Static Bandwidth Allocation for Input-Queued Switches with strict QoS Bounds

Rainer Schoenen and Guido Post
schoenen@ert.rwth-aachen.de
http://www.iss.rwth-aachen.de
Institute for Integrated Signal Processing Systems
Aachen University of Technology. Templergraben 55, 52056-Aachen, Germany

Abstract

Input buffered switches are known to suffer from head-of-line blocking that limits the throughput to 58.6%. It has been shown that with virtual output queueing (VOQ) 100% throughput can be achieved using arbitration algorithms. While dynamic algorithms decide based on some metrics in each time slot, a static arbitration called allocation reserves time slots for specific connections in advance. This fixed schedule offers bandwidth guarantees, strict delay bounds for worst-case traffic and active per-flow traffic shaping. In this paper we treat the performance, dimensioning and connection admission control of services that are treated with allocation. This is extended to provide bounds for the end-to-end delay. Several variants, the hybrid TDM/ATM, priority and connection allocation are discussed.

1 Introduction

Providing Quality-of-Service (QoS) per connection with broadband switching architectures is very important for ATM and IPv6. Input queued switches offer the most powerful architecture, 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% [1]. The Virtual Output Queueing architecture (VOQ) [2] can avoid this by providing a separate queue for each output. It has been shown that a throughput of 100% can then be achieved [3, 4].

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.

This can be performed in two different ways: Either 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 first way, discussed in this paper, offers firm QoS bounds while for the second the QoS problem is not solved for VOQ.

There are several ways for combining static and dynamic arbitration with a number of graduations that enables adjusting between performance and flexibility. In this paper we discuss the trade-offs between fully static (per-VC allocation) and fully dynamic arbitration. We show that static arbitration can guarantee delay bounds for traffic accepted by ATM-CAC or RSVP [5]. This supports CBR and VBR services as well as IPv6 guaranteed service [6]. Due to the boundedness of the departure process bounds can as well be established for end-to-end connections, as shown by analysis and simulation results.

The paper is organized as follows. Section 2 discusses related work. The static and dynamic arbitration is explained in section 3 and 4. Section 5 contains performance results for the static arbitration method and hybrid variants. CAC is treated in section 6.

2 Background and Related Work

We assume the VOQ configuration [2, 7], which consists of M ports for input and output, a nonblocking switch fabric and an arbitration unit. For this paper, arriving cells on input port i are placed into per-VC queues, groups of which are logically arranged for their destination port o. In each time slot the arbiter selects unique pairs of input and output ports (a "match" (i,o)). Dynamic arbitration algorithms decide based on information sent to it from the input ports. For this bipartite graph matching problem [2] a number of solution algorithms exist based on maximum size or weight matching (MWM, MSM) [8]. It has been shown that 100% throughput can be achieved for uniform and independent traffic [3] with MWM. A number of approximations for the computationally complex MSM and MWM have been proposed, e.g. MCFF, iMCFF [4] and SIMP [7] or the iterative algorithms PIM [9] and iSLIP [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 (e.g. time/space/time [10]). For packet-switched networks there is no frame reference time. Due to the asynchronous nature of packet traffic a periodic structure cannot immediately be exploited [10, p.46]. Ideas to integrate precompute schedules for packet switches first appeared in [10, p.161] and have been mentioned in [9].

For ATM (IPv6) we have to cope with different service classes, and their traffic descriptors (RSVP-TSpec) determine the bounds for the cell delay. In order to guarantee bounds on the delay a bounded traffic model is used. According to the notation of Cruz [11], the amount of traffic A (given by its rate function R) in an interval (t_1,t_2) is bounded by

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

This can be written $R \sim (\vec{\sigma},\vec{\rho})$ for n pairs (σ_i,ρ_i). For ATM[1] the applications specify one (for CBR) or two (for VBR) pairs at the UNI [12]. We define[2]

$$\begin{aligned} &\sigma_1 = BS = 1 + \lfloor CDVT/(T_{PCR} - T_{link}) \rfloor, \\ &\rho_1 = PCR, \quad \sigma_2 = MBS, \quad \rho_2 = SCR \end{aligned} \qquad (2)$$

with $T_{PCR} = 1/PCR$ and $T_{link} = 1/r_{link}$ and the other parameters defined at the UNI.

