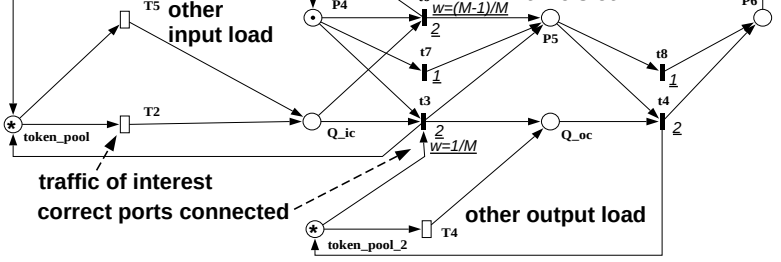

Figure 9: more abstract PN system for VOQ arbiter

using the mean deadline value [23]

$$\bar{D}_0 = \frac{\sum_i \lambda_i D_{0i}}{\sum_i \lambda_i} \tag{5}$$

Near the graph origin the asymptotes for EDF are given by formulas [21] for a static-priority (SP) system, because for $\Delta D_{0i} \to \infty$ EDF converges to SP. The deadlines can be chosen differently from the demanded $d_{max}[VC]$ by adjusting an offset $c[VC]$ in the case that different cell loss target values $\varepsilon[VC]$ are requested [7]. The simulation results in fig. 7 show that in this example the delay requirements are fulfilled with an ε of $3 \cdot 10^{-3}$.

4 VOQ Scheduling

A priority scheme must be established globally by the arbiter [19] (fig. 1). In a VOQ configuration there are additional M virtual schedulers for each priority in each input port, one for each output. Any of these schedulers is only activated to serve a cell if induced by the arbiter. Thus the arbiter appoints the service interval, opposed to an OQ switch, where a cell is served in each time slot.

The VOQ results in fig. 10 show that in principle the connection separation works the same way as in the OQ architecture (fig. 7). For all loads ρ these results can be obtained using the delay quantiles in fig. 4. These delay quantiles are obtained by simulation and analytical modelling. With a stochastic PN system (Petri Net [12]) the arbitration can be modelled most accurately. Fig. 11 shows the net for a $2x2$ switch, where cyclic polling of ports ($RR0 - RR1$) and weight-dependent firing in the transitions $S00 - S11$ have been explicitely included. The stationary solution of this system is computationally very intensive as the number of Markov states grow large.

Alternatively a more abstract modelling of the reasons for a higher delay, the input and output conflicts, provides quite good results. For the performance from port i to o this is modelled in fig. 9 by having the input and output port loaded by ρ_i^{in} and ρ_o^{out} (eq. 1) respectively. With an independent probability $1/M$ a token in Qic is transferred to Qoc. The performance is similar to two virtual $M/D/1$-type queues in series. Thus the calculated approximations in fig. 4 can

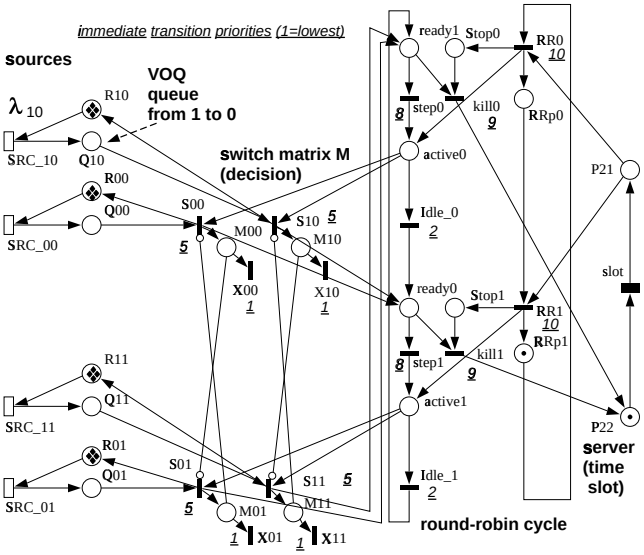

Figure 11: PN system for SIMP arbiter

be obtained based on eq. 2. The general delay quantiles are given in eq. 6 and especially for Poisson traffic in eq. 7.

