PAPER *IEICE/IEEE Joint Special Issue on Recent Progress in ATM Technologies*

An Architecture Supporting Quality-of-Service in Virtual-Output-Queued Switches

Rainer SCHOENEN†, *Member*

SUMMARY Input buffered switches most efficiently use memory and switch bandwidth. With Virtual Output Queueing (VOQ), head-of-line blocking can be avoided, thus breaking the throughput barrier of 58.6%. In this paper a switch architecture based on VOQ is proposed, which offers deterministic and stochastic delay bounds for prioritized traffic. This is achieved by a hybrid static and dynamic arbitration scheme, which matches ports both by a precalculated schedule and realtime calculations. By using weighted dynamic arbitration algorithms 100% throughput with lowest delays under all admissible traffic can be achieved. An integrated global priority scheme allows the multiplexing of realtime and data traffic. Following the arbitration decision, a cell scheduler decides locally in the input ports upon the next connection from which a cell is forwarded. Cell scheduling based on earliest-deadline-first (EDF) is shown to perform similar to its behaviour in an output-queued switch.
key words: ATM, VOQ, arbitration, allocation, scheduling

1. Introduction

ATM networks will have to consist of switches supporting the individual per-connection Quality-of-Service (QoS) ATM promised to deliver. Even IPv6 switches offering Diffserv/Intserv need architectures with built-in QoS features. Among the switching architectures, 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. With Virtual Output Queueing (VOQ) [1], where each input manages a separate queue for each output, it has been shown that a throughput of 100% can be achieved [2]–[4]. *Arbitration algorithms* resolve the contention for output ports in each time slot. The achievable throughput and delay performance heavily depends the used algorithm.

Arbitration can principially be performed by two ways: Statically by assigning time slots to specific connections or connection groups in advance or dynamically by resolving the contention for the same output port in each time slot anew.

The *static arbitration* (also called *allocation*) reserves time slots for specific connections in advance. This fixed schedule offers bandwidth guarantees, deterministic delay bounds for worst-case traffic and actively shapes (smoothes) the traffic. As shown in this paper, static arbitration can guarantee delay bounds for traffic accepted by the connection admission control (CAC) of ATM by assigning a constant service bandwidth to each connection. This immediately supports CBR and VBR services, whose traffic parameters are known at CAC time. Similarily, IPv6 guaranteed service [5] with RSVP [6] is supported. Due to the boundedness of the departure process, bounds can as well be given end-to-end. Analysis and results are in this paper.

On the other hand, *dynamic arbitration* distributes the left-over bandwidth to contending ports. Weighted algorithms [3] offer the best delay performance compared to unweighted algorithms and they operate much better in more difficult than symmetric load configurations [7]. Additionally, a static priority distinction between service classes must be integrated in a global arbitration method. A local priority decision within each input port alone is not sufficient. In this paper a method for prioritized arbitration and its performance are treated.

After the the centralized arbitration decision, the input ports are notified, which output port they have to send a cell to. A local *cell scheduler* within each input port must select the first queued cell of a suitable connection 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. Thus problems of individual QoS, flow separation and fairness can be addressed. The difficulty is the complexity of the system involving the hierarchical decision of both arbiter and scheduler. In this paper EDF scheduling is shown to perform similar in the VOQ configuration as it does in output-queued switches. Together with an approximative analytic model of the arbitration, the delay performance for FCFS and EDF can be obtained in a closed form.

The paper is organized as follows. Section 2 discusses VOQ and arbitration algorithms and related work. The static arbitration is explained and analysed in Sect. 3. Dynamic methods are treated in Sect. 4. Section 4.3 contains performance results for the dynamic arbitration method. Based on an approximative analytic model of the dynamic arbitration in Sect. 5, cell scheduling is treated in Sect. 6.

Manuscript received May 28, 1999
Manuscript revised August 26, 1999
†The author is with the Institute of Integrated Signal Processing Systems (ISS) at Aachen University of Technology (RWTH) (www.ert.rwth-aachen.de).

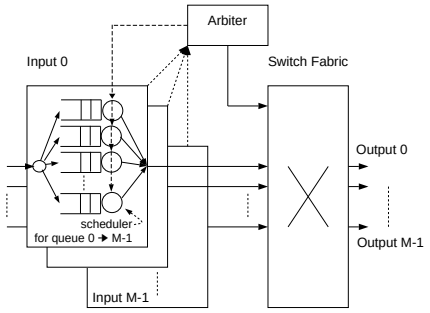

Fig. 1 VOQ architecture.

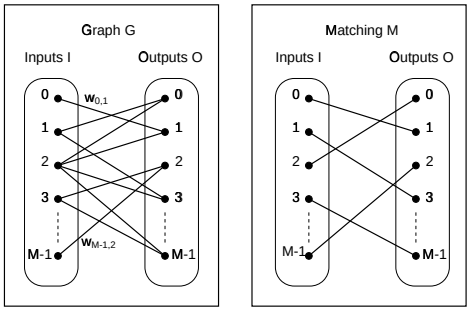

Fig. 2 Bipartite matching.

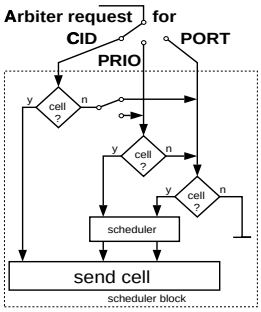

Fig. 3 Input scheduling.

2. VOQ System Description

The VOQ configuration [1] shown in Fig. 1 consists of a nonblocking switch fabric, an arbitration unit and M input and output ports. Each port has a link rate of r_{link}. Arriving cells on input port i are placed into the corresponding queue for their destination port o. This queue does not necessarily maintain cells in FCFS order, as often assumed, it can as well serve cells in weighted round-robin or earliest-deadline-first (EDF) order [8], as shown in Sect. 6. Let its current queue size be $q_{i,o}$. The process of arrivals to this queue is characterized by a mean rate $\lambda_{i,o}$. Let the input and output loads be

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

This is admissible [1] 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)) either by a lookup in the allocation table (if there is an entry) or by realtime calculations based on weights $w_{i,o}$ sent to it from the input ports. This task is equivalent to a bipartite graph matching problem [1] as shown in Fig. 2.