[1] the IPv6 RSVP Tspec bounds traffic in the same way
[2] in units of cells and cells/second

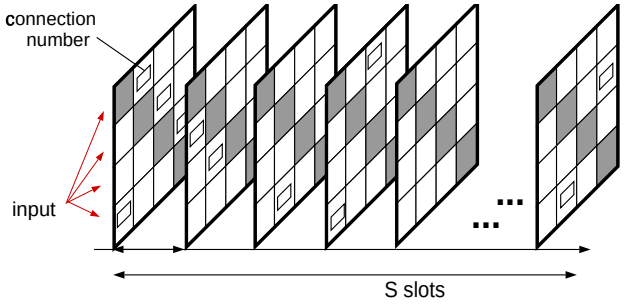

Figure 1: allocation table

For the calculation of delay bounds a worst-case traffic model [13] 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*)[3].

3 Static Arbitration

A switch that supports static arbitration maintains an allocation table $T = [\vec{t_k}]$ $(0 \le k < S)$ that contains the precomputed and periodically repeated schedule for S time slots ($T_{period} = S \cdot T_{link}$). This is visualized in fig. 1. For a slot k, $\vec{t_k}$ contains for all output ports o the number of the input port i_o to connect o, such that the precomputed matches $(i_o \to o)$ are

$$(i_o \to o) \Leftrightarrow t_{k,o} = i_o \tag{3}$$

By this compact storage only $S \times M$ numbers must be held in a memory. The indexing with o is advantageous for point-to-multipoint connections. Additionally for each port a connection identifier ($CID = f(VPI,VCI)$) and optionally a priority level *PRIO* are specified. Empty entries are available for dynamic arbitration that works only on the set of unmatched input and output ports.

For each *CID* a number of slots $n(CID)$ are reserved after positive connection admission control (CAC):

$$n_{CBR}(CID) = \lceil \frac{\omega \cdot PCR}{r_{slot}} \rceil, \quad n_{VBR}(CID) = \lceil \frac{\omega \cdot SCR}{r_{slot}} \rceil \tag{4}$$

Each allocated slot contributes to a bandwidth[4] of

$$r_{slot} = r_{link}/S = T_{period}^{-1} \tag{5}$$

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

$$\mu_s(CID) = T_\mu^{-1} = n(CID) \cdot r_{slot} \tag{6}$$

which can be used to express an individual (per-VC) load ρ_{CID}

$$\rho(CID) = \lambda_a/\mu_s = T_\mu/\bar{a} \tag{7}$$

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. 4). For that reason ω changes slightly to

$$\omega(CID) = \mu_s(CID)/PCR(CID) = 1/\rho(CID). \tag{8}$$

[3]Any cells violating the traffic contract policed by the GCRA [12] are considered lost and do not contribute to the delay performance

[4]corresponds to the smallest supported rate, e.g. 64kb/s

3.1 Allocation Procedure

An algorithm that distributes $n(CID)$ slots into the allocation table is outlined here: Calculate the required real number T_μ. Start with an arbitrary slot s_1 that is free for both input and output port at $T_1 = s_1 \cdot T_{link}$. Calculate the ideal next time positions $T_i = (T_1 + i \cdot T_\mu) \bmod T_{period}$. Occupy the nearest free slot around the ideal position, e.g. by trying the offsets $0, -1, +1, -2, +2, ..., \mp\Omega_{max}$ within a limit given by the tolerable COV_{out} or $CDVT_{out}$, the output traffic process of this connection should have.

$$\Omega_{max} = \lfloor CDVT_{out}/(2T_{link}) \rfloor \tag{9}$$

This method suffers from a certain "slot saturation" which indicates in some rejected connections above a total load of about 80..90%. Similar numbers are reported in [14] for TDM systems. However, it can guarantee a bounded jitter.

An alternative precomputation algorithm [10, 9] resolves slot conflicts by swapping the disturbing port pairings to other locations, where they also may experience conflicts. This is repeated iteratively until no more conflict remains. The whole table must be reestablished each time a new connection is admitted. This method can potentially allocate all slots as long as there is bandwidth available for the input and output link. But the swapping method disrupts all previously allocated connections, such that the resulting service slots may look like geometrically distributed with $COV_s \to 1$. Only for a singlerate TDM system [10] (one slot per connection) this is a viable way.