$$d_\varepsilon = \frac{E[w^2]}{E[w]} ln(2\frac{E[w]^2}{\varepsilon E[w^2]}) \tag{6}$$

$$d_{\varepsilon,Poisson} = \frac{T_{slot}(2+\rho)}{3(1-\rho)} ln(\frac{3\rho}{\varepsilon(2+\rho)}) \tag{7}$$

Using the calculated moments the results in fig. 4 show an acceptable accuracy. This has been used to approximate the FCFS and EDF VOQ-scheduling performance in fig. 10.

5 Conclusion

It is shown that in a switch using virtual-output-queueing (VOQ) with weighted arbitration cell scheduling algorithms are feasible and perform the same manner as in an output-queued (OQ) configuration, where they have traditionally been used. For any scheduler the delay distributions are only a small factor higher than in their OQ variant. Some typical schedulers are compared by simulation and analytical methods are given to approximately calculate the delay performance for FCFS and EDF. This enables QoS-supporting schedulers to be used in this high-speed switch architecture.

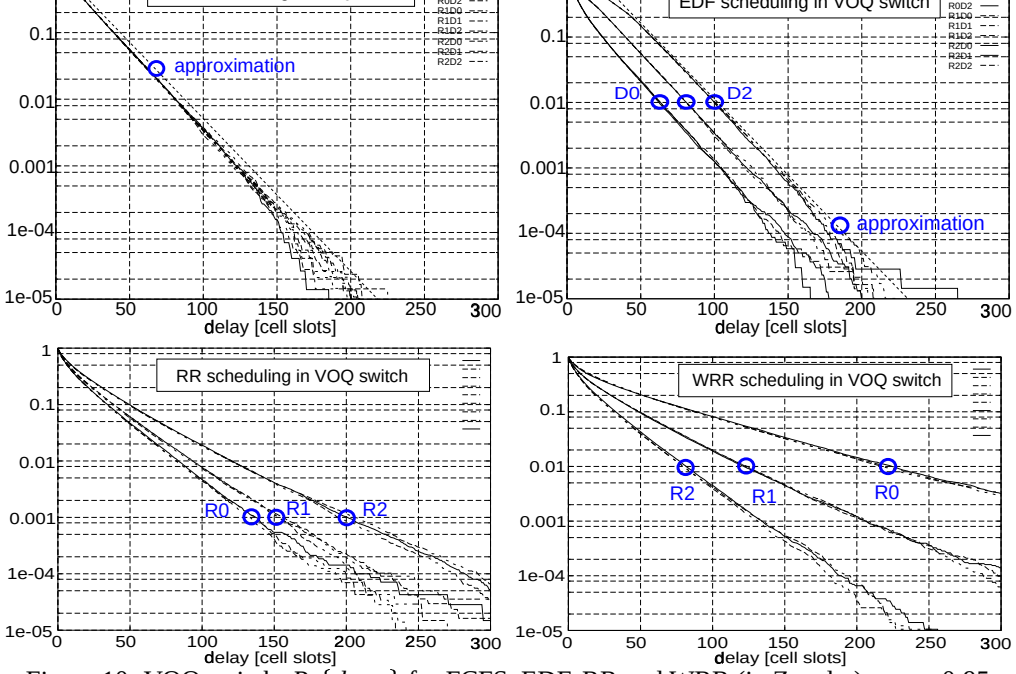

Figure 10: VOQ switch: $Pr\{d > t\}$ for FCFS, EDF, RR and WRR (in Z-order) at $\rho = 0.95$

References

[1] T. Anderson, S. Owicki, J. Saxe, and C. Thacker. High Speed Switch Scheduling for Local Area Networks. *ACM Transactions on Computer Systems*, ?(11), Nov 1993.

[2] The ATM Forum, Prentice Hall, Englewood Cliffs, N.J. *ATM user-network interface (UNI) specification version 3.1*, 1994.

