Performance evaluation
of an optical transparent packet switch

P. Cadre* A. Gravey† C. Guillemin‡

France Télécom CNET§

15 September 1996

Abstract

This paper addresses the performance evaluation of an optical transparent packet switch that solves contention using optical delay lines. Output buffers are emulated using scheduling on a small number of delay lines with non-consecutive delays. Several analytical models are presented, and validated by simulation. These models provide efficient bounds for estimating packet loss probability, under the assumption of regular, balanced input traffic. It is shown that the proposed switch architecture allows to achieve a good performance in terms of packet loss, with a number of delay lines significantly smaller than the one currently used in other architectures.

1 Introduction

B-ISDNs are intended to support many services, with variable bandwidth requirements; in particular, the need to support services (such as e.g. interactive multimedia services) generating very high bit rate and bursty traffic is expected to accelerate the demand for flexible, very high bit rate networks.

*Corresponding author. P. Cadre is with France Télécom, LAA/EIA/EVP, 22307 Lannion, FRANCE; phone: 33 2 96 05 18 60, fax: 33 2 96 05 39 45, email Philippe.Cadre@lannion.cnet.fr
†A. Gravey is with France Télécom, LAA/EIA/EVP, 22307 Lannion, FRANCE; phone: 33 2 96 05 39 84, fax: 33 2 96 05 39 45, email Annie.Gravey@lannion.cnet.fr
‡C. Guillemin is with France Télécom, LAB/RIO/ARO, 22307 Lannion, FRANCE; phone: 33 2 96 05 34 67, fax: 33 2 96 05 39 45, email Christian.Guillemin@lannion.cnet.fr
§This study has been partially supported by the European ACTS project KEOPS (Keys for OPtical Switching)
Although transmission technologies, based on optics, do offer a huge bandwidth potential, this potential cannot presently be used by ATM networks with electronic switching systems. Therefore, one can observe a significant effort in designing optical packet networks and associated switching devices, based on photonic, that could take advantage of high bandwidth optical transmission systems.

Packet networks are flexible by nature, since very different types of communications can be thus accommodated; moreover, if the optical packet is transparently transferred, i.e. its payload is maintained in optical format from source to destination, the packet rate may vary from packet to packet, which naturally enhances the flexibility of the network. Obviously, the design of an optically transparent packet network requires revisiting most of the options that have been taken when designing ATM networks and switches [1]. One of the stable options, concerning the design of optical transparent packet networks, is the choice of fixed duration optical packets.

An optical packet is composed of guard times, a header and a payload. Routing information is derived from the packet header which is either optically or electronically processed in the switching fabric, whereas the packet payload is handled transparently. Guard times allow a synchronous operation of switches, independently of the wavelengths carrying optical packets.

One of the major problem that needs to be solved when designing optical switches is contention resolution. Indeed, large and fast random access optical memories are currently not available. Two mains trends to solve contention in optical switches are respectively deflecting packets on alternative routes, and scheduling packets by means of optical delay lines [7].

Several switching architectures based on delay lines are currently proposed. In the switch described in [9], delay lines, with consecutive delays, are used to exactly emulate FIFO output buffers. The technology that is used however sets limits on the depth of the FIFO buffers thus emulated (at most 32 places): this implies either low throughput or poor performance in terms of packet loss even for regular input traffic. Another optical switch, using the same type of delay lines, and which is described in [6], directly emulates a (large) central memory shared by all outputs in a FIFO manner; however, this implementation requires extremely large switching blocks. Another approach described in [2, 5]
allows the emulation of large input buffers, with non consecutive delay lines; however, as shown in [8] the throughput of a switch with input queues is limited due to Head of the Line Blocking.

The intent of this paper is to present preliminary performance evaluation results of an optical switch [4], which also solves contention using optical delay lines. In this particular architecture, FIFO output buffers are emulated by providing access, for each input, to a (small) set of optical delay lines with non consecutive delays. This is done using a two stage switching architecture, extensively using wavelength conversion functions to perform routing within the switch. This architecture is one of the two switch architectures currently considered within the European ACTS project KEOPS “Keys for Optical Switching”.

The present paper addresses the evaluation of the scheduling process that is used to solve contention. It is shown that this process allows to emulate large FIFO output buffers, with only restricted access to the buffering facilities. Practical implementation problems are beyond the scope of this paper, and the interested reader is referred to [4].

