Performance of Various Input-buffered and Output-buffered ATM Switch Design Principles under Bursty Traffic: Simulation Study

Soung C. Liew, *Senior Member, IEEE*

***Abstract*— This paper investigates the packet loss probabilities of several alternative input-buffered and output-buffered switch designs with finite amounts of buffer space. The effects of bursty traffic, modeled by geometrically distributed active and idle periods, are explored. Methods for improving switch performance are classified, and their effectiveness for dealing with bursty traffic discussed. This work indicates that bursty traffic can degrade switch performance significantly and that it is difficult to alleviate the performance degradation by merely restricting the offered traffic load. Unless buffers are shared, or very large buffers provided, strategies that improve throughput under uniform random traffic are not very effective under bursty traffic. For input-buffered switches, our investigation suggests that the specific contention resolution scheme we use is a more important performance factor under bursty traffic than it is under uniform random traffic. In addition, many qualitative results true for uniform random traffic are not true for bursty traffic. The work also reveals several interesting, and perhaps unexpected, results: 1) output queueing may have higher loss probabilities than input queueing under bursty traffic; 2) speeding up the switch operation could results in worse performance than having several output ports per output address under bursty traffic; and 3) if buffers are not shared in a fair manner, sharing buffers could make performance worse than not sharing buffers at high traffic loads. Simulation results and intuitive explanations supporting the above observations are presented.**

I. Introduction

A future ATM (Asynchronous Transfer Mode), or fast packet-switching, network has been proposed as an effective way of carrying information of widely varying bandwidth requirements and formats, such as voice, computer file transfers, interactive computer data, and video. Most switch performance analysis to date (e.g., [1] – [6]) has been carried out assuming uniform random traffic, in which the destination outputs of packets are uncorrelated and uniformly distributed across all outputs. The assumption of uncorrelated destinations is not realistic, however, when individual service sessions are allowed to use up a large portion of the line capacity, which may occur with very high-speed data or video services.

In the worst case, the traffic at each input is characterized by bursty packet arrivals, as shown in Fig. 1. Here the packet arrivals consist of bursts to different destinations, and within each burst, packets with a common destination arrive continuously in a stream which instantaneously uses up the whole line capacity. This would be the case, for example, when a long urgent message is to be transported quickly and it is partitioned into many fixed-size packets. This paper investigates the performance implications of this worst-case bursty traffic for various input-buffered and output-buffered switch designs. Specifically, the packet loss probability due to buffer overflow is taken as a performance measure. Although some of the results presented here can be obtained analytically, most are difficult to derive exactly. For consistency, this paper presents and discusses only simulation results.

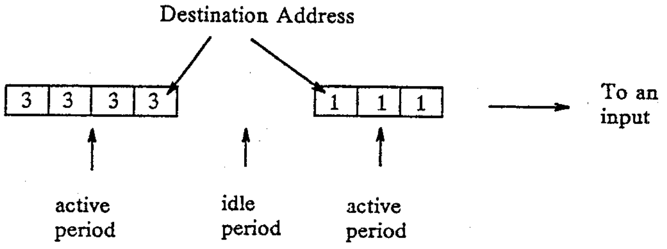

Fig. 1. Packet arrivals to an input under geometric traffic model.

The organization of this paper and the main results are summarized as follows. We first describe and classify several switch schemes of interest in Section II. Their expected relative performance is discussed here on an intuitive basis. Section III details the setup and traffic model used in our simulation experiments. Section IV presents and interprets the simulation results. We show that for both input- and output-queueing schemes, performance degrades significantly when traffic is bursty. Letting separate queues share a common buffer tends to *smooth out* their individual bursty buffer usage, and results in much improved performance for bursty traffic. For input-buffered switches, our investigation suggests that the specific contention resolution scheme we use is a more important performance factor under bursty traffic than it is under uniform random traffic. The simulations also reveal several interesting, and perhaps unexpected, results: 1) output queueing may have higher loss probabilities than input queueing under bursty traffic; 2) speeding up the switch operation could result in worse performance than having several output ports per output address under bursty traffic; and 3) if buffers are not shared in a fair manner, sharing buffers could actually make performance worse than not sharing buffers at high traffic loads. Finally, Section V presents conclusions and

Paper approved by M. S. Goodman, the Editor for Optical Switching of the IEEE Communications Society. Manuscript received June 1, 1991; revised May 4, 1992. This paper was presented in part at IEEE Globecom '90, San Diego, California, December 1990.