[3] F. Chiussi and V. Sivaraman. Achieving High Utilization in Guaranteed Services Networks using Earliest-Deadline-First Scheduling. In *Proceedings of the International Workshop on QoS*, 1998.

[4] M. J. Karol, M. G. Hluchyj, and S. P. Morgan. Input versus output queueing on a space-division packet switch. *IEEE Transactions on Communications*, (12):1347–1356, Dec. 1987.

[5] M. G. H. Katevenis, S. Sidiropoulos, and C. Courcoubetis. Weighted round-robin cell multiplexing in a general-purpose ATM switch chip. *IEEE Journal on Selected Areas in Communications*, 9(8), Oct. 1991.

[6] H. Kim, C. Oh, and K. Kim. A High-Speed ATM Switch Architecture Using Random Access Input Buffers and Multi-Cell-Time Arbitration. In *Proceedings of the IEEE GLOBECOM*, 1997.

[7] H. Kist and D. Petras. Service Strategy for VBR Services at an ATM Air Interface. In *2nd. European Personal Mobile Communications Conference - EPMCC*, 1997.

[8] L. Kleinrock. *Queueing Systems, Vol. I: Theory*. John Wiley & Sons, New York, 1975.

[9] S. W. Lee, D. H. Cho, and Y. K. Park. Improved dynamic weighted cell scheduling algorithm based on Earliest Deadline First scheme for various traffics of ATM switch. In *Proc. IEEE Globecom '96*, pages 1959–1963, London, 1996.

[10] J. Liebeherr and D. Wrege. Priority Queue Schedulers with Approximate Sorting in Output-Buffered Switches. *IEEE Journal on Selected Areas in Communications*, 17(6):1127, June 1999.

[11] J. Liebeherr, D. Wrege, and D. Ferrari. Exact Admission Control for Networks with a Bounded Delay Service. *IEEE/ACM Transactions on Networking*, 4(5):885, Dec 1996.

[12] M. Marsan. *Modelling with Generalized Stochastic Petri Nets*. Wiley, 1996. ISBN 0-471-93059-8.

[13] N. McKeown. *Scheduling Algorithms for Input-Queued Cell Switches*. PhD thesis, UC Berkeley, 1995.

[14] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand. Achieving 100% Throughput in an Input-Queued Switch. *IEEE Transactions on Communications*, 47(8), Aug 1999.

[15] A. Mekkittikul and N. McKeown. A Starvation-free Algorithm For Achieving 100% Throughput in an Input-Queued Switch. In *Proc. of the IEEE International Conference on Communication Networks*, 1996.

[16] A. Mekkittikul and N. McKeown. A Practical Scheduling Algorithm to Achieve 100% Throughput in Input-Queued Switches. *Proceedings of the IEEE INFOCOM*, 1998.

[17] G. Nong, K. Muppala, and M. Hamdi. Analysis of Non-blocking ATM Switches with Multiple Input Queues. In *Proceedings of the IEEE GLOBECOM*, 1997.

[18] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization*. Prentice-Hall, Inc., 1982.

[19] R. Schoenen, G. Post, and G. Sander. Prioritized Arbitration for Input-Queued Switches with 100% Throughput. In *Proc. of ATM Workshop'99*, 1999.

[20] R. Schoenen, G. Post, and G. Sander. Weighted Arbitration Algorithms with Priorities for Input-Queued Switches with 100% Throughput. In *Proceedings of the IEEE Broadband Switching Systems*, 1999.

[21] H. Takagi. *Queueing Analysis - Discrete-Time Systems*, volume 3. North-Holland, ISBN 0-444-81611-9, 1991.

[22] Y. Tamir and H.-C. Chi. Symmetric Cross Bar Arbiters for VLSI Communication Switches. *IEEE Transactions on Parallel and Distributed Systems*, 4(1):13–27, 1993.

[23] B. Walke and W. Rosenbohm. Waiting-Time Distributions for deadline-oriented Serving. *Performance of Computer Systems*, page 241, 1979. North-Holland Publishing Company.