Offering only a restricted access to buffering facilities has an obvious drawback: it implies a possible performance degradation in terms of packet loss probability or packet de-sequencing. Exact and approximate queueing models have been developed to analyze the impact of this method (see Sections 4 and 5); the present paper describes these models, and provides, in Section 6, a preliminary set of methods for dimensioning the optical switch.

2 Operation of the optical switch

Let \( n_{\lambda} \) wavelengths be multiplexed on each of the \( n_f \) input or output fibers as shown in Figure 1. Traffic arrives synchronously on each of the \( N = n_f \times n_{\lambda} \) inputs. Let \( \Delta \) denote the duration of a time-slot, which is also the packet transmission time. At time \( i\Delta \), a packet may arrive on each wavelength of each input, and is routed to any wavelength of a given output.

If, at a given time, more than \( n_{\lambda} \) packets need to be routed to a given output, there is contention. Contention is solved by scheduling excess packets using optical delay lines, and thus emulating an
output buffer. If enough delay lines, with consecutive delays, are available, it is possible to exactly emulate FIFO output buffers as shown in Figure 2. This system is straightforwardly modeled by a discrete-time FIFO queue with batch arrivals, services and a constant service time $\Delta$. Since packet headers arrive at instants $i\Delta^+$ and packet payloads services end at times $(i + 1)\Delta^+$, departures take place before arrivals.

Figure 2 graphically represents the emulation of a FIFO output queue of depth $F = 7$, with $n_\lambda = 1$. Optical delay lines, corresponding to consecutive delays $\{i \times \Delta, i = 1, \ldots, F\}$, are used. An excess packet is routed on a delay line of length $r$, where $r$ is chosen as the minimum available delay. A given delay line may be used by several packets, one per slot time; scheduling is performed in such a way that at most one packet appears in any column. With this architecture, each input accesses $F$ delay lines.
In order to limit the cost of the switch, one may wish to limit the total number of delay lines accessible by each input. With the formalism developed in Figure 2, this can be done by giving access, for each input, to only a limited subset of delays in \( \{i \times \Delta, i = 1, \ldots, F\} \) (see Figure 3). Let each input have access to only \( m \) lines: to the line of length 1 and to \((m - 1)\) other lines the length of which is between 2 and \( F \); \( F, m \) and the distribution of the remaining \((m - 1)\) delay lines lengths are thus characteristics of the switching scheme. In case of contention, an excess packet is scheduled for transmission using one of the available delay lines; furthermore, when several delay lines correspond to schedulable transmission times, the shortest delay is chosen. A packet is then lost only if none of the available delay lines correspond to a schedulable transmission time. This may happen even if less than \( F \) packets are already scheduled, since a given input accesses only a subset of the \( F \) scheduling times.

Figure 3: Partial emulation of a FIFO output buffer. \( F = 7, m = 3, n_\lambda = 1 \).

For example, consider Figure 3. Let an incoming excess packet have access to the delay lines of lengths 1, 4 and 6, and assume that transmissions are already scheduled at times \( \Delta, 2\Delta \) and \( 5\Delta \). The packet is then scheduled for transmission at time \( 4\Delta \) (although no packet is scheduled at time \( 3\Delta \)).

This shows in particular that the emulation of the FIFO output queue is not exact, in the sense that packets are not always served in the order of their arrivals. Furthermore, the system is non conservative: a given slot may carry an empty packet although user packets are scheduled for future instants. Lastly, unless supplementary scheduling rules are specified, the scheduling mechanism described above does not maintain packet sequence integrity: a given packet may overtake another packet of the same
connection by taking advantage of an earlier scheduling date, that was not accessible by the previous packet.

3 Modeling the scheduling scheme

The behaviour of discrete time FIFO queues with batch arrivals and services has been thoroughly studied, at least for non bursty arrivals: [3] addresses the case of Binomial arrivals, limited capacity and $n_{\lambda} \geq 1$, [8] addresses the case of Binomial or Poisson arrivals, infinite capacity and $n_{\lambda} = 1$.

This paper is focused on the restricted access case ($m \leq F$), for Binomial arrivals and limited capacity.

Consider a particular output queue of depth $F$. A batch arrival of size $L_{YE}$ occurs at time $i\Delta$ and departures, if any, take place before arrivals, i.e. at time $i\Delta^-$. The queuing system is observed at $i\Delta^+$, i.e. after taking into account possible arrivals and departures.

