Efficient fast packet switch fabric with shared input buffers

E. Del Re
R. Fantacci

Indexing terms: High-speed packet switching, Queueing theory

Abstract: The authors deal with an efficient high-speed packet switching fabric suitable for applications in future high speed networks. An advanced implementation architecture based on shared buffers at each input is studied. An important result is that the proposed switching fabric achieves optimum throughput-mean switching delay performance jointly with reduced buffering requirements and without having to resort to a faster switching fabric.

1 Introduction

Future optical-based broadband integrated services digital networks (B-ISDNs) will be suitable for the transmission of information at rates greater than 100 Mb/s. The concept of B-ISDN has undergone considerable discussion and evolution. The asynchronous transfer mode (ATM) is considered to be the ground on which B-ISDN is to be built. In ATM systems, all information to be transmitted is organised in fixed-size packets named cells.

ATM networks are characterised by advanced switching techniques able to handle multimedia traffic. In particular, the fast packet switching (FPS) technique seems to be a promising approach for such networks. In any FPS switching fabric the cell routing is usually based on hardware techniques by making use of the information contained in the header of each cell. The main problem to be solved is the output conflict which occurs whenever two or more cells arrive simultaneously at different switch inputs and require to be routed to the same output. Only one of these cells achieves routing. In order to avoid loss, queueing is necessary for the other to wait for the next routings.

Two classic alternatives to queue the unrouted cells are input queueing and output queueing. Switching fabrics with input queueing are quite simple in architecture but unfortunately, the maximum possible throughput is bounded at 0.586 due to the head-of-line blocking problem [1]. Switching fabrics using output queueing avoid this drawback and achieve optimal delay throughput performance, in particular the maximum possible throughput approaches 1 as the mean arrival rate of cells per slot p approaches 1.

The main problem which arises with output queueing is the requirement of a faster switching fabric. In the worst case of N cells that simultaneously require to be routed to the same output the switch fabric has to operate N times faster than the input–output links. This requirement represents the main drawback to a widespread use of output queueing switching fabrics in high speed networks.

This paper mainly deals with a switching fabric based on the multiple queueing approach [2], implemented at each input of the switch fabric by means of shared buffers.

2 A multiple queueing switching fabric with shared buffers

In the switch fabric under consideration (Fig. 1) all the cells which arrive at the same input share the same buffer (SB) and are logically separated in distinct queues, one for each possible output destination. Each memory address of the SB can be allotted to any output as the occasion demands, not permanently to one particular output. This permits a remarkable improvement of the cell loss probability over classic switching fabric also using output queueing.

For the switching fabric shown in Fig. 1 whenever a cell is stored in an input SB a routing request packet (RRP) is broadcast over the associated bus, one for each input, to all the output controllers (arbiters in Fig. 1). Collisions over the same bus are impossible because at almost one cell may arrived per slot at each input port.

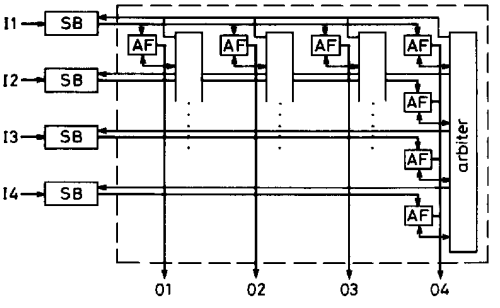

Fig. 1 *Proposed switch fabric architecture ($N = 4$)*

This work was carried out with financial support from the National Research Council (CNR) in the context of the Telecommunication Project and partially by the Italian Ministry of University, Scientific Research and Technology (MURST).

Paper 9562I (E7), first received 23rd September 1992 and in revised form 11th March 1993

The authors are with Dipartimento di Ingegneria Elettronica, Università di Firenze, Via S. Marta, 3 50139 Firenze, Italy

Each RRP contains: the address of the switch output to which the cell is destined; a single activity bit, to inform arbiters about the presence (logic 1) or absence (logic 0) of RRPs to be processed at their inputs; and the SB address where the cell has been stored.

At the beginning of each time slot, the path through each of the N address filters (AF in Fig. 1) is open, initially allowing all arriving RRPs to pass through to the arbiters. The output address bits for each arriving RRP are compared bit-by-bit against the output address of all AFs, one for each possible output port. If at any time the address of a RRP differs from that of an AF, the further progress of the RRP to the arbiter is blocked. That is the output of the AF is set at logic 0 for the remainder of the time slot. By the end of the output address, the AF will have either blocked the RRP, and hence also set its activity bit to 0, or allowed to the RRP to continue on to the arbiter. Note that even though a portion of the address bits of a blocked RRP passing through the filter, these bits are no longer processed by the arbiter as the activity bit of that RRP has been set to 0.

The basic arbiter configuration can be realised using a simple first in first out (FIFO) buffer. Any new arriving RRP is stored in the FIFO buffer to form a routing requests queue (RQ).

From the above, it follows that the switching operation is performed here in two stages. In stage one, the RRP associated with each cell is analysed while in stage two the cell itself is transmitted into the output link, whenever the associated RRP reaches the head of the appropriate RQ. The performance analysis of the switching fabric with multiple queueing and SB at each input (MQSB switch) discussed above have been derived by making use of well known results for the discrete-time queueing system [4–7]. Let us assume that the arrival processes at the N input links, as N independent Bernoulli processes, with the probability of an arrival per slot equal to p. Each cell has an equal probability to be addressed to any of the other N output links and successive cells require independent routing.

The N RQs have been modelled as N discrete G/D/1/N/N queueing systems with arrivals in batches of random size. The validity of this assumption will be verified later by comparing theoretical and simulation results.