3.2 Quality of Allocation

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 degree of deviation can be quantified with the coefficient-of-variation (COV)[5] of the interservice time[6] t_s.

$$COV_s = \sqrt{Var(t_s)}/E(t_s) \tag{10}$$

It is a measure of the quality of the allocation in the sense of a smooth service time distribution. It is desirable to have $COV = 0$, i.e. a fully deterministic service ($G/D/1$). The actual COV_s depends on S, $n(CID)$, the number of other allocated (blocked) slots and the allocation algorithm. The slot quantization[7] contributes with $Var(t_s) = \sigma_s^2 = T_{link}^2/12$.

The output traffic stream is shaped such that it assimilates the service process. Thus we observe a strong correlation between COV_s and the output COV_o. In addition, unused slots lead to gaps in the output traffic stream which itself produce variations: Let for simplicity the random process *slot used* be i.i.d. and the probability of a used slot be $p = 1 - \rho(CID)$. Then the assumed GEO^1 interdeparture time has

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

which is e.g 0.218 for $\omega(CID) = 1.05$. In fig. 2 we see this value and the correlation for several connections arriving with $COV_a = 0.4$.

4 Dynamic Arbitration

In each time slot a dynamic arbitration matches those input/output pairs that are not blocked by the allocation (see section 2). The decision is based on the global state of the virtual output queues $w_{i,o}$ [3]. Each input port regularily sends

[5]typical: 0 for constant, 1 for geom. distributed distances

[6]distance between successive slots $t_{s,i} = t_{i+1} - t_i$

[7]only by rounding ideal time positions T_i to nearest slots

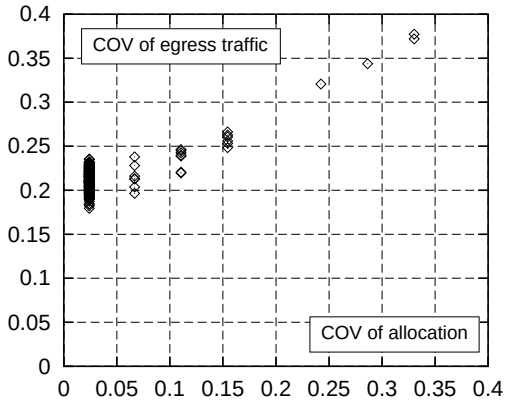

Figure 2: correlation: COV_o vs. COV_s

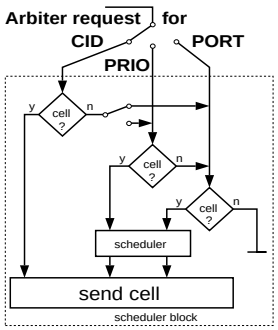

Figure 3: reaction to arbiter request

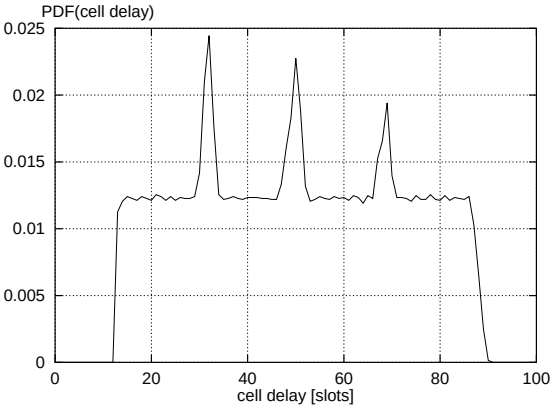

Figure 4: Bounded CBR traffic: PDF(d)

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% switch utilization
non work-conserving globally → mean delay is higher than with dynamic arb.
hybrid arbitration offers lower mean delay but higher miss ratio

Table 1: Properties of Allocation

state information to the arbiter. This state only includes those connections that are eligible to dynamic arbitration. The state can optionally include the connection group that is served by allocation, in which case the arbiter must decide on them with higher priority. In fig. 3 we see the freedom of choice the scheduler has depending on whether a slot was allocated for CID, the priority group or dynamically assigned (destination port only). This architecture allows mixing the service for real-time traffic to allocated plus dynamic. The resulting hybrid arbitration performance is discussed later. Pure dynamic arbitration has better mean delay performance, but tight QoS support and per-VC isolation are not solved for VOQ switches in the literature. Thus it is recommended for best-effort traffic.

5 Allocation Performance

In this section we study the performance using allocation with different allocation types and traffic models. Allocation has a number of positive properties (see table 1) which can be quantified for given traffic bounds, i.e. during CAC the model parameters are known for the connections demanding real-time QoS. To study delay distributions and mean values stochastic traffic models are more suitable. Performance results are given analytically and by simulation results using OPNET [15] and a switch model with $M = 16$ ports[8] and $T_{link} = 10^6 c/s$.