Let $E_i$ denote the state of the system at time $i\Delta^+$. In the following, we use various descriptions of the state of the system; obviously, one can fully describe the state of the queue, but this leads to a very large state space (see Section 4). Simplifying assumptions shall be made in Section 5 to reduce the state space and provide approximative evaluation results.

Under the assumption of regular, balanced input traffic, a packet may arrive in each time slot, on each of the $N = n_f \ast n_{\lambda}$ inputs of the switch with probability $p$ and a given packet has equal probability $\frac{1}{n_f}$ of being addressed to any of the $n_f$ outputs. $\alpha_i$ is a binomial random variable with parameters $q = \frac{p}{n_f}$ and $N$.

$$P(\alpha_i = j) = f(j) = \binom{N}{j} q^j (1 - q)^{N-j}$$

All the models presented below describe $(E_i)_{i \geq 0}$ as a Markov Process. Furthermore, the present paper focuses on the case $n_{\lambda} = 1$.

Let $P$ denote the transition matrix for $(E_i)_{i \geq 0}$: $P_{it} = \mathbb{P} (E_{i+1} = t \mid E_i = s)$. If $\Pi = (\Pi_0 \Pi_1 \ldots)$ is
the steady-state probability vector for \((E_i)_{i \geq 0}\), the queueing system is simply solved using: \(\Pi P = \Pi \) and \(\sum_{j=0}^{\infty} \Pi_j = 1\). Practically, this is done using either a Gauss or a conjugate gradient squared (CGS) method.

4 Exact Model of the scheduling scheme

For each input, let exactly \(m\) different delays be chosen; delay 1 is part of these delays (if the output is idle, a packet can be immediately transmitted), and \((m-1)\) other delays are chosen, which are uniformly distributed without replacement in \(\{2, \ldots, F\}\).

Furthermore, let:

- \(Y_i = (y^i_1, \ldots, y^i_N)\) characterize arrivals at time \(i\Delta\) \((y^i_k = 1\) if a packet arrives on input \(k\), \(y^i_k = 0\) otherwise).

- \(E_i = (e^i_1, \ldots, e^i_F)\) be the state of the system at time \(i \Delta^+\) \((e^i_k = 1\) if a packet is scheduled for transmission in slot \(i + k\), \(e^i_k = 0\) otherwise).

We assume here that arrivals are taken into account according to ascending indexes \(k\) \((k = 1, \ldots, N)\). The transition probability matrix \(P\) is then derived step by step from the scheduling rules.

Unfortunately, the total number of states is obviously equal to \(2^F\), which leads to an untractable system even for a medium size queue. Approximate models are developed in the following Section.

The exact model allows however to investigate the influence of the choice of the \((m-1)\) delay lines on the performance of the system. This is shown in particular on Figure 4. This Figure represents the probability that a user packet is scheduled for transmission in \(i\) slot times, for different choices of delay lines. Accesses are generated using a random number generation subroutine, for a given distribution (uniform distribution). Figure 4 clearly shows that, for large values of \(i\), the probability that a user packet is scheduled in slot \(i\) clearly varies with \(I_{seed}\), seed of the random number generation subroutine. This has also an impact on the performance, in terms of packet loss, of the scheme, but
this particular point is not further addressed in the present paper.

Figure 4: Probability that a packet is scheduled in \( i \) slot times. \( n_\lambda = 1, n_f = 8, p = 0.8, m = 5, F = 14. \)

5 Approximate models of the scheduling scheme

5.1 General method of resolution

To reduce the number of states in order to obtain a numerically tractable model, we describe less exactly the state \( E_i \) of the system. Instead of describing, at each slot time, the set of scheduled times, we only keep the number of scheduled packets, the largest scheduled time, and in the more precise approximate model, also the smallest scheduled time.

Furthermore, in order to schedule incoming packets, we assume that, for each scheduled packet, the \( m \) available accesses are uniformly distributed, without replacement, independently of the input line. This models the behaviour of a large system for which the \((m - 1)\) accesses are uniformly distributed over \( \{2, \ldots, F\} \).