From the previous considerations it follows immediately that each input queue can be modelled as a Geom/G/1 queueing system with the service time per cell equal to the total delay spent by the corresponding RRP in the RQ.

Fixing our attention on a particular input queue (the tagged input queue), the imbedded Markov chain approach developed in studying the M/G/1 model is also applicable here for deriving the mean total delay per cell. The probability generating function of the number of cells in the tagged input queue, assuming equilibrium, is derived in Appendix 7.1 as

$$Q(z) = \frac{Q_0 A(z)(z-1)}{z - A(z)} \tag{1}$$

where Q_0 is the probability of having an idle queue and $A(z)$ is the probability generating function of the number of arrivals during the service period of a customer. We have also (Appendix 7.2)

$$A(z) = (1 - p + pz)G(1 - p + pz) \tag{2}$$

where $G(z)$ is the probability generating function of the waiting time (normalised to the cell duration time) or equivalently in our case, the probability generating function of the total time spent by RRPs waiting to reach the head of the RQ.

In deriving an expression for $G(z)$ it must be taken into account that in the case considered, the RRPs may arrive at the appropriate RQ in batches of random size [2, 5, 7]. In particular, we assume that RRPs which arrive at the RQ at a same instant are served in a random order while RRPs which arrived earlier, are served first on the basis of the FIFO discipline. Therefore, the total time spent in the RQ waiting for service by any routing request is due to the sum of two terms. The first term takes into account the time necessary to serve all the RRPs which are waiting in the queue at the arrival instant, while the second term is an additional delay due to the service of the RRPs which arrived at the same instant and were randomly selected to be served first.

Let us assume that k cells are already waiting for routing in a particular RQ (the tagged RQ), the probability that a RRP (the tagged RRP) arrives in a batch of size i is given by

$$P(i \mid k) = \frac{i}{(N-k)\alpha} \binom{N-k}{i} \alpha^i (1-\alpha)^{N-k-i} \tag{3}$$

where α is the probability of having a cell from one of the input queues requesting routing to the tagged output link.

The probability generating function of the waiting time, on condition that k requests are waiting for routing in the tagged RQ and that the tagged RRP arrives in a batch of size i is

$$G(z \mid i, k) = \sum_{j=0}^{i-1} \frac{z^{j+k}}{i} = \frac{1 - z^i}{i(1-z)} z^k \tag{4}$$

Therefore, $G(z \mid k)$ is given by

$$\begin{aligned} G(z \mid k) &= \sum_{i=1}^{N-k} G(z \mid i, k) P(i \mid k) \\ &= \frac{1 - (1 - \alpha + \alpha z)^{N-k}}{(N-k)\alpha(1-z)} z^k \end{aligned} \tag{5}$$

The probability $P_R(k)$ of having k RRPs, $0 \leqslant k \leqslant N-1$, in the RQ can be obtained numerically by an application of the Markov chain balance equations. Fig. 2 shows an

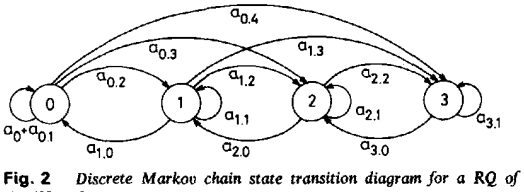

Fig. 2 *Discrete Markov chain state transition diagram for a RQ of size ($N = 4$)*

example of the Markov chain to be considered when $N = 4$. The final results are

$$P_R(1) = P_R(0) \frac{(1 - a_{0,0} - a_{0,1})}{a_{1,0}} \tag{6}$$

$$P_R(k) = \frac{1 - a_{k-1,1}}{a_{k,0}} P_R(k-1) - \sum_{i=2}^{k} \frac{a_{k-i,i}}{a_{k,0}} P_R(k-i) \qquad 2 \leqslant k \leqslant N-1 \tag{7}$$

with $P_R(0)$ determined to verify the following equation

$$\sum_{k=0}^{N-1} P_R(k) = 1 \tag{8}$$

and the terms $a_{i,j}$ (Fig. 2), given by

$$a_{i,j} = \binom{N-i}{j} \alpha^j (1-\alpha)^{N-i-j} \tag{9}$$

Therefore, the probability generating function of the waiting time $G(z)$ can be derived as a function of α as:

$$G(z) = \sum_{k=0}^{N-1} G(z \mid k) P_R(k)$$
$$= \sum_{k=0}^{N-1} \frac{[1-(1-\alpha+\alpha z)^{N-k}]}{(N-k)\alpha(1-z)} P_R(k) \tag{10}$$

In deriving an expression for α we define: A_m as the overall number of RRPs arrived at all the RQs at the beginning of the mth time slot; and F_m as the number of free input queues at the beginning of the mth time slot.

According to our assumptions, an input queue is free at the mth time slot if it is idle or if the cell at its head has been selected to be routed at the beginning of the $(m-1)$th time slot. It is evident that an arrival at a RQ must come only from a free input queue. It follows that

$$P\{A_m = j\} = \binom{F_m}{j} (N\alpha)^j (1-N\alpha)^{F_m} \tag{11}$$

Therefore, the mean number of arrivals is

$$A_m(F_m) = F_m N\alpha \tag{12}$$

Letting H_m denote the number of RRPs in the tagged RQ, we can write

$$F_m = N - H_m \tag{13}$$

Therefore, in a steady state condition

$$\bar{F} = N - \bar{H} \tag{14}$$

where the mean number of RRPs in the tagged RQ can be derived as

$$\bar{H} = \sum_{n=0}^{N-1} n P_R(n) \tag{15}$$

By assuming equilibrium we also have:

$$(N-\bar{H})N\alpha = Np \tag{16}$$

Hence

$$\alpha = \frac{p}{N - \sum_{n=0}^{N-1} n P_R(n)} \tag{17}$$

Eqn. 17 defines a nonlinear equation in α. Solving this equation numerically it is possible to determine α.