The author is with Department of Information Engineering, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong. This work was done while the author was at Bellcore, New Jersey, U.S.A.

IEEE Log Number 9401035.

suggests several research issues that deserve further attention.

II. Alternatives for Improving Switch Performance

This section considers various alternatives for improving switch performance. We assume ATM transport in which data streams are partitioned and transferred in packets of fixed size. On a conceptual level, time is therefore divided into slots corresponding to the packet transmission time. Synchronous switch operation is assumed; that is, packets arrive synchronously at the inputs at the beginning of each slot, and all packets gaining access to their output lines are cleared by the end of each time slot.

A. Input Queueing

With input queueing, an arriving packet enters a FIFO input buffer and waits for its turn to access its addressed output. When multiple packets from different input buffers try to access a common output port, arbitration or contention resolution is necessary in order to determine the winning packet. With ordinary input buffering, only the head-of-line (*HOL*) packets of the FIFOs are involved in the arbitration.

It is well-known that, under uniform random traffic, the maximum throughput of an input-buffered switch is limited to 0.586 [2]. Under uniform but geometrically distributed bursty traffic, the maximum throughput could further degrade to 0.5 [7], [8]. The packet loss probabilities for offered loads close to the throughput limitation are high regardless of buffer size. Therefore, one way to improve loss probability is to increase the maximum throughput of the switch by modifying the original input-buffered switch.

In the following, we discuss various switch design strategies and discuss their impact on switch performance. These strategies represent different "degrees of freedom" for switch design. Although they are studied separately here in order to identify their individual effects clearly, they can be combined in a single switch architecture in practice. The so-called completely shared buffering scheme and input-smoothing scheme proposed in Reference [2] actually represent switch designs which combine several concepts presented here. The input-smoothing scheme incorporates the concepts of input-port expansion and output-port expansion. The completely shared buffering scheme ties the two degrees of freedom represented by input-port expansion and buffer sharing into one; increasing the buffer size will necessitate increasing the input-expansion factor by the same amount.

To help explain the strategies, Table 1 lists the packets that will be cleared, assuming the initial queueing state depicted in Fig. 2. In the figure, packets a and e are queued at input 1, packets b and f at input 2, packets c, g, and i at input 3, and packets d and h at input 4. Numbers beside the packet labels represent destination output addresses. For simplicity, we assume in Table 1 that no new packets arrive in the next two time slots, and that packets at the upper input ports will be favored over those at the lower input ports under contention.

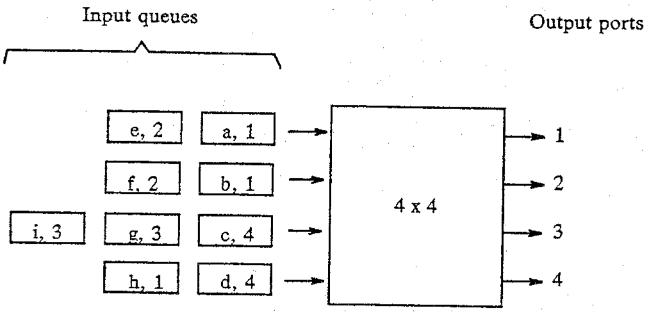

Fig. 2. Illustrating example: packets queueing at Inputs of a 4 × 4 Switch.

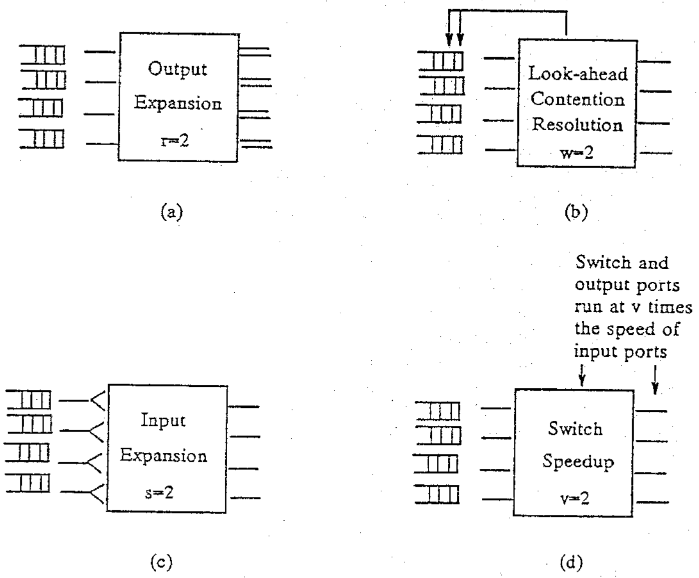
Output Expansion r=2
(a)
Look-ahead Contention Resolution w=2
(b)
Input Expansion s=2
(c)
Switch and output ports run at v times the speed of input ports
Switch Speedup v=2
(d)