5.1 Bounded Input Traffic

When bounded (deterministic) traffic (section 2) is applied, an explicit delay bound is guaranteed (fig. 5). Assuming a fully equidistant service ($COV_s = 0$) with a service rate T_μ higher than *PCR* for CBR or *SCR* for VBR, the time between instances when the queue is empty (busy time) is less than the period of a worst case traffic bounded by the given traffic parameters. During that period there are at least as many allocated slots as arriving cells. A first coarse bound is then given by the number of cells in a burst (assumed to arrive at one instant) times the slot distance.

$$\max(d_{CBR}) \leq BS \cdot T_\mu \tag{12}$$

[8] In the graphs the total offered load ρ means the sum of all connection rates λ_a of any (symmetric) port normalized on the link rate. For each graph the number of connections per port is constant, so $\lambda_a \sim \rho$ holds

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	$PCR = 10Mb/s$, $SCR = 3.3Mb/s$, $MBS = 1000$	$107ms$	$400\mu s$

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

$$\begin{aligned} &\leq (1 + \lfloor CDVT/(T_{PCR} - T_{link}) \rfloor) \cdot T_\mu \\ \max(d_{VBR}) \leq &\quad MBS/(\omega \cdot SCR) \end{aligned} \tag{13}$$

5.1.1 CBR

A more precise 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 over the interval $0..T_\mu$. The distribution of d_b using the assumed worst case traffic is

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

The resulting delay distribution is the convolution of these components ($\prod \otimes \| \| \|$). This is confirmed by simulation results (for $BS = 4$ see fig. 4). The maximum value leads to this closer delay bound:

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

5.1.2 VBR

The situation for VBR is similar. One level of burstiness more makes it more complex, however. For a smooth worst case model [13] the same as for CBR holds:

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

For the heavy VBR model[9] eq. 17 holds.

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

5.2 End-to-end Delay Bounds

With allocation an end-to-end delay guarantee d_N over N hops can be given (fig. 6).

[9] maximal length bursts on link and PCR rate [13]

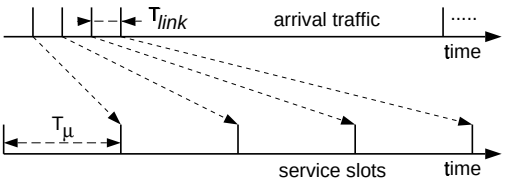

Figure 5: Servicing worst case traffic

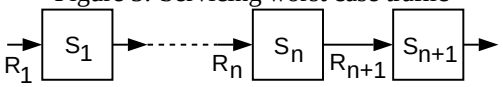

Figure 6: End-to-end traffic bounds

Lemma 1: For any switch n on the route and given CBR[10] input traffic parameters $R_n \sim (\sigma_n, \rho_n)$ the output is shaped such that its traffic is bounded by pairs $R_{n+1} \sim (\sigma_{n+1}, \rho_{n+1})$. Two pairs are of interest: eq. 18 at the service time and eq. 19 at the declared ρ_n.

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

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 an $N = 1$ network.

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.

5.3 Stochastic Input Traffic

Queueing theory is applicable for stochastic traffic. The server statistics can be modelled as E_k[11] with the known values of mean (T_μ) and COV_s of the service time distribution.

Approximations for the performance of a G/G/1 queue can be applied [16] for renewal traffic with given COV_a. When $COV_a = 1$ and $COV_s \to 0$ the heavy-traffic approximation for M/D/1 is applicable.

Results obtained with this model are within the confidence intervals of the simulated values. Due to $\rho(CID) \neq f(\rho)$, $E(q)$ only depends on $\rho(CID)$ which is controlled by ω. Due to Little's Law $E(d)$ decreases with higher traffic rates (fig. 7).