Static arbitration (allocation) has been used in traditional circuit-switched TDM systems, where the arrival and departure instances of frames are known in advance. TDM switches only have to precompute a periodic schedule to control the crosspoints in a switching network. For packet-switched networks (ATM) there is no frame reference time. Due to the asynchronous nature of packet traffic a periodic structure cannot immediately be exploited. Ideas to integrate precompute schedules for packet switches appeared in [9]–[11]. As shown in Sect. 3, the constant bandwidth which is guaranteed per connection combined with the bounded traffic model in Sect. 3.1 offers delay bounds per switch and end-to-end. Figure 3 shows that when the connection indentifier (CID) is allocated and a cell of that connection is available, no more decisions have to be taken. However, if there is no cell at all, the input-output port match cannot be changed and this slot is unused. Since the distance between service slots is deterministic and usually† larger than with dynamic arbitration (see Fig. 16), the average cell delay can be much lower with dynamic arbitration. This is why in a second stage a dynamic arbitration is performed on all unmatched ports.

A number of algorithms exist to solve the dynamic matching optimally, either as a maximum size matching (MSM) or a maximum weight matching (MWM) [12]. MSM finds the maximum number of input-output connections. MWM maximizes the weight sum of the match. It has been shown that 100% throughput can be achieved for all admissible i.i.d. arrivals [3] with MWM and $w_{i,o} = q_{i,o}$ (LQF).

Algorithms for solving the MWM problem are computationally complex with $O(M^3 \log M)$. A number of alternatives have been proposed, such as iMCFF [4], iLQF [2] and SIMP [7]. Even the unweighted MSM problem is rated $O(M^{2.5})$. MSM approximations exist with PIM [10], iSLIP [2], WFA [13] or FARR [14].

Without weight information instability and unfairness are likely, because queue backlogs can accumulate and cell delays become very large for bursty or asymmetric load [7]. Explicit treatment of priorities [15] is important, because it is required to separate traffic with extremely different QoS requirements (realtime and data traffic). For ABR an additional credit-based flow control mechanism [16] is required to guarantee zero cell loss by avoiding buffer overflows.

Cell scheduling within the input ports to offer differentiated QoS is a quite new topic. As shown in Fig. 3 the cell scheduler decides on which exact cell of which connection to forward, given the destination port o. The priority is given implicitely, because the arbiter has already chosen globally that this port has the highest priority for output o. With a static priority scheme the requested priority is served. But within the priority level a cell scheduler can decide using any known scheduling algorithm, not only FCFS. An approach for WFQ can be found in [17]. In Sect. 6 we show that given the performance of FCFS which is derived from

†In an underloaded switch.

an arbiter model, the performance of EDF can be calculated.

3. Static Bandwidth Allocation

The delay guarantee concept of allocation is based on the individual (per-VC) guarantee of a service rate. This guarantee is firm, i.e. the probability $Pr\{d > d_{max}\}$[†] to exceed a delay d_{max} is zero. This is in contrast to statistical delay bounds, where $Pr\{d > d_{max}\}$ is never zero but only tends to zero for higher d_{max} (typically with an exponential decay [18]) and the performance heavily depends on the total load of all other connections and their statistical properties (an undesired property). The requirement for firm delay bounds with allocation is that the traffic of the connection-of-interest is bounded, as shown now.

3.1 Bounded Traffic

Traffic bounds are common in integrated services networks. For ATM, the known traffic descriptor (PCR, $CDVT$) bounds the peak cell rate over an interval, (SCR, MBS) bounds the average rate over a longer interval. The contract conformance can easily be policed by the GCRA [19] algorithm. IPv6 offers a TSpec in its RSVP message [20] during connection admission, which is very similar.

The parameters are fitted into the traffic bound model of Cruz [21], [22] which bounds the amount of traffic A (given by its rate function R) in an interval (t_1, t_2) by

$$\int_{t_1}^{t_2} R(t)dt = A(t_1,t_2) \leqq \min_{1 \leqq i \leqq n} \{\sigma_i + \rho_i(t_2 - t_1)\} \quad (2)$$

This can be written $R \sim (\vec{\sigma}, \vec{\rho})$ for n pairs (σ_i, ρ_i). Each pair bounds the traffic rate to ρ_i with a burst tolerance σ_i. ATM offers $n = 1$ for CBR and $n = 2$ for VBR in the traffic contract at the UNI [19]. Using $T_{PCR} = 1/PCR$ and $T_{link} = 1/r_{link}$ we give the following mapping of parameters[††]

$$\begin{aligned} \sigma_1 = BS &= 1 + \lfloor CDVT/(T_{PCR} - T_{link}) \rfloor, \\ \rho_1 = PCR, \quad \sigma_2 &= MBS, \quad \rho_2 = SCR \end{aligned} \quad (3)$$

To calculate firm delay bounds, a worst-case traffic model [23] using these parameters must be applied. It consists of periodically repeated bursts of length BS (MBS f. VBR) cells followed by silence such that the average rate during the period length is PCR (SCR). The benefit of this model is that delay bounds derived for the worst-case cannot be exceeded by any other traffic which also obeyes the traffic contract.

[†] CDF, a complementary distribution function.

[††] In units of *cells* and *cells/second*.

3.2 Construction of the Allocation Table

A switch that supports static arbitration maintains an allocation table $T = [\vec{t_k}]$ $(0 \leqq k < S)$ that contains the precomputed and periodically repeated schedule for S time slots in a frame [11] ($T_{period} = S \cdot T_{link}$), see Fig. 4. In a slot k, $\vec{t_k}$ contains the input port numbers i_o for each precomputed match $(i_o \to o)$:

$$(i_o \to o) \quad \Leftrightarrow \quad t_{k,o} = i_o \quad (4)$$

For each entry a connection identifier ($CID = f(VPI, VCI)$) is specified. Empty entries are readily available for dynamic arbitration. For each CID a number of slots $n(CID)$ are reserved (for CBR: $r = PCR$, VBR: $r = SCR$):