Fig. 3. Strategies for improving performance of input-buffered switches.

Output-port Expansion

If there are more output ports than input ports, then the offered load per output port (and therefore contention among inputs for available outputs) is reduced. Figure 3(a) shows a particular output-port expansion scheme based on the channel-grouping concept [5], [9]. With r output ports provided for each output address, up to r packets can access any output address simultaneously. However, no more than one packet can be cleared from any single input in a given time slot. Here, it does not matter which output port a packet accesses, as long as the output port belongs to the targeted output address.

The switch shown in the figure is n inputs × nr outputs. To construct a symmetric $n \times n$ switch, each group of r output ports must be recombined (concentrated) back

Table 1. Sequence of cleared packets based on queueing state shown in Fig. 2.

Switch scheme	*Packets cleared in 1st time slot*	*Packets cleared in 2nd time slot*	*Packets remaininng at end of 2nd time slot*
Ordinary input queueing	a, c	e, b, g, d	f, i, h
Output-port expansion, $r = 2$	a, b, c, d	e, f, g, h	i
Look-ahead contention $w = 2$	a, f, c	e, b, g, d	i, h
Input-port expansion $s = 2$	a, e, c, g	b, f, i, d	h
Switch speed-up $v = 2$	a, c, e, b, g, d	f, i, h	
Input buffer sharing	same as ordinary input queueing for this example		

to one port. As a result, buffers are needed at the outputs because of potential simultaneous packet arrivals. Depending on the actual application, the recombination of packets may not be necessary; e.g., the switch shown could be a switch module within an overall multistage switch architecture [5] in which the internal bandwidth is greater than the external bandwidth. This report considers the input buffers for the channel-grouping scheme, but does not consider a combination of input buffers and output buffers.

Switch Speed-up

In a switch that operates at v times the input line speed, the effective offered load to the switch is reduced by a factor of v. Given $v = r$, the switch speed-up strategy should be better than channel-grouping, since speed-up allows more than one packet from a given input port to be cleared in the same time slot. Note that the speed-up scheme described in Reference [10] operates in an analogous fashion to the channel-grouping approach described above. We think there is no reason in the speed-up scheme to intentionally limit the number of clearable packets from each input port to one.

If the output line speed is the same as the input line speed, a FIFO buffer would be needed at each output port because of potential simultaneous packet arrivals. As in the channel-grouping scheme, the buffer is not needed if the output line speed is scaled up accordingly (which would be the case if this switch were one of many switch modules in a multistage switch [5]). This paper will not consider buffering at the outputs for the speed-up scheme.

Look-ahead Contention Resolution Scheme

In look-ahead contention resolution [2], the contention-resolution process consists of w cycles of contention resolutions. In the first cycle, only the HOL packets are allowed to contend for the outputs. At the end of this cycle, there may still be some unclaimed outputs simply because no HOL packets are destined for them. In the second cycle, the second packets of the input queues that have lost contention in the previous cycle contend for the remaining outputs. This process is repeated w times, and the winning packets at the end of w cycles then access their respective outputs in the same time slot. Note that only one packet can be cleared from any single input. In general, the maximum allowable throughput increases with w [2]. The price is that the arbitration must be carried out at w times the original speed. If the arbitration for the next time slot is carried out while the winning packets of the current time slot are being transmitted, then arbitration-time overhead can be avoided if $w \leq T_p/T_a$, where T_p is the packet transmission time and T_a is the time needed for one cycle of contention resolution.