For bursty VBR traffic modelled by a *MMPP*[12] an exact

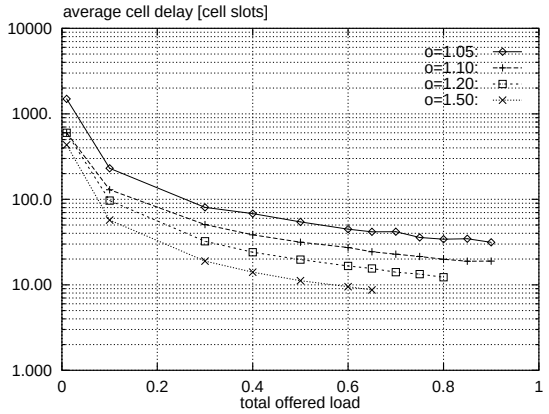

Figure 7: Renewal traffic ($\rho/15$) with $COV_a = 0.4$

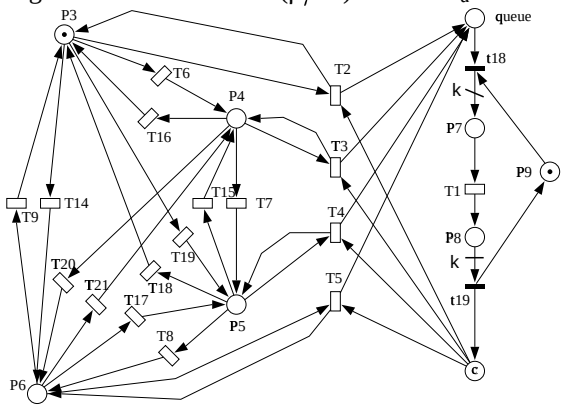

Figure 8: The stochastic Petri-net for solving the $IPP/E_k/1$ System is a special case of the $MMPP/E_k/1$ system displayed here. Left: 4 modulating state places, middle: transitions with state-dependent rate, right: Erlang-Server

solution for the queue size distribution and mean values can be obtained by solving the Markov-chain associated with the stochastic Petri-net in fig. 8 with numeric methods [17].

In an example scenario there are 150 bursty sources per input port having *IPP*[12] characteristics. Every $C = 10$ connections share the same route through the switch. The parameters are: $SCR = \rho \cdot r_{link}/150$, $PCR = 4 \cdot SCR$ (burstfactor $b = 4$), mean burstlength $P = 10$ cells, $\omega = 1.2$ and $S = 16384$ slots. Exploring the choices for allocation and scheduling according to fig. 3, the results for the mean delay for the five offered methods is shown in fig. 9.

With a per-VC allocation (a_only) burst-scale queueing leads to the highest mean delay. But as fig. 10 shows, COV_o is reduced most effectively.

If we allocate the priority group as a whole (prio alloc), the system changes such that an allocation now serves all C connections routed to the same output with a rate C times higher than in the previous case. Simulations indicate that this performs the same as if the CID is allocated but in case of nonexistence an alternative CID is chosen (a+share). The Petri-net analysis yields a mean delay that is C times smaller[13].

In these figures also the dynamic arbitration performance (SIMP [7]) is visible. In this case advantage can be drawn from statistical multiplexing of 150 connections at any port.

The hybrid method (a+dyn) is applied by servicing cells with allocation as well as dynamic arbitration. Its mean delay performance is a trade-off between the pure methods mentioned above. The drawback is that many requests for allocated connections fail when there is no cell available. This probability is shown in fig. 11. With full CID allocation it must be around $1 - \rho(CID)$, the probability for an empty queue.