Starting from the previous considerations, it is shown in Appendix 3 that the mean delay per cell (normalised to the cell duration time) for the FIFO selection policy is

$$T = 1 + \sum_{k=0}^{N-1} \frac{(N+k-1)\alpha}{2} P_R(k)$$
$$+ p\left\{ \frac{\sum_{k=0}^{N-1} k[k-1+\alpha(N-k-1)] P_R(k)}{2N - p\left[2 + \sum_{k=0}^{N-1} (N+k-1)\alpha P_R(k)\right]} \right.$$
$$\left. + \frac{\sum_{k=0}^{N-1} (N-k-1)(N-k-2)\alpha^2 P_R(k)}{3\left\{2N - p\left[2 + \sum_{k=0}^{N-1} (N+k-1)\alpha P_R(k)\right]\right\}} \right\} \tag{18}$$

Fig. 3 shows T as a function of p for different values of N. It is evident that the maximum possible throughput approaches 1 as p approaches 1. Simulation results have been also reported in Fig. 3 to highlight the good agreement.

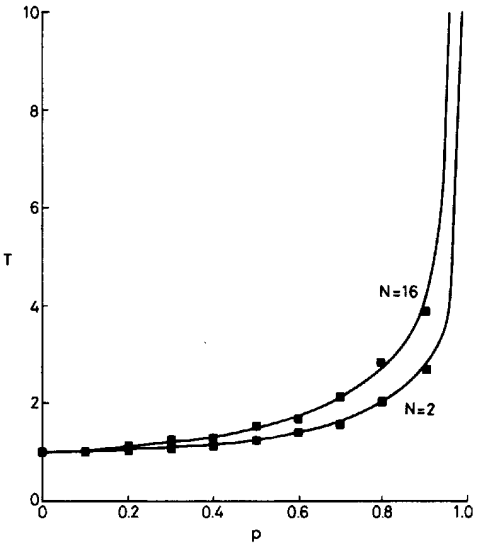

Fig. 3 *Mean normalised total delay*
——— analytical results
■ simulation results

The simulation results shown herein have been obtained by means of simulation programs based on the SIMSCRIPT programming language [3]. Confidence intervals for the simulation results have been derived by the method of independent replications. The simulation is run J independent times and J estimates are thus obtained for each performance measure of interest (i.e. mean switching delay or cell loss probability) [8]. Each set of J values thus represents J independent samples of the parameter to be estimated and standard statistical techniques can be used to derive the confidence interval [9]. In our simulation approach we have set J equal to 10. The confidence intervals for the simulation results so obtained are very tight and, therefore they are not quoted in the figures.

Fig. 4 shows T as a function of p for the single queueing on inputs and for the proposed multiple queueing on inputs in comparison with that obtained by using the output queueing approach [4, 17]. It is evident in this figure that the proposed multiple queueing approach achieves the same performance as the output queueing approach without resorting to a more complex switching fabric [1, 18].

3 Finite buffer analysis

The case of a SB of finite capacity is now considered. A parameter that is of particular interest in this case is the cell loss probability. Unfortunately, for the switching system under consideration, the analtyical evaluation of the cell loss probability leads to a queueing problem that is too complex to be solved in a closed form. However, an approximation of the cell loss probability can be derived. The validity of this approximation will be verified later by comparing analytical and simulation results.

We start our analysis by considering SBs of infinite size. Unfortunately, it was shown in Reference 19 that the number of cells forming the logical queue for a particular

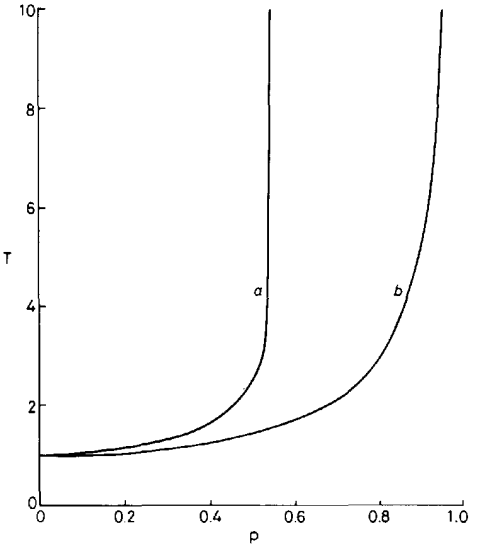

Fig. 4 *Mean total delay comparison ($N = 16$)*

a input queueing
b output queueing and multiple queueing

output is not independent of the number of cells forming the other $N - 1$ logical queues. To simplify our analysis, we consider the overall number of cells stored in the SB as sum of N independent identically distributed (i.i.d.) random variables n_i, denoting the number of cells forming the queue for output i ($1 \leqslant i \leqslant N$). Note that this simplified approach is consistent with that used in Reference 17 for deriving the performance of a switch fabric with a completely shared (output) buffering. Under this assumption, it is shown in Appendix 7.4 that the probability generating function of n_i is

$$B_i(z) = \sum_{k=0}^{\infty} z^k q_i(k) = B^N(z) = \left[\sum_{j=0}^{\infty} z^j q(j)\right]^N \tag{19}$$

where $B(z)$ is the probability generation function of the number of cells in one of the N logical queues, $q(j)$ is the probability of having j cells in a logical queue, and $q_i(k)$ is the probability of having k cells stored in the SB. In deriving an expression for $B(z)$ we again resort to the imbedded Markov chain approach. The imbedded Markov chain to be considered is shown in Fig. 5 where