$$n(CID) = \left\lceil \frac{\omega \cdot r}{r_{slot}} \right\rceil \quad (5)$$

Each allocated slot contributes to a bandwidth[†††] of

$$r_{slot} = r_{link}/S = T_{period}^{-1} \quad (6)$$

The overallocation factor ω ($\omega > 1$) is used to adjust the service rate by a desired amount. This connection is then served with an allocated bandwidth of

$$\mu_s(CID) = T_\mu^{-1} = n(CID) \cdot r_{slot} \quad (7)$$

which determines the individual (per-VC) load ρ_{CID}

$$\rho(CID) = \lambda_a/\mu_s(CID) = T_\mu/\bar{a} \quad (8)$$

where the arrival traffic rate $\lambda_a = \bar{a}^{-1}$ is the mean rate of the stationary traffic process. Due to the discrete nature of μ_s, the load can only be adjusted in discrete steps (Eq. 5). For that reason the effective ω changes slightly to

$$\omega(CID) = \mu_s(CID)/PCR(CID) = 1/\rho(CID). \quad (9)$$

3.3 Metrics for Allocation

Algorithms for the distribution of $n(CID)$ slots into the allocation table can be obtained from [9]–[11]. The connections are served in a per-VC manner with the service time given by the slot distances. A number of allocated slots for a connection implies an average slot distance of $T_\mu = T_{period}/n(CID)$. However, due to quantization and blocking effects this is not equidistant. The maximum tolerable slot offset is controlled with a parameter Ω_{max}, which bounds the $CDVT_{max}$ of the allocated slots of each connection. The deviation is quantified with the coefficient-of-variation COV_s of the interservice time $\tau_{s,i} = t_{i+1} - t_i$.

$$COV_s = \sqrt{Var(\tau_s)}/E(\tau_s) \quad (10)$$

[†††] Corresponds to the smallest supported rate, e.g. $64kb/s$.

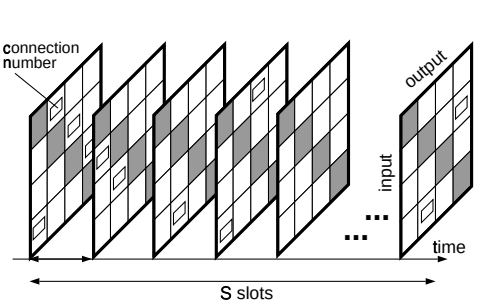

Fig. 4 Allocation table.

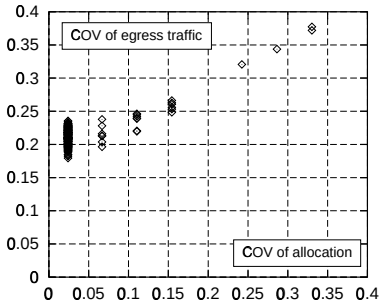

Fig. 5 COV_o vs. COV_s.

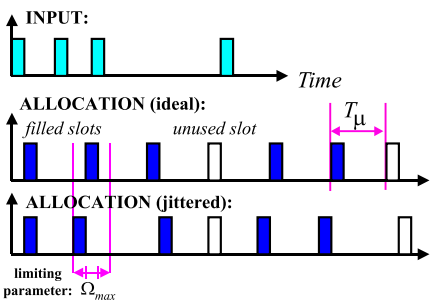

Fig. 6 Allocation jitter.

It is a measure of the quality of the allocation in the sense of a smooth service time distribution. It is desirable to have $COV_s = 0$ $(G/D/1)$. The actual COV_s depends on S, $n(CID)$, the total number of other allocated (blocked) slots, the allocation algorithm and Ω_{max} (see Fig. 6).

The output traffic stream is shaped such that it assimilates the service process. Thus COV_s and the output traffic variation COV_o are strongly correlated. COV_o also depends on $\omega(CID)$: Even with ideal allocation ($COV_s = 0$), unused slots lead to gaps in the output traffic stream (Fig. 6) which adds one component to the COV_o: Let for simplicity the random process *slot used* be i.i.d. and the probability of an unused slot be $p = 1 - \rho(CID)$. Then the assumed geometrically distributed interdeparture time has

$$COV_o = \sqrt{p} = \sqrt{1 - \rho(CID)} \tag{11}$$

For example, with $\omega(CID) = 1.05$, $\rho(CID) \approx 0.95$ the contribution is $COV_s = 0.218$, which can be seen in Fig. 5. Here the traffic arrived with $COV_a = 0.4$. The graph has been obtained by allocation of 240 connections. Each dot represents the values for one connection. The COV_s due to quantization is

$$COV_{s,quant} = \sqrt{T_{link}^2/12T_\mu^2} \tag{12}$$

which is 0.015 here. Allocation jitter due to Ω_{max} (Fig. 6) is at most $COV_{s,\Omega} = 0.33$ here, which can be seen in Fig. 5. It is bounded by

$$COV_{s,\Omega} \leqq \Omega_{max} \cdot T_{link}/(\sqrt{3} \cdot T_\mu) \tag{13}$$

3.4 Allocation Performance

Allocation has a number of positive properties (Table 1) which can be quantified for given traffic bounds, i.e. during CAC the model parameters are known for connections demanding real-time QoS.

When bounded (deterministic) traffic (section 3.1) is applied, an explicit delay bound is guaranteed (Fig. 8). Assuming a fully equidistant service ($COV_s = 0$) with a service rate $1/T_\mu$ higher than PCR for CBR or SCR for VBR, the time between instances where the queue is empty (busy time) is less than the period of the periodic worst case traffic bounded by the given traffic parameters. Thus delay bounds can be given.