To illustrate the look-ahead strategy, consider the queueing state in Fig. 2, and assume $w = 2$. In the first time slot, packets a and c are selected in the first cycle, and packet f is selected in the second cycle. Packets e and g are not involved in the second-round contention because packets a and c ahead of them have already been selected. Packet h will not be selected because the destination output address has already been assigned in the preceding cycle. Thus, packets a, c and f are cleared in the first time slot. In the second time slot, packets e, b, g, and d are cleared.

Input-port Expansion

Figure 3(c) depicts a particular input-port expansion scheme in which each input port is expanded into s ports before packets enter an asymmetric $ns \times n$ switch. Up to s packets from each input queue can be presented to the inputs for contention. With $s = w$, one would expect this switch to have a higher maximum throughput than the look-ahead scheme, because it is possible for more than one packet to be cleared from each input queue in the same time slot. With the queueing state in Fig. 2, note that packet i can also be cleared in addition to all packets that can be cleared by the look-ahead scheme.

B. Output Queueing

With output queueing, arriving packets destined for any output immediately enter an output FIFO buffer and wait for their turns to access the output line. Thus, while an input buffer generally contains packets destined for different outputs, an output buffer contains only packets destined for the corresponding output. Conceptually, we can define output queueing as performing switching (or destination sorting) prior to buffering. For output queueing, given an offered load smaller than 1 and some definite loss probability requirement, one can always find a buffer size b that lets the switch meet the loss probability. Since the required b may be very large (say, more than a hundred packets long), however, it is also desirable to find ways to reduce b.

Output-bandwidth Expansion

As in the input-buffered switch, we can also expand the number of ports per output address so that packets buffered at the outputs can be cleared quickly. Here, we assume there is a common queue for the output ports of the same output address. The same performance can also be obtained by speeding up the output ports rather than expanding them. In either case, the total bandwidth on the output side would be larger than the total bandwidth on the input side. This results in inefficient trunk usage if this switch is a stand-alone system, but would not be unreasonable if it is one of many switch modules in a multistage switch architecture [5].

Output-buffer Sharing

Under bursty traffic, it is likely that some buffers are relatively empty while others are full. If the buffers can somehow be shared, then the packet loss probability can be reduced. The sequence of cleared packets when the buffers do not overflow is the same as in the separate-buffer scheme. In the generic buffer-sharing scheme, however, no packets will be lost as long as the total buffer occupancy at the outputs is no more than nb, where b is the buffer size per output. Reference [11] describes a switch design with this property.

III. Traffic Model and Simulation Setup

To model switch performance quantitatively, we adopt the uniform geometrically bursty traffic model in which an input alternates between active and idle periods of geometrically distributed duration [12]. Packets destined for the same output arrive continuously in consecutive time slots during an active period. The duration of the active period is characterized by a parameter p. The event that the active period will terminate after a time slot is a random process which occurs with probability p. The probability that the active period (burst) lasts for a duration of i time slots (consists of i packets) is then

$$P(i) = p(1-p)^{i-1}, \qquad i \geq 1. \tag{1}$$

Note that we assume there is at least one packet in the burst. The mean burst length is given by

$$E_B[i] = \sum_{i=1}^{\infty} iP(i) = 1/p. \tag{2}$$

The idle period is geometrically distributed with parameter q. The probability that an idle period lasts for j time slots is

$$Q(j) = q(1-q)^j, \qquad j \geq 0. \tag{3}$$

Unlike the duration of an active period, the duration of an idle period can be 0. The mean idle period is given by

$$E_I[j] = \sum_{i=0}^{\infty} jQ(j) = (1-q)/q. \tag{4}$$

Given p and q, the offered load ρ can be found by

$$\rho = E_B[i]/(E_I[i] + E_B[j]). \tag{5}$$

We assume there is no correlation between different bursts, and the destination of each burst is uniformly distributed among the outputs. Note that the uniform random traffic model discussed in References [2] and [3] is a special case with $p = 1$ and $q = \rho$.

Our simulation experiments measure the packet-loss probability of a 128×128 switch. Switches of dimensions larger than 32×32 should have approximately the same results. Simulated packets that arrive at fully occupied queues are discarded and considered lost. The statistics of about 10 million packets are collected (over all queues in the switch) for each data point. Thus, loss probabilities smaller than 10^{-7} are not measurable, and loss probabilities below 10^{-5} could contain non-negligible errors. If needed, however, loss probabilities below 10^{-5} can be extrapolated based on our results. Average burst lengths of 1 (the uniform random traffic) and 8 are considered.