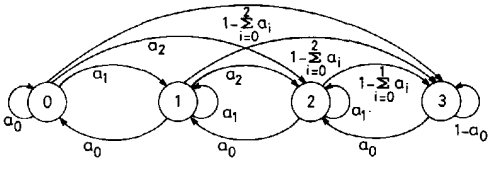

Fig. 5 *Discrete Markov chain transition diagram for multiple queueing input queue*

the instants of service completion have been assumed as the imbedded points. In this figure the terms a_i are defined as

$$a_i = \sum_{m=1}^{\infty} \binom{m}{i} (p/N)^i (1 - p/N)^{m-i} p(m) \tag{20}$$

where $p(m)$ denotes the probability of the cell at the head of the queue having a service time equal to m (slots).

Fig. 6 shows, as an example, the probabilities of the service time for the case $N = 16$ and $p = 0.8$. The terms

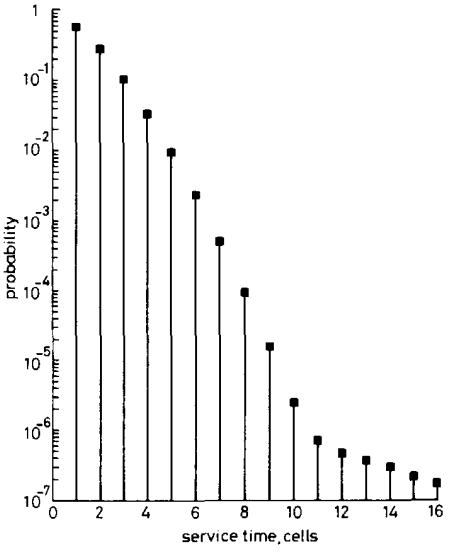

Fig. 6 *Service time probability ($N = 16$, $p = 0.8$)*

$q(k)$ can be derived as

$$q(i) = q(0)\frac{(1 - a_0 - a_1)}{a_1} \tag{21}$$

$$q(k) = \frac{1 - a_1}{a_0} q(k-1) - \sum_{i=2}^{k} \frac{a_i}{a_0} q(k-i) \qquad 2 \leqslant k \leqslant N-1 \tag{22}$$

with, as usual, $q(0)$ defined to verify the following equation

$$\sum_{k=0}^{\infty} q(k) = i \tag{23}$$

Therefore, it is possible to numerically derive the probability of having n cell stored in the SB. In our approach we approximate the cell loss probability P_B for a SB of capacity L (cells) as the probability of having a number of cell greater than L stored in a SB of infinite size. Figs. 7–9 show the derived approximation in comparison with the cell loss probability achieved by using the output queueing approach as a function of the buffer size (cells) for different values of p. These figures clearly point out that the proposed MOSB switch outperforms the classical output queueing switch. In Fig. 10 our approximation is compared with the simulation results for the case $N = 16$ and $p = 0.9$. It is evident in this figure that

our approximation is tight for low values of the cell loss probability (typically, the values of interest).

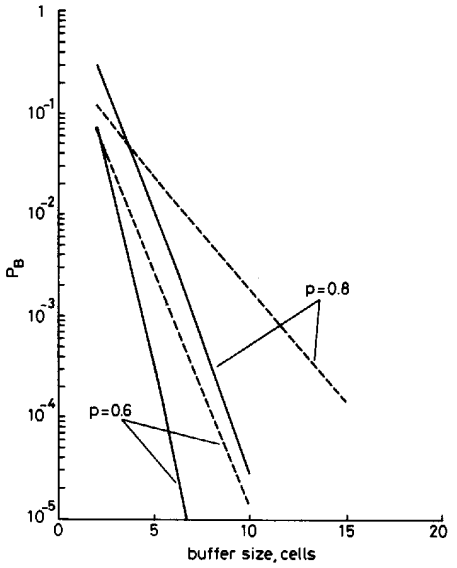

Fig. 7 *Cell loss probability comparison ($N = 8$)*
—— MQSB switch
- - - - output queueing switch

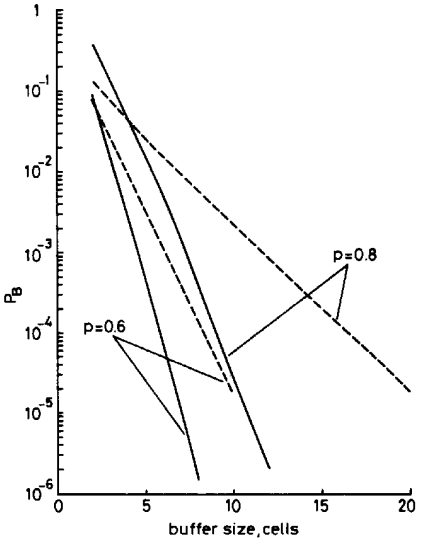

Fig. 8 *Cell loss probability comparison ($N = 16$)*
—— MQSB switch
- - - - output queueing switch

4 Performance comparison with the knockout switch

The knockout switch was proposed in Reference 20 for a pure packet-switched environment. This type of switch uses one broadcast input bus from every input port to all output ports as shown in Fig. 11. By means of packet filters, each knockout concentrator receives only the cells for the associated output. The knockout concentrator uses a suitable algorithm to select up to L cells from the N incoming links to the knockout concentrator. The selected cells are stored in and removed from the output shared buffer according to the order of their arrival by means of the shifter equipment.