Table 1 Properties of Allocation.

isolated, per-VC treatment (no influence of other traffic)
pre-determined constant delay bound → QoS
output traffic is shaped and bounded
self-policing (output conforms to GCRA)
QoS is adjustable with a single parameter ω
simple model for stochastic traffic: $G/E_k/1$
can be used together with dynamic arbitration → 100% utilization
non work-conserving globally → mean delay is higher
hybrid arbitration offers lower mean delay but higher miss ratio

A bound for CBR is derived here in continuous time. The delay a cell of bounded stream experiences is composed of the sum of two statistically independent terms d_e =*delay until service if queue is empty* and d_b =*delay due to cell's position in burst*. With a random phase, d_e is equally distributed (rect-shaped) over the interval $[0; T_\mu]$: $PDF(d_e) = p_e(d_e) = \prod(d_e/T_\mu - \frac{1}{2})$. The distribution $PDF(d_b)$ is a function shaped like a sequence of dirac (δ) impulses ($|||$):

$$p_b(d_b) = \frac{1}{BS} \sum_{i=0}^{BS-1} \delta(d_b - i \cdot (T_\mu - T_{link})) \tag{14}$$

The resulting delay distribution is the convolution ($\otimes$) of these components: $p(d) = p_e(d) \otimes p_b(d) = \prod \otimes |||$. For the example, with $BS = 4$ a sequence of four *diracs* convoluted with a *rect* should have the sandcastle shape, as the simulated result in Fig. 7 shows. The maximum value leads to this delay bound:

$$\max(d_{CBR}) \leqq (BS - 1) \cdot (T_\mu - T_{link}) + T_\mu \tag{15}$$

The situation for VBR is similar. One level of burstiness more makes it more complex, however. For a smooth worst case model [23] the same holds:

$$\max(d_{VBR}) \leqq (MBS - 1) \cdot (T_\mu - T_{PCR}) + T_\mu \tag{16}$$

For the heavy VBR model† Eq. (17) holds.

$$\begin{aligned} T_O &:= BS \cdot (T_\mu - T_{PCR}) \\ \max(d_{VBR}) &\leqq \lfloor (MBS - 1)/BS \rfloor \cdot T_O \\ &+ [(MBS - 1) \bmod BS] \cdot (T_\mu - T_{link}) + T_\mu \end{aligned} \tag{17}$$

†Maximal length bursts on link and PCR rate [23].

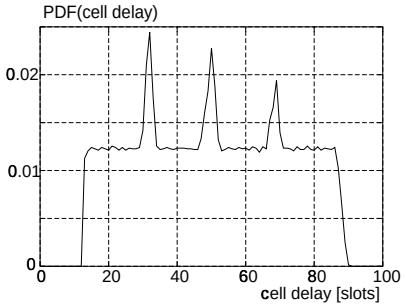

Fig. 7 Example $PDF(d)$.

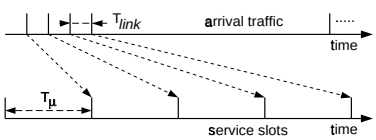

Fig. 8 Worst case traffic.

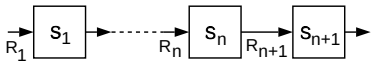

Fig. 9 Traffic bounds.

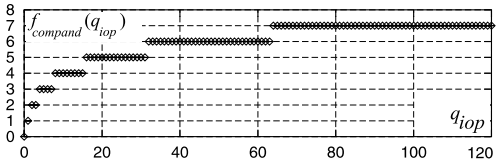

Fig. 10 $f_{compand}$.

When the assumption $COV_s = 0$ is relaxed, an additional delay of $CDVT_{max}$ must be added.

3.5 End-to-End Delay Bounds

With allocation an end-to-end delay guarantee d_N over N hops can be given (Fig. 9).

Lemma 1: For any switch n on the route and given CBR† input traffic parameters $R_n \sim (\sigma_n, \rho_n)$ the output is shaped such that its traffic is bounded by pairs $R_{n+1} \sim (\sigma_{n+1}, \rho_{n+1})$. Of the infinitely many possible pairs, two are of interest: Eq. (18) at the service time $\rho_n \cdot \omega_n$ and Eq. (19) at the declared ρ_n.

$$\begin{aligned} \rho'_{n+1} &= PCR_{n+1} = \rho_n \cdot \omega_n && (18)\\ \sigma'_{n+1} &= 1 + \lfloor 2T_{link}\Omega_{max}/(T_{PCR_{n+1}} - T_{link}) \rfloor \\ \rho''_{n+1} &= PCR_{n+1} = \rho_n && (19)\\ \sigma''_{n+1} &= 1 + \lfloor CDVT''_{max}/(T_{PCR_{n+1}} - T_{link}) \rfloor \\ CDVT''_{max} &= (\sigma_n - 1)(T_{PCR} - T_\mu) \\ &\quad + T_\mu + 2\Omega_{max}T_{link} \end{aligned}$$

In fact, after the first allocating switch the output traffic is much smoother due to active shaping. What results are lower delay bounds for any following switch. Overallocation ω is not necessary for switches $n > 1$, because overload is impossible by construction.

Theorem 1: For any limited number of hops n the total cell delay $d_{sum,n}$ and its output traffic R_{i+1} is bounded. Let R_i be the traffic bounds before switch i and $d_i(R_i)$ the resulting delay bound for it. Then holds

$$d_{sum,n} = \sum_{i=1}^{n} d_i(R_i) \qquad (20)$$

Proof (by induction): With given bounded traffic parameters R_1, the first switch at the network ingress has a bounded delay $d_{sum,1} = d_1$ given by Eqs. (15) and (17). This is obviously the delay for $N = 1$.
Let the theorem be true for switch n. Switch $n+1$ then has also a bounded delay output.
Proof: The worst-case delay of switch $n+1$, d_{n+1}, adds to $d_{sum,n}$, the maximum delay guaranteed so far. d_{n+1} is bounded because its input traffic R_i is bounded (Lemma 1) and allocation then guarantees eqs. 15 and 17.
Thus Theorem 1 holds for all n.
In Table 2 example values for typical voice/video traffic show the practical suitability of allocation. CAC rules can be obtained easily [11] by using the bounds above.

†For VBR substitute $SCR \rightarrow PCR$.

Traffic	Descriptors	$max(d_1)$	$max(d_n)$
CBR	$PCR = 64kb/s$, $CDVT = 5ms$	$5.5ms$	$5.5ms$
CBR	$PCR = 1Mb/s$, $CDVT = 1.4ms$	$1.76ms$	$352\mu s$
VBR	$PCR = 10Mb/s$, $SCR = 3.3Mb/s$ $MBS = 1000$ (MPEG-2)	$107ms$	$400\mu s$

Table 2 Delay bounds for example realtime applications (with $maxCTD = 200ms$, $\omega = 1$) for switch 1 and the following (n).