Let \( X_i^j \) be the state of the system at time \( i\Delta \) after taking account of arrival \( j \) \((j = 0, \ldots, \alpha_i; \alpha_i \leq N) \). Clearly, \( E_i = X_i^{\alpha_i} \).
Transition probabilities are obtained as follows:

\[ P_{st} = \mathbb{P}(E_{i+1} = t \mid E_i = s) \]

\[ = \sum_{l=0}^{N} \mathbb{P}(\alpha_{i+1} = l) \mathbb{P}(X_{i+1}^{\alpha_{i+1}} = t \mid E_i = s) \]

Let \( T_{st} = \mathbb{P}(X_{i+1}^{\alpha_{i+1}} = t \mid E_i = s) \) be the probability that state \( t \) is reached at time \((i + 1)\Delta^t\); let also \( W_{st} \) be the probability that state \( t \) will be reached after taking account of a single arrival, given an initial state \( s \). \( T \) and \( W \) are computed using simple combinatorial analysis. Then:

\[ P = \sum_{j=0}^{N} f(j)TW^j \]

As in many discrete time queues, packet loss probability is not directly yielded by the steady state solution \( \Pi \), but using simple mean value analysis. Let \( A \) be the mean number of arrivals per slot \((A = n_\lambda p = p)\) and \( S \) be the mean number of packets served per slot. The loss probability \( P_{\text{loss}} \) satisfies:

\[ P_{\text{loss}} = \frac{p - S}{p} \]

5.2 Description of the approximate models

Three approximate models, that differ on the exact state description, and on the use of this description, are proposed below; scheduling rules, that are common to all models, are as follows:

- each packet can be scheduled in slot 1 (the server) and in \((m - 1)\) other slots, which are uniformly chosen, without replacement, between 2 and \( F \);
- the packet is scheduled in the smallest available slot, if any;
- if no slot is available, the packet is lost.

Clearly, for \( m = F \), the scheduling scheme corresponds to FIFO queueing.
**Model 1**: Let \( i \) be the number of scheduled packets. We assume that the \( i \) packets are randomly distributed between 2 and \( \text{Max} \), where \( \text{Max} \) is the largest scheduled time. The state of the system is characterized in Model 1 by the total number of scheduled packets, the state of the server (busy or free) and \( \text{Max} \). The total number of state equals \( N_S = F^2 - 2F + 3 \). This model is pessimistic in the sense that the choice, by the scheduling process, of the shortest available delay, is not taken into account when interpreting the state of the system.

**Figure 5: Approximate Model 1**

**Model 2**: We develop an optimistic variant of the previous model. The state of the system is described as in the previous model, but this state is now interpreted as if all scheduled packets were consecutively scheduled, starting from 1. This does take account of the shortest delay scheduling rule, but is obviously optimistic since a FIFO behaviour is assumed although only partial access is provided.

**Figure 6: Approximate Model 2**

**Model 3**: A more precise model has lastly been developed, to refine Model 1, by considering, together with the largest scheduled time (\( \text{Max} \)), the smallest scheduled time (\( \text{Min} \)). As in Model 1, we assume that the \( i \) scheduled packets are randomly distributed between two bounds, \( \text{Min} \) and \( \text{Max} \). Keeping \( \text{Min} \) in the description of the system makes it possible to know whether the server shall be either busy or free due to scheduled packets in the next slot. A state is then characterized by the total number of packets in the queue, the state of the server (busy or free), the smallest and the
largest scheduled time. The total number of state is \( N_S = \frac{1}{3}F^3 - \frac{2}{3}F^2 + \frac{25}{6}F - 2 \) which is significantly larger than in the previous models and which leads to an untractable model for large values of \( F \).

![Diagram](image)

Figure 7: Approximate Model 3

5.3 Validation of the approximate models

In order to check the applicability of the previous approximate models, a scheduler emulating a discrete-time queue with restricted accesses (i.e. \( m \leq F \)), batch arrivals and constant service has been simulated.

The simulation package records, after each scheduled transmission time the exact state of the queue, instead of considering, as in the approximate models, only a partial description of the queue.

In the simulation study, we make the simplifying assumption, as in the approximate models, that the set of accessible scheduling times are randomly chosen (following a uniform distribution) for each packet, in case of contention. This simplifying assumption is believed to be valid for a large switch with regular and balanced input traffic, since two packets of the same connection are, in this case, unlikely to access the same output buffer.

Figure 8 represents the packet loss probability versus \( m \), for the three approximate models and for the simulation.

As expected, the three approximate models behave as FIFO queues when \( m = F \). Considering now the accuracy of our approximations, Figure 8 shows that both models 1 and 3 provide upper bounds for the exact packet loss probability, whereas model 2 provides a lower bound. All bounds are rather tight, and the variation of the packet loss probability with \( m \) is qualitatively identical in the simulation and in the approximate models. This result allows to consider the three approximate models as correct tools for dimensioning the scheduling mechanism at very low packet loss probabilities. that
Figure 8: Comparison of approximate models. $n_\lambda = 1. n_f = 16. p = 0.8. F = 25.$

can obviously not be simulated with appropriate confidence.