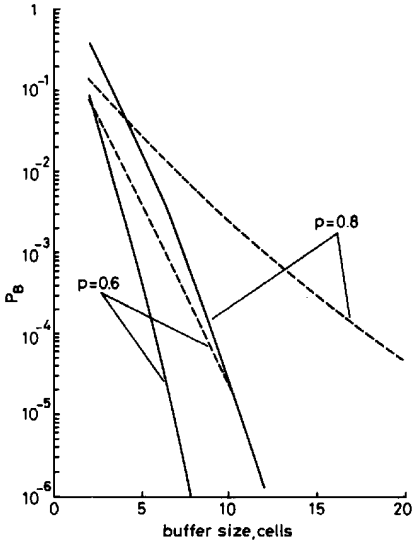

Fig. 9 *Cell loss probability comparison ($N = 32$)*
—— MQSB switch
- - - - output queueing switch

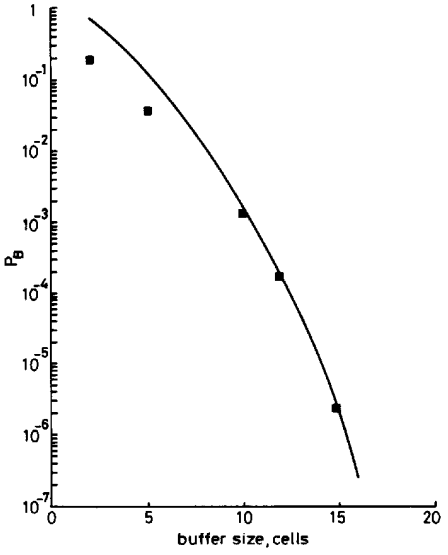

Fig. 10 *Cell loss probability comparison ($p = 0.9$)*
—— upper bound
■ simulated results

To permit a fair comparison between the knockout switch performance and that obtained through the switch fabric discussed in Section 3, it is necessary to derive the mean switching delay and the cell loss probability attained by the knockout switch. We start our analysis

by deriving an expression for T. By means of the knockout concentrator the probabilities of arrivals are modified as

$$a_i = \binom{N}{i}\alpha^i(1-\alpha)^{N-i} \quad i = 0, 1, \ldots, L-1 \tag{24}$$

$$a_L = \sum_{i=L}^{N}\binom{N}{i}\alpha^i(1-\alpha)^{N-i} \tag{25}$$

$$a_i = 0 \quad i = L+1, L+2, \ldots, N \tag{26}$$

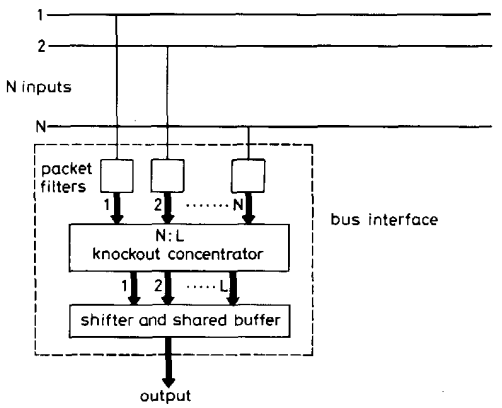

Fig. 11 *Basic structure of knockout switch*

where we have assumed a reduction from N to L and α equal to p/N. The probability generating function (PGF) results in

$$A(z) = \sum_{i=0}^{L} z^i a_i \tag{27}$$

Following a standard approach in queueing analysis [6, 20] we obtain the PGF for the steady queue size as

$$Q(z) = \frac{q(0)a_0(1-z)}{A(z)-z} \tag{28}$$

with the term $q(0)$ derivable by eqns. 21–23 where the terms a_i is given by eqns. 24–26.

We are interested in deriving an expression for the total switching delay. In performing our analysis we model the knockout switch as a classical switch with output queueing [4, 17] i.e. as a discrete G/D/1 queueing system with arrivals in batches of random sizes [5]. This leads us to assume that all cells arriving at the same output queue in a time slot are served in random order. However, all cells arriving in earlier time slots are served first, according to the FIFO selection policy, within batches. It thus follows that the mean switching delay has three components:

(i) The cell service time, equal to one time slot

(ii) The time (u_1) that must have elapsed before our cell reaches the head of the queue to be served

(iii) The time (u_2) necessary to serve the cells which arrived in the same batch, but which were selected to be served first

Starting from the previous considerations it was found [4, 17] that the mean normalised switching delay results in

$$T = 1 + \sum_{i=1}^{\infty}\frac{iq(i)}{p} + \frac{\sum_{i=2}^{L} i(i-1)a_i}{2p} \tag{29}$$

Fig. 12 shows the parameter T as a function of p for $N = 64$ and $L = 8$.