4. Dynamic Arbitration

Dynamic arbitration serves as a complementary mechanism to use unallocated slots for connections which do not require strict delay bounds or require a lower mean cell delay, which cannot be provided by static mechanisms.

4.1 Priority Support for Weighted Matching

Weighted matching algorithms can achieve full throughput under admissible load [3]. However, with different priority classes a stable operation cannot be guaranteed. In fact an admissible load of higher priority cell streams can nevertheless lead to unbounded delay if lower priority (best effort) connections experience congestion (high queue weight $w_{i,o}$). When a smaller weight on another port represents real-time cells only, they have to suffer. Only a globally prioritizing arbitration can achieve the correct separation.

The goal is to have no influence of lower priority queue weights over higher priority weights. A three-dimensional extension of the known weighted matching algorithms is discarded because of the complexity. We aim for a method operating in two dimensions only.

Consider a queue organization as shown to Fig. 1 and assume the queues are additionally separated per priority (with P priority levels, 1=highest). Let the

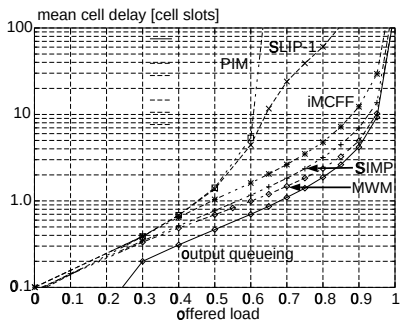

Fig. 11 Std. scenario.

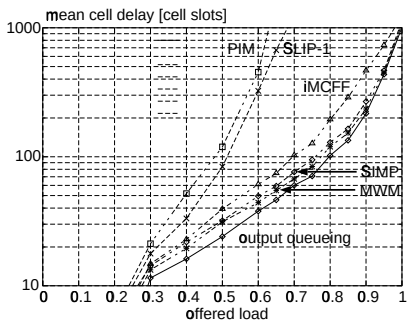

Fig. 12 Burst scenario.

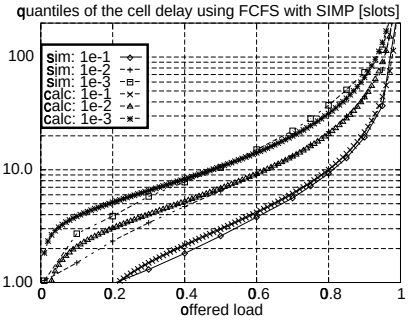

Fig. 13 $d_{quantile(x)}(\rho)$.

queue states on input i, output o and priority level p be $q_{i,o,p}$. The arrivals to these queues are characterized by a mean rate $\lambda_{i,o,p}$. All weighted matching algorithms operate on a two-dimensional weight $w_{i,o}$. Therefore we map $[q_{i,o,p}]$ to $[w_{i,o}]$ in a special way. The simplest case is $w_{i,o} = \sum_{p=1}^{P} q_{i,o,p}$ (no priority support).

The key idea of the simplification proposed in this paper is to use the following mapping function (modified weighted summation) to reduce q_{iop} to a two-dimensional format that can be handled by any weight dependent arbitration algorithm [15].

$$w_{io} = \sum_{p=1}^{P} c_p \cdot f(q_{iop}) \tag{21}$$

$$c_p = \prod_{q=p+1}^{P-1} k_q = \prod_{q=p+1}^{P-1} 2^{b_q} \tag{22}$$

The coefficients c_p are computed in Eq. (22) such that the weight of a queue with priority p is k_{p+1} times higher (b_{p+1} bits) than that of the next lower priority $p+1$. For the function $f(q_{iop})$ the alternatives are

$$f_{eq}(q_{iop}) = q_{iop} \tag{23}$$

$$f_{sat}(q_{iop}, k_p) = \{ \begin{array}{ll} q_{iop} & \text{if} \quad q_{iop} < k_p \\ k_p - 1 & \text{if} \quad q_{iop} \geq k_p \end{array} \} \tag{24}$$

$$f_{compand}(q_{iop}) = \left\{ \begin{array}{r} 0 \ : \ q_{iop} = 0 \\ \lfloor \log_2(q_{iop}) + 1 \rfloor \ : \ \text{else} \\ 2^{b_p} - 1 \ : \ q_{iop} \geq 2^{b_p} \end{array} \right. \tag{25}$$

f_{eq} is a simple addition, which still has a problem when due to large values an overflow into the bits reserved for higher priorities occurs. A sufficient priority separation is achieved if the overflow probability is very low. The saturation f_{sat} cuts off any overflow. This has the trade-off that the wordlength (the dynamic range) is small. An example for saturation is $P = 3; b_p = 3 \Rightarrow k_p = 8; q_{io1} = 2 = 010_b, q_{io2} = 5 = 101_b, q_{io3} = 15 \Rightarrow f_{sat}(q_{io3}) = 111_b \Rightarrow w_{io} = 010101111_b$. If a wider dynamic range is necessary (e.g. for bursty traffic) but bits for the wordlength must be saved, the companding characteristic $f_{compand}$[†] is recommended. It requires $2^{b_p} - 1 = \lfloor \log_2(2^{b_v} - 1) + 1 \rfloor$ or $2^{b_p} - 1 = b_v$. The most effective value is $b_p = 3$ bit offering a dynamic range of $b_v = 7$ bit (Fig. 10), which has shown to be sufficient for good results. The operation can very well be implemented in hardware, since only the position of the highest 1 bit must be determined for the $\log_2$ operation.

With this mechanism priorities are supported because queued cells of a higher priority are given a higher weight $w_{i,o}$. The number of bits per priority b_p representing the dynamic range within the class allows scaling of the arbitration performance from MSM ($b_p = 1$) to MWM ($b_p \to \infty$). This has consequences especially for asymmetric load [7].

[†]Similar to the μlaw used for speech coding.

Table 3 The SIMP Algorithm.