As model 3 requires more computation time and memory space than model 1, the latter is preferred in the following.

A last remark can be made on Figure 8: it appears that, for a relatively small value of $m$, the upper bound is tighter than the lower bound, and that the reverse is true for a comparatively larger value of $m$. This is to be expected, since increasing the number of accessible delays allows a better emulation of a FIFO behaviour of the scheduling scheme.

6 Performance analysis of the scheduling scheme

Figure 9 represents the upper bound, obtained with Model 1, of packet loss probability, versus $m$, for several values of $F$. Figure 10 represents the lower bound, obtained with Model 2, of packet loss probability, versus $m$, for several values of $F$.

Both Figures 9 and 10 show that packet loss rapidly decreases for small values of $m$, whereas the curves flatten when $m$ increases. This result is valid for a large range of emulated buffer depths $F$. This means that a large number of accessible delay lines is not necessary to emulate a queueing behaviour similar to a standard FIFO output queue. Furthermore, we also conclude that, in order to
Figure 9: Upper bounds for packet loss probability. $n_\lambda = 1$. $n_f = 16$. $p = 0.8$. Model 1.

Figure 10: Lower bounds for packet loss probability. $n_\lambda = 1$. $n_f = 16$. $p = 0.8$. Model 2.
achieve a given level of packet loss, one may either prefer a large number of accesses, corresponding to rather short delays, or to a smaller number of accesses, corresponding to larger delays.

As an example of this trade-off, consider Figure 9 (which should be preferred for dimensioning purposes since it provides an upper bound for packet loss probability): in order to achieve a loss probability lower than $10^{-9}$, $m = 23$ for $F = 55$, $m = 33$ for $F = 45$ and $m = 42$ for $F = 42$.

Although it appears that a limited number of delay lines ensures a low packet loss probability, the developed models also show that there is a lower bound to the packet loss probability that can be achieved using a small, fixed number of delay lines. Figure 11 represents the packet loss probability versus $F$ for a fixed value of $m$. The curve $m = F$ corresponds to the standard FIFO queue, the behaviour of which is shown in any case to be the best that can be expected from the system with limited access.

![Figure 11: Loss probability vs. queue capacity $F$ for given number $m$ of accesses. $n_\lambda = 1, n_f = 16, p = 0.8$.](image)

Figure 11 shows that, in the relevant range of packet loss levels, two different behaviours are observed: for a very small value of accessible delays ($m = 4$), the achievable packet loss level seems to be limited to a rather high value $10^{-2}$, even for a rather deep buffer ($F = 60$). The packet loss level observed by simulation is represented, together with both upper and lower bounds derived from the approximate models.
Conversely, if the number of accessible delays per input is increased \((m = 16)\), any reasonable packet loss level seems to be achievable, by appropriately increasing the range of accessible delays. More research is however needed to confirm this fact.

7 Conclusion

This paper has presented preliminary evaluation results of an optical transparent packet switch using delay lines to solve contention. Under the assumption of regular and balanced input traffic, our performance results show that it is indeed possible to ensure a very small packet loss probability within the switch, and a high throughput, using a number of delay lines per input that is significantly smaller than the one use in other optical switch architectures. These preliminary results are promising, since they show that one of the technically blocking points for the design of optical packet networks (unavailability of large and fast random access optical memories) can be solved, at least partially.

The proposed switch architecture is shown to achieve a performance, in terms of packet loss and packet delay, comparable to that of a standard switch with output or central buffers.

Several points however have not yet been addressed. The option of maintaining packet sequence integrity within optical packet networks is still being discussed; clearly, optical switches solving contention by means of deflection routing are not able to fulfill this particular option. The scheduling process presented in the present paper is not able either to maintain packet sequence integrity; supplementary scheduling rules that would ensure this function have an impact both on switch complexity and on its performance.

Another aspect of the scheduling process should also be addressed: its performance has been shown in Section 4 to depend on the choice of the delays affecting the accessible delay lines. As shown in preliminary studies not reported here, modifying the distribution of the delays also has an impact on the switch performance. The analytical models described in the present paper need to be revisited in order to take care of this particular aspect.

Research is ongoing to address these technical issues.
References