[10] for VBR substitute $SCR \to PCR$

[11] Erlang-k with $k = \lfloor \sqrt{1/COV_s} \rfloor$ phases

[12] MMPP/IPP=Markov modulated/interrupted poisson process

[13] The result shows the gain for multiplexing C bursty streams together

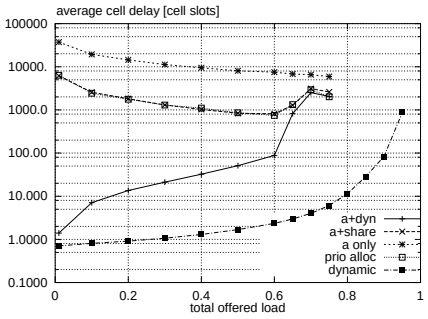

Figure 9: Bursty traffic: $E(d)$

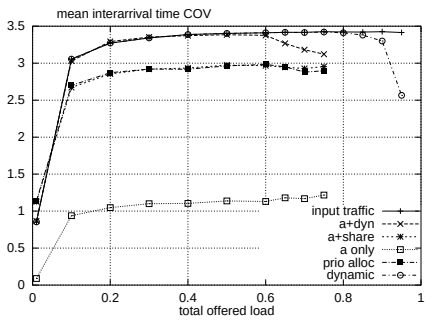

Figure 10: Bursty traffic: COV_o

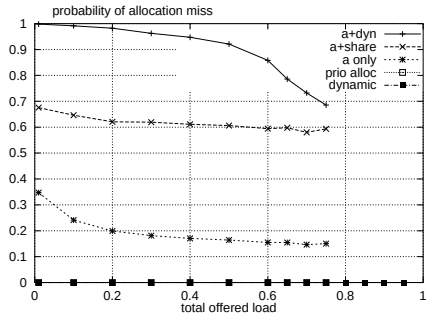

Figure 11: $Pr(CIDmiss)$

5.4 Hybrid Arbitration

The mean delays observed for dynamic arbitration are much smaller than for allocation. The probability distributions, however, differ a lot. So for dynamic arbitration the delay, for which the excess probability is below 10^{-9}, is far beyond the mean, whereas for allocation the mean is half the maximum for worst-case traffic. Due to its allocation component the hybrid method combines the advantages.

Theorem 2: Hybrid arbitration offers the same bounds as pure allocation.

Proof: Hybrid arbitration has at least the service slots at the same reserved positions as pure allocation. Any previous delay bound cannot be exceeded. The additionally granted slots only decrease the mean time until service.

5.5 Shared Allocation

Instead of allocation for connections a whole priority class can be allocated for any (i,o) pair. Bounds can then be given by using the multiplexed traffic bounds [11]

$$R_{mux} \sim (\sigma_{mux}, \rho_{mux}) = (\sum \sigma_i, \sum \rho_i) \qquad (21)$$

For homogeneous sources the resulting bound is comparable to the isolated bounds. The mean delay is reduced. Shaping here also depends on the schedulers, which should select connections in a round-robin manner for best results.

6 Dimensioning and CAC

Given the traffic parameters and the QoS requirements of a traffic source the buffer dimensioning and connection admission control can easily be performed for allocated connections because of its independence of other traffic. CAC for a new connection can be performed in two steps.

First, check whether the QoS requirements to be guaranteed by this switch can be satisfied (eqns. 15, 17). Note that subsequent switches experience a shaped stream $R \sim (\sigma_s, \rho_s(CID))$ (Lemma 1) and therefore do not introduce a delay as high as the first "allocating" switch. When using the CBR approximations $T_\mu \approx T_{PCR}/\omega$ and $T_{PCR} \gg T_{link}$ the check for the first switch simplifies to $d_{max} <^? (T_{PCR} + CDVT)/\omega$.

Second, the required number of slots (eq. 4) must be allocatable within the bounds explained in section 3.

The per-VC buffer size depends on the expected traffic and can be dimensioned as follows.

$$q_{max,CBR} = \lfloor \frac{CDVT}{T_{PCR} - T_{link}} \rfloor \quad q_{max,VBR} = \max(d_{VBR})/T_\mu \qquad (22)$$

By allowing buffer overflow this mechanism is also self-policing: Discarded cells are non-conforming to the GCRA.

7 Conclusion

With allocation a mechanism for the control of the internal routing (scheduling) in VOQ switches has been treated. Due to its independent service for each VC, end-to-end performance guarantees for bounded traffic can be given unregardedly of other connections. Being non-work-conserving in the global sense the delay is comparable to an active shaper. For CBR and most VBR applications this is within the typical QoS requirements. Due to active shaping further experienced delay downstream can be significantly reduced. Together with dynamic algorithms all QoS classes for ATM or IPv6 are supported.

References

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

[2] 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.

[3] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% Throughput in an Input-Queued Switch," *IEEE Transactions on Communications*, 1998.

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

[6] S. Shenker, C. Partridge, and R. Guerin, "Specification of Guaranteed Quality of Service." IETF RFC 2212, 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 Symposium*, 1999.

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

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

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

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

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

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

[14] C. Rose and M. Hluchyj, "The Performance of Random and Optimal Scheduling in a Time-Multiplex Switch," *IEEE Transactions on Communications*, vol. COM-35, p. 813, Aug 1987.

[15] MIL 3, Inc., 3400 International Drive, NW Washington, DC 20008, USA, *OPNET Release 4.0*, 1998. http://www.mil3.com.

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

[17] B. Haverkort, "Matrix-Geometric Solution of Infinite Stochastic Petri Nets," *Proc. Int'l Performance and Dependability Symposium*, 1995.