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

4.2 An Approximation for Weighted Arbitration

As one low complexity method to achieve weighted matching by approximating MWM, the SIMP [7] algorithm is outlined shortly in Table 3. SIMP approximates MWM by successively choosing the edge with the highest weight for each output node in order (step 4). This resolves the output contention problem for each port in a cyclic manner. It cannot generally find the global maximum MWM would find. Visually we can imagine that SIMP only considers one triangular half of the weight matrix $[w_{io}]$, instead of the whole as with MWM. Due to the cyclic shift of this viewport the number of weights considered for a specific output port changes cyclically from 1 to M. As we see in Sect. 4.3 the approximation yields reasonable results.

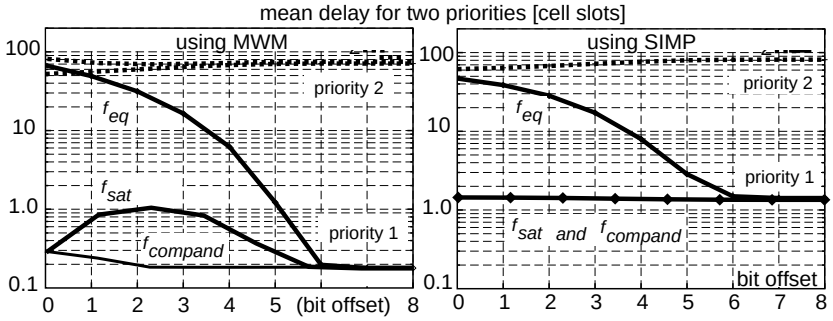

Fig. 14 Priority distinction.

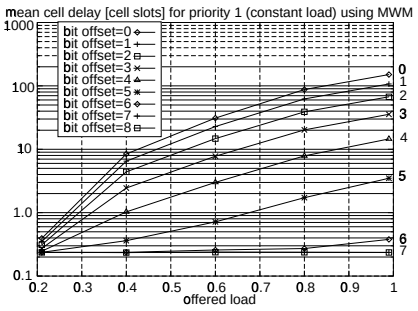

Fig. 15 Priority scenario.

4.3 Performance Results for Arbitration

In this section we compare the performance results for reference algorithms output queueing, MWM, iSLIP, PIM, iMCFF with SIMP using simulations for an 16×16 switch with OPNET [24].

With Bernoulli traffic and a symmetrically chosen destination (1 of $M-1$), each input port carries a total offered load of ρ, i.e. $\lambda_{i,o} = \rho/(M-1)$.

Bursty traffic is important for many reasons. According to [25] we use a packet train model producing a burst of $B(t)$ cells at full rate followed by empty slots such that the mean rate $\lambda_{i,o} = \rho/(M-1)$ is the desired fraction of the input rate. B is exponentially distributed here; the *mean burst length* $\bar{B}$ is 32. This corresponds to the mean length of an Ethernet PDU segmented into ATM cells.

In Fig. 11 the results for PIM and iSLIP are as in [25]. The best case is output queueing which we want to approximate with VOQ. The ideal arbitration for VOQ is MWM which most closely approximates output queueing. As we see, iMCFF performs worse than SIMP. The performance of SIMP is in between iMCFF and MWM.

In Fig. 12 we see the delay performance for the bursty scenario. The most important characteristics are: (i) the absolute delay is two decades higher than for the previous scenario due to the burst scale queueing effect and the short-term asymmetries. (ii) PIM and iSLIP perform similar with more than 300 cell transmission times above $\rho = 0.6$, i.e. noticably worse than the other algorithms. (iii) As before, iMCFF, SIMP, MWM and output queueing are quite close to each other with improved performance (less delay) in ascending order.

We observe that the algorithms that decide based on weights (iMCFF, SIMP, MWM) perform very well for typical traffic.

When using the first priority mechanism f_{eq} the main parameter for achieving priority separation is the priority weight factor $k_q = 2^{b_q}$ (b_q shift bits). The other two degrees of freedom in the choice of traffic parameters manifest in $\rho_{total} = \rho_1 + \rho_2$ and s_{p1} (share of first priority cells) with $\rho_1 = s_{p1} \cdot \rho_{total}$ and $\rho_2 = (1 - s_{p1}) \cdot \rho_{total}$. The behavior for two priority classes[†] can be seen in Fig. 15, where $\rho_1 = 0.2$ is held constant and ρ_2 is varied as bursty traffic. We see that there is almost no influence of the lower priority load on the higher priority performance when a wordlength of seven or more bits is used.

To analyse quantitatively how many bits b_p are needed we study the mean delay for both priority groups at a fixed load $\rho_{total} = 0.7$ and a fixed ratio $s_{p1} = 0.2$ as a function of different b_p. The fixed values used have revealed to be the most expressive.

For bursty traffic Fig. 14 shows that about $b_p = 6$ bits are needed to fully separate the priorities. We can verify that with enough bits $\bar{d}_{prio1} = 0.2$ slots at load $\rho_1 = 0.2$. This is exactly the delay for MWM with $\rho = 0.2$ in Fig. 11. Thus we have an exact priority separation. The higher priority traffic performance does only depend on its total load.

The necessary value of b_p depends on the traffic in priority p. This is not desirable, so we apply the deterministic separation by saturation (Eq. (24)), where no overlapping is possible, i.e. $w_{io,p+1}$ cannot become higher than 2^{b_p}. We lose some of the weight dynamics in this priority because the values possible are in the interval $[0, 2^{b_p} - 1]$.

With saturation we observe in Fig. 14 that now the separation is much improved for $1 \ldots 5$ bits. There is still some influence typical for MWM[††]. We can even choose one bit per priority and achieve a considerable separation. This, however, ignores all weights within a priority level and yields a bad performance under asymmetric workload as MSM does. With companding (Fig. 14) we finally observe the very best separation. Here three bits are sufficient for a complete separation, whereas the dynamics is as good as pure MWM.

[†] Higher priority traffic is Bernoulli, lower is bursty.

[††] The maximum matching is the one that maximizes the sum of the matched weights. This sum might be still higher for some lower priority matches (e.g. $7 + 7$) than for the alternative match (e.g. $8 + 0$).