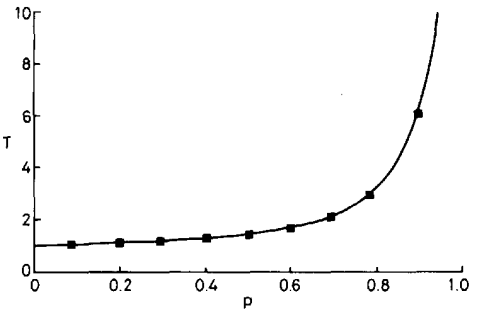

Fig. 12 *Normalised switching delay for knockout switch as a function of p for N = 64 and L = 8*

analytical results
■ simulation results

The analytical evaluation of the cell loss probability is the subject of the remainder of this section. A cell may be lost in the knockout switch when it arrives in a batch of size greater that L and loses the knockout competition, or when it successfully passes through the knockout concentrator and finds the output buffer full. From above it follows that

$$P_B = 1 - \frac{q(0)a_0}{p} \tag{30}$$

where $q(0)$ is derivable from eqns. 21–23 with the arrival probabilities defined as

$$a_{ij} = \begin{cases} a_j & i+j \leqslant M \\ \sum_{k=j}^{L} a_k & i+j = M+1 \end{cases} \tag{31}$$

for $L > M$, likewise for $L \leqslant M$

$$a_{ij} = \begin{cases} a_j & i+j \leqslant M \quad j \leqslant L \\ 0 & j > L \\ \sum_{k=j}^{L} a_k & i+j = M+1 \quad j \leqslant L \end{cases} \tag{32}$$

Figs. 13 and 14 show P_B as a function of the buffer size M for different values of N and p. The mean switching delay can be also derived for the case of a finite buffer

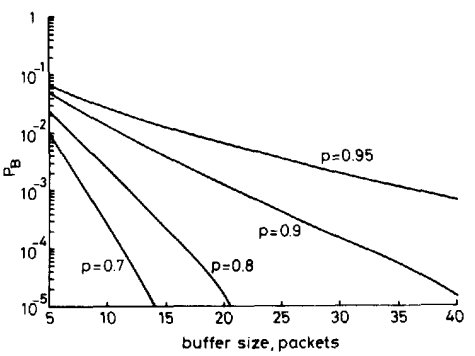

Fig. 13 *Cell loss probability for knockout switch as a function of output buffer size for N = 16, L = 8 and different values of p*

size. The parameter T can be obtained by the Little formula as

$$T = 1 + \frac{\sum_{i=1}^{M} iq(i)}{p(1 - P_B)} \tag{33}$$

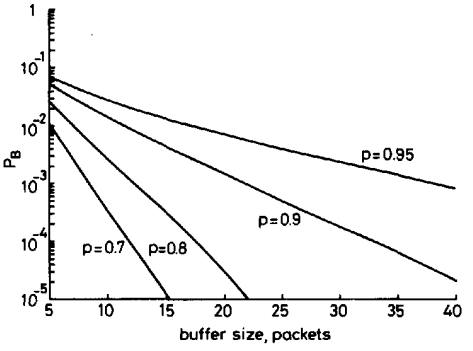

Fig. 14 *Cell loss probability for knockout switch as a function of output buffer size for $N = 64$, $L = 8$ and different values of p*

Fig. 15 shows T as a function of p for $N = 64$ and different values of the buffer size M. Fig. 16 shows P_B as a function of the buffer size for $p = 0.9$ in comparison with its values attained for the MQSB switch. An implementation complexity comparison in terms of the total number of gates is given in Table 1. The values reported in Table 1 do not account for the implementation complexity of the SB for the MQSB switch (input) and the knockout switch (output). In particular, in deriving the implementation complexity of the knockout switch we have used the approach outlined in Reference 20. Fig. 16 and Table 1 clearly highlight the advantages of the MQSB switch over the knockout switch.

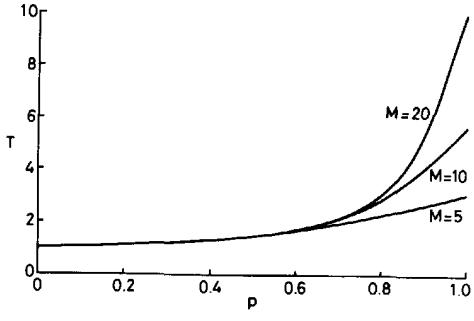

Fig. 15 *Normalised switching delay for knockout switch as a function of p, for $N = 64$, $L = 8$ and different values of output buffer size*

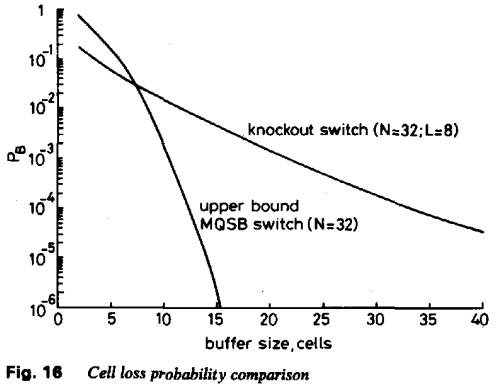

Fig. 16 *Cell loss probability comparison*

Table 1: Implementation complexity comparison

Inputs/outputs N	Knockout switch $L = 8$ (gates/output)	MQSB switch (gates/output)
16	2128	80
32	4256	160
64	8512	320
128	17024	640