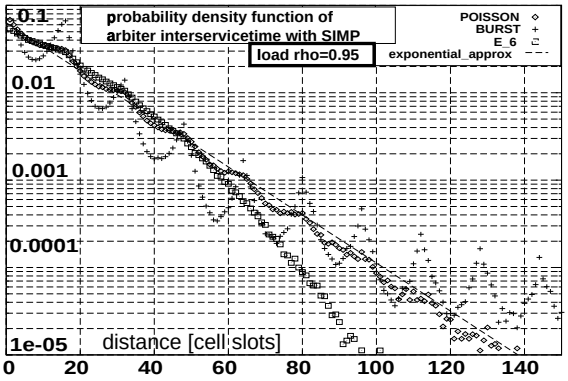

Fig. 16 VOQ switch: PDF for inter-service time of SIMP arbiter.

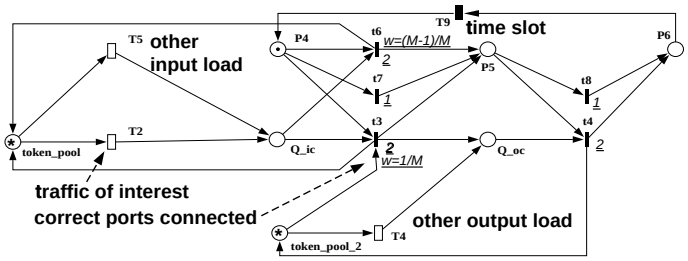

Fig. 17 PN system for VOQ.

5. Calculating VOQ Performance

In a VOQ configuration there are M virtual schedulers for each priority in each input port, one for each output. Each 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. In Fig. 16 the time between successive service events is shown for a number of configurations, e.g. different load and different traffic for SIMP. Observe the periodicity in multiples of $M \cdot T_{slot}$. For lower load or smooth traffic with SIMP an exponential distribution looks very similar, but assuming an $M/M/1$ queueing system is wrong because in the VOQ system there is no statistical independence between arrivals and service. The dependence is exactly what we want weighted matching algorithms to achieve.

For determining the delay performance of the scheduler a performance model for the arbiter is needed. The delay quantiles can be obtained by simulation and analytical modelling. With a detailled stochastic PN system (Petri Net [26]) the arbitration can be modelled most accurately [8], but the stationary solution 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. 17 by having the input and output port loaded by ρ_i^{in} and ρ_o^{out} 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 approximation can be obtained based on Eq. (26).

$$Pr\{w \leq t\} = 1 - ae^{-bt} \tag{26}$$

$$a = \frac{2E[w]^2}{E[w^2]} \quad \text{and} \quad b = \frac{2E[w]}{E[w^2]} \tag{27}$$

where the first and second moment are obtained for the $M/D/1/\infty/FCFS$ system [27]. For realtime applications the QoS demand per switch can be expressed as $Pr\{d > d_{max}\} < \epsilon$ where ϵ and d_{max} are derived from ATM traffic and QoS parameters [19]. The delay quantiles are given in Eq. (28), and especially for Poisson traffic in Eq. (29).

$$d_\epsilon = \frac{E[w^2]}{E[w]} \ln(2\frac{E[w]^2}{\epsilon E[w^2]}) \tag{28}$$

$$d_{\epsilon,Poisson} = \frac{T_{slot}(2+\rho)}{3(1-\rho)} \ln(\frac{3\rho}{\epsilon(2+\rho)}) \tag{29}$$

In Fig. 13 the simulated and calculated results for $\epsilon \in \{10^{-1}, 10^{-2}, 10^{-3}\}$ are compared and show an acceptable accuracy. This has been used to obtain approximations for the FCFS scheduling performance in Fig. 19 at $\rho = 0.95$.

6. Cell Scheduling in VOQ Switches

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 scheduler in the 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† scheduler with Poisson traffic is appropiately modelled by an $M/D/1$ queueing system. In many cases the performance can be numerically calculated as the cell delay distribution $PDF(d)$ in stationarity.

More complex schedulers have evolved for the need to treat distinct traffic streams differently. Here we concentrate only on the most promising algorithm $EDF^\dagger$ [29], [30], because the principal properties of VOQ scheduling are to be shown. EDF offers a parameter $D[VC]$ which has the meaning of deadline. With a

†FCFS=first come first served [28], EDF=earliest deadline first.

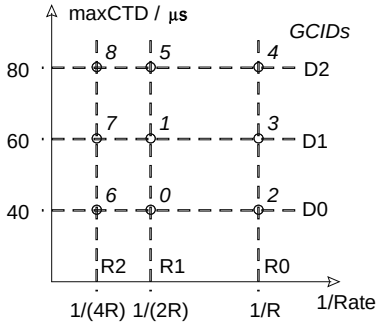

Fig. 18 $Rate[VC]$ and $d_{max}[VC]$.

properly chosen parameter $D[VC] = d_{max}[VC] + c$ this very successfully guarantees individual statistical delay bounds.

For the graphs shown in this paper the slot time is assumed to be $T = 1\mu$s in a 16×16 switch. The analyzed simulation scenario consists of 2160 streams[†] with nine different characteristics in rate and deadline, as shown in Fig. 18. All ports are homogeneously and symmetrically loaded with $\rho = 0.95$ (Poisson traffic).

The results in Fig. 19 and 20 show that in principle the connection separation works the same way as in an OQ architecture. For all loads ρ these results can be obtained using the delay quantiles in Fig. 13.

For EDF, a good approximation for the CDF is given by

$$Pr\{w \leq t\} = 1 - ae^{-b(t+\bar{D}_0-D_{0i})} \tag{30}$$

using the mean deadline value [27] and the shape parameters a, b obtained from the VOQ model in Eq. (27).

$$\bar{D}_0 = \frac{\sum_i \lambda_i D_{0i}}{\sum_i \lambda_i} \tag{31}$$

Using Eq. (30) and a, b from Eq. (27), the performance can be calculated. The simulated graphs in Fig. 20 show the typical behaviour: A separate performance for each delay class, the distance is exactly the difference of the deadlines. The slope of the graphs is exactly the same as for FCFS, which validates that our approximation is applicable. As it can be seen, EDF supports deadlines as it does in an output-queued switch.

7. Conclusion

It is shown that in a switch using virtual-output-queueing (VOQ) a number of arbitration algorithms can be applied. Static allocation offers deterministic, i.e. firm, delay bounds for bounded traffic per-VC. However, the mean delays can become quite high. Dynamic algorithms perform quite close to the ideal output queueing. With weighted algorithms priorities can be supported and statistical delay bounds can be given,

[†]16 ports · 15 destinations each · 9 classes.

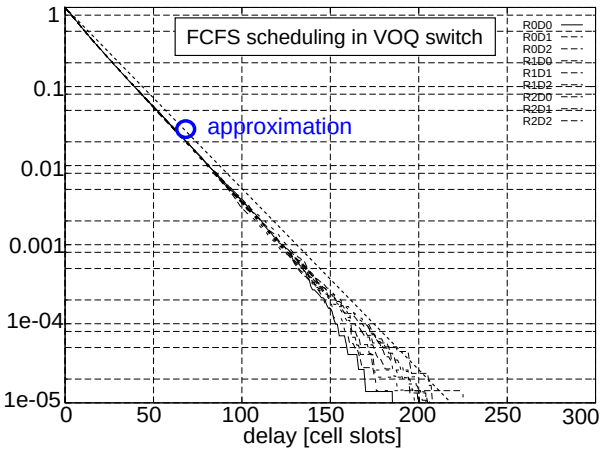

Fig. 19 VOQ switch: $Pr\{d > t\}$ for FCFS.

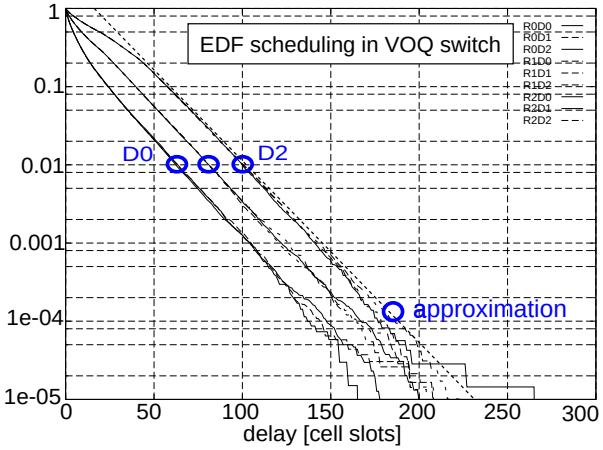

Fig. 20 VOQ switch: $Pr\{d > t\}$ for EDF.

the performance of which depends on other connections, however. Scheduling algorithms can further separate connections and operate similar to their output-queued variant. With the performance approximations for the SIMP algorithm given in this paper the performance of more complex schedulers can be derived, as shown for EDF here.

Acknowledgements

The autor would like to thank the anonymous reviewers for their valuable comments.

References

[1] 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.

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

[3] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% Throughput in an Input-Queued

Switch," *IEEE Transactions on Communications*, vol. 47, Aug 1999.

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

[5] S. Shenker, C. Partridge, and R. Guerin, "Specification of Guaranteed Quality of Service." IETF RFC 2212, Sep 1997.

[6] J. Wroclawski, "The Use of RSVP with IETF Integrated Services." IETF RFC 2210, Sep 1997.

[7] 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.

[8] R. Schoenen and R. Hying, "Distributed Cell Scheduling Algorithms for Virtual-Output-Queued Switches," in *Proceedings of the IEEE GLOBECOM*, 1999.

[9] J. Y. Hui, *Switching and Traffic Theory for Integrated Broadband Networks*. ISBN 0-7923-9061-X: Kluwer, 1990.

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

[11] R. Schoenen and G. Post, "Static Bandwidth Allocation for Input-Queued Switches with strict QoS bounds," in *Proceedings of the IEEE Broadband Switching Systems*, 1999.

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

[13] Y. Tamir and H.-C. Chi, "Symmetric Cross Bar Arbiters for VLSI Communication Switches," *IEEE Transactions on Parallel and Distributed Systems*, vol. 4, no. 1, pp. 13–27, 1993.

[14] C. Lund, S. Phillips, and N. Reingold, "Fair Prioritized Scheduling in an Input-Buffered Switch," in *Proceedings of the IEEE International Conference on Broadband Communications*, 1996.

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

[16] R. Schoenen, G. Post, and A. Müller, "Analysis and Dimensioning of Credit-Based Flow Control for the ABR Service in ATM Networks," in *Proceedings of the IEEE GLOBECOM*, 1998. Vol.4 p.2399.

[17] D. Stephens and H. Zhang, "Implementing Distributed Packet Fair Queueing in a Scalable Switch Architecture," in *Proceedings of the IEEE INFOCOM*, 1998.

[18] L. Kleinrock, *Queueing Systems, Vol. II: Applications*. New York: John Wiley & Sons, 1976.

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

[20] S. Shenker and J. Wroclawski, "General Characterization Parameters for Integrated Service Network Elements." IETF RFC 2215, Sep 1997.

[21] R. Cruz, "A Calculus for Network Delay, Part I: Network Elements in Isolation," *IEEE Transactions on Information Theory*, vol. 37, p. 114, Jan 1991.

[22] H. Michiel and K. Laevens, "Teletraffic Engineering in a Broad-Band Era," *Proceedings of the IEEE*, vol. 85, p. 2007, Dec 1997.

[23] J. Roberts, U. Mocci, and J. Virtamo, eds., *Broadband Network Teletraffic: Performance Evaluation and Design of Broadband Multiservice Networks; final report of action COST 242*. LNCS1155, Springer, 1996.

[24] MIL 3, Inc., 3400 International Drive, NW Washington, DC 20008, USA, *OPNET Release 6.0*, 1999. http://www.mil3.com.

[25] N. McKeown and T. Anderson, "A Quantitative Comparison of Scheduling Algorithms for Input-Queued Switches," *unpublished*, 1997. available at http://http.cs.berkeley.edu/%7Etea/atm.html.

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

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

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

[29] 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.

[30] 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*, (London), pp. 1959–1963, 1996.

Rainer Schoenen was born in Düsseldorf, Germany, on March 9, 1970. He received the Diplom-Ingenieur degree in electrical engineering from RWTH Aachen University of Technology, Germany, in 1995. Currently he is a Ph.D. candidate in the ISS department of RWTH Aachen. His research interests include traffic management for QoS supporting networks, switch architectures, credit-based flow control and stochastic Petri nets. He is a member of the IEEE.