Different output queueing switching fabrics with completely shared buffering have been proposed recently. In References 17 and 21 the completely shared buffering is obtained by increasing the switch fabric size. The switching fabric proposed in References 17 and 21 permits one to save on the total amount of buffering needed to achieve a specified cell loss probability, but the required increase in the size of the switch fabric may become unacceptable for some applications. A switching fabric with input and output buffering has been also proposed [22]. With respect to the classical input queueing approach, the joint input–output queueing approach permits a number of cells greater than one to be routed to the desired output per time slot. However, the maximum value of this number is less than N, in the case of a $N \times N$ switching fabric, making it possible to reduce the speed-up in the switching operation in comparison with the classical implementation of the output queueing. With respect to the switching fabric proposed in Reference 22, the MQSB switch permits us to save on the total amount of buffering and to reduce the implementation complexity. Another switching fabric based on a two-stage switching approach similar to that used in the MQSB switch is the staggering switch described for a special application in Reference 23. In the staggering switch the two stages of the switching operation are named the scheduling stage and the switching stage, respectively. Each of the stages is implemented by means of a nonblocking switching fabric. Considering an $N \times N$ switching fabric, the scheduling stage is an $N \times M$ nonblocking switch while the switching stage is an $M \times N$ nonblocking switch, with $M \geqslant N$. The scheduling stage is connected to the switching stage by M delay lines. The scheduling stage distributes the cells arriving at the switch inputs to the delay lines to avoid two cells arriving at the switching stage being destined for the same output. In comparison with the staggering switch, the MQSB switch again exhibits a lower implementation complexity and achieves a better performance. We can validate this affirmation with an example. Let us consider a 16×16 staggering switch with $M = 16$ at $p = 0.8$. The cell loss probability P_B results equal 1.2×10^{-3} where for the MQSB switch with SBs of size 8 (cells) P_B is equal to 2.8×10^{-5}.

5 Conclusions

In this paper a novel switching fabric suitable for applications in future high speed networks has been described and analysed. An important result is that the optimum delay-throughput performance can be achieved, without having to resort to a switch fabric which runs N times

faster than the input and output links. A performance comparison with the knockout switch and other switching fabrics has been also presented to highlight the better performance of the MQSB switch.

6 References

1 AHMADI, H., and DENZEL, W.E.: 'A survey of modern high-performance switching techniques', *IEEE J. Sel. Areas Commun.*, 1989, **7**, pp. 1091–1103
2 DEL RE, E., and FANTACCI, R.: 'A fast packet switching satellite communication network'. IEEE INFOCOM '91, Miami, FL, 1991
3 RUSSEL, E.C.: 'Building simulation models with SIMSCRIPT II.5'. CACI, Los Angeles, USA, 1983
4 KAROL, M.J., HLUCHYJ, M.G., and MORGAN, S.P.: 'Input versus output queueing on a space-division packet switch', *IEEE Trans.*, 1987, **COM-35**, (12), pp. 1347–1356
5 MEISLING, T.: 'Discrete-time queueing theory', *Oper. Res.*, 1958, **6**, pp. 99–105
6 KOBAYASHI, H., and KONHEIM, A.G.: 'Queueing models for computer communications system analysis', *IEEE Trans.*, 1977, **COM-25**, (1), pp. 2–28
7 BURKE, P.J.: 'Delays in single-server queues with batch input', *Oper. Res.*, 1975, **23**, pp. 830–833
8 ROBERTAZZI, T.G.: 'Computer networks and systems: queueing theory and performance evaluation' (Springer-Verlag, New York, USA, 1990)
9 LAW, A.M.: 'Statistical analysis of simulation output data', *Oper. Res.*, 1983, **31**, pp. 983–1029
10 KLEINROCK, L.: 'Queueing system' (Wiley, New York, 1975), Vol. 1
11 HAYES, J.F.: 'Modeling and analysis of computer communications networks' (Plenum Press, New York, 1984)
12 SCHWARTZ, M.: 'Telecommunication network: Protocols, modeling and analysis' (Addison-Wesley, Reading, MA, 1987)
13 BERTSEKAS, D., and GALLAGER, R.: 'Data network' (Prentice-Hall, Englewood Cliffs, NJ, 1987)
14 HAMMOND, J.L., and O'REILLY, P.J.P.: 'Performance analysis of local computer networks' (Addison-Wesley, Reading, MA, 1986)
15 TANENBAUM, A.S.: 'Computer networks' (Prentice-Hall, Englewood Cliffs, NJ, 1989)
16 HUI, J.Y., and ARTHURS, E.: 'A broadband packet switch for integrated transport', *IEEE Trans.*, 1987, **SAC-5**, (8), pp. 264–1273
17 HLUCHYJ, M.G., and KAROL, M.J.: 'Queueing in high-performance packet switching', *IEEE Trans.*, 1988, **SAC-6**, (9), pp. 1587–1597
18 HUI, J.Y.: 'Switching and traffic theory for integrated broadband networks' (Kluwer, Norwell, MA, 1990)
19 ECKBERG, A.E., and HOU, T.-C.: 'Effect of output buffer sharing requirements in an ATDM packet switch'. IEEE INFOCOM '88, pp. 459–466
20 YEH, Y.S., HUCHYJ, M.G., and ACAMPORA, A.S.: 'The knockout switch: a simple, modular architecture for high-performance packet switching', *IEEE Trans.*, 1987, **SAC-5**, (8), pp. 1274–1283
21 HUANG, A., and KNAURER, S.: 'Startlite: a wideband digital switch'. IEEE GLOBECOM '83, 1983, pp. 45–50
22 OIE, Y., MURATA, M., KUBATA, K., and MIYAHARA, H.: 'Performance analysis of nonblocking packet switch with input and output buffers', *IEEE Trans.*, 1992, **COM-40**, pp. 1294–1297
23 HAAS, Z.: 'Staggering switch: an almost-all optical packet switch'. IEEE Globecom '92, 1992, pp. 1593–1599

7 Appendix

7.1 Derivation of probability generating function $Q(z)$

In this Appendix the probability generating function $Q(z)$ of the number of customers in a Geom/G/1 queueing system is derived. Time is divided into slots of equal length. We assume that arrivals are characterised by a Bernoulli process with the probability of having an arrival per slot equal to p and that the service discipline is any nonpreemptive work conserving discipline. Nonpreemptive means that once a customer enters service, he reaches service completion before a new customer is selected to be processed. The method of imbedded Markov chains [5, 6, 10–15] can be used to derive $Q(z)$. In particular, the imbedded points are assumed as the instants of service completion for customers. Let q_i be the number of customers in the system just after the service completion of the ith customer and let a_i be the number of customers entering the system during the service of the ith customer, we have

$$q_{i+1} = q_i - u(q_i) + a_{i+1} \tag{34}$$

where $u(q_i)$ is

$$u(q_i) = \begin{cases} 1 & \text{if } q_i > 0 \\ 0 & \text{otherwise} \end{cases} \tag{35}$$

The probability generating function of the stationary distribution of q_i can be defined as

$$\begin{aligned} Q(z) &= \sum_{k=0}^{\infty} \Pi_k z^k \\ &= \lim_{i \to \infty} E\{z^{q_{i+1}}\} \\ &= \lim_{i \to \infty} E\{z^{a_{i+1}}\} E\{z^{q_i - u(q_i)}\} \end{aligned} \tag{36}$$

In eqn. 36 we have used the independence of q_i and a_i. If we define $A(z)$ to be the probability generating function of the number of arrivals during a service period of a customer, from eqn. 36 we have

$$Q(z) = A(z)\{Q_0 + z^{-1}[Q(z) - Q_0]\} \tag{37}$$

where Q_0 denotes the probability of having an idle system. Hence eqn. 1 follows immediately from eqn. 37.

7.2 Derivation of eqn. 2

Let $A(z)$ be the probability generating function of the number of arrivals during a service period of a cell. For the discrete queueing system defined in Section 2 we have

$$\begin{aligned} A(z) &= \sum_{j=1}^{N} \sum_{k=0}^{j} z^k \binom{j}{k} p^k (1-p)^{j-k} b_j \\ &= \sum_{j=1}^{N} (1 - p + pz)^j b_j \\ &= B(1 - p + pz) \end{aligned} \tag{38}$$

where

$$B(z) = \sum_{j=1}^{N} b_j z^j \tag{39}$$

is the generating function of the service time distribution $\{b_j\}$.

The cell service time for the Geom/G/1 queueing system defined in Section 2 is given by the cell transmission time on the outgoing link (one slot) plus the total time the PRR spends in the appropriate RQ waiting for service. Let $G(z)$ be the generating function of the PRR waiting time distribution $\{g_i\}$ we have

$$B(z) = zG(z) \tag{40}$$

Hence, from eqns. 38 and 40 we easily obtain eqn. 2.

7.3 Derivation of eqn. 18

The generating function $Q(z)$ defined by eqn. 1 allows us to find the moments of the distribution of the number of cells in the queue. We start our analysis by writing eqn. 1 as

$$Q(z)[z - A(z)] = Q_0 A(z)(z - 1) \tag{41}$$

Differentiating eqn. 41 successively we have

$$Q'(z)[A(z) - z] + Q(z)[A'(z) - 1] = -Q_0 A(z) + Q_0(1 - z)A'(z) \quad (42)$$

$$Q''(z)[A(z) - z] + 2Q'(z)[A'(z) - 1] + Q(z)A''(z) = -2Q_0 A'(z) + Q_0(1 - z)A''(z) \quad (43)$$

where $C'(z)$ and $C''(z)$ denote the first and second derivative of $C(z)$ with respect to z.

We let $z = 1$, since $A(1) = P(1) = 1$ and the mean number of cells in the queue is equal to $Q'(1)$ we have

$$E[q] = \frac{Q_0 A'(1)}{1 - A'(1)} + \frac{A''(1)}{2[1 - A'(1)]} \quad (44)$$

where $E[c]$ denotes the mean value of c.

The unknown term Q_0 in eqn. 44 can be derived through eqn. 34. Under the assumption that a steady-state condition exists, by taking expections on both sides of eqn. 34 we have

$$\lim_{i \to \infty} E[q_{i+1}] = \lim_{i \to \infty} E[q_i - u(q_i) + a_{i+1}] \quad (45)$$

hence

$$E[u(q)] = E[a] = A'(1) \quad (46)$$

The term $u(q)$ defined by eqn. 35 in a steady-state condition can be considered as the indicator function of the event that the number of cells in the queue is greater than 0. Accordingly we have

$$E[u(q)] = 1 - Q_0 \quad (47)$$

Therefore, eqn. 44 can be rewritten as

$$E[q] = A'(1) + \frac{A''(1)}{2[1 - A'(1)]} \quad (48)$$

with $A'(1)$ and $A''(1)$ derived by successively differentiating eqn. 2 and setting $z = 1$. By using the Little formula from eqn. 48 we obtain eqn. 18.

7.4 Probability generating function of n_t

Let us consider N independent identical Geom/G/1 queueing systems and let λ be the probability of an arrival at one of these N queueing systems. The event of an arrival at a particular queueing system is assumed to be independent of arrivals at the other $N - 1$ queueing systems. The overall number of cells in the N queues under a steady-state condition can be defined as a sum of N independent and identically distributed random variables as

$$n_t = n_1 + n_2 + \cdots + n_N \quad (49)$$

each term n_i in eqn. 49 denotes the number of cells in the queue i. Let us assume that the generating function of n_t as $B_t(z)$ and of each n_i as $B_i(z)$. Recalling that:

(i) The N random variables n_i are independent identically distributed random variables, hence

$$B_i(z) = B(z) \quad \text{for all } i \quad (50)$$

(ii) the generating function of a sum of independent random variables is the product of generating functions

Thus we can easily derive eqn. 19 from eqns. 49 and 50.