For the purpose of contention resolution in the input-buffered schemes, the simulation programs assume that priorities of the input ports are ordered in a cyclic fashion. In every round of contention, a random port, say $Port_{top}$, is chosen to be the port with the top priority. The priorities of the ports are then ordered as $Port_{top}$, $Port_{top+1 \pmod{128}}$, $\ldots$, $Port_{top+127 \pmod{128}}$. Note that this is *not* the fixed-priority assumed in Table 1.

IV. Simulation Results and Interpretations

A. Input Queueing

Effects of Bursty Traffic on Ordinary Input-buffered Scheme

Figure 4 shows graphs of packet-loss probability P_{loss} versus the offered load ρ for buffer size b of 16, 32, and 64 packets per port. As can be seen, P_{loss} degrades significantly when the burst length l is increased from 1 to 8. For the sake of argument, let's define the acceptable load ρ_a to be the offered load at which $P_{loss} = 10^{-6}$. The graphs clearly show that burstiness in traffic results in low ρ_a.

Although we would usually not operate a switch at high offered loads with correspondingly high P_{loss}, it is interesting to study the switch performance at high loads because much insight about the switching mechanisms can be gained this way. Under both uniform random and bursty traffic conditions, regardless of b, the curves converge until they overlap as the offered load increases beyond the maximum allowable throughput ρ_{max}. This is typical of input queueing, since queues start to saturate in this region. In fact, ρ_{max} can be approximated by $1 - P_{loss}(\rho = 1)$.

Overall, bursty traffic has two effects: 1) ρ_{max} is reduced; and 2) the slope of the P_{loss} vs. ρ curve for $\rho < \rho_{max}$ becomes less steep so that decreasing ρ is not as effective in lowering P_{loss} as it is for non-bursty traffic. Although the degradation of ρ_{max} is small, the degradation of ρ_a is

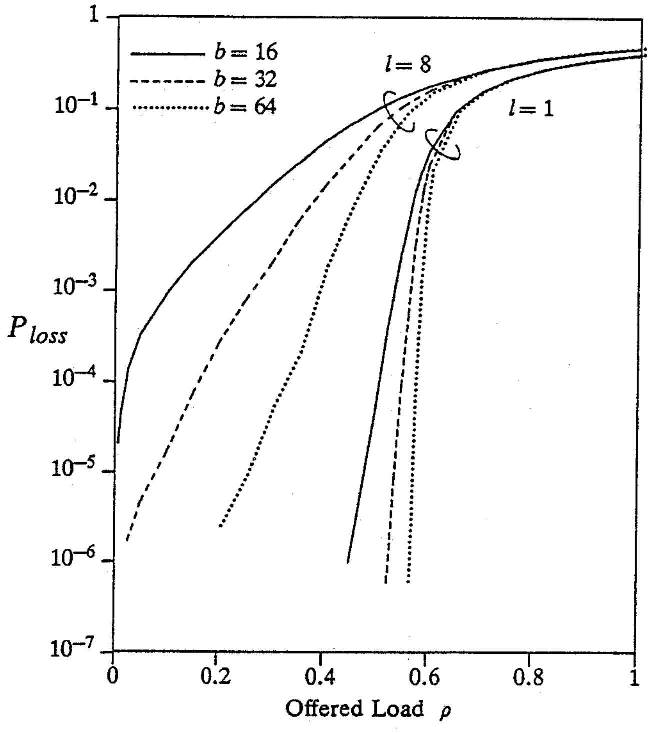

Fig. 4. P_{loss} vs. ρ for ordinary input-buffered scheme.

large because of 2). As a result, ρ_{max} is not a good metric for performance under bursty traffic, since we would most likely operate at loads much below that to limit P_{loss}.

Input-Buffered Scheme under Uniform Random Traffic

Figure 5 shows that P_{loss} and ρ_{max} under uniform random traffic improve as we go from the ordinary input-buffered scheme (o) to the look-ahead scheme (with $w = 2$), to the input-port expansion scheme (with $s = 2$), and then to the output-port expansion scheme (with $r = 2$). The results for the speed-up scheme (with $v = 2$) are not shown because the corresponding loss probabilities are too small to be measured with our simulation experiments; e.g., $\rho_{max} > 1$, and $P_{loss}(\rho = 1)$ is 3×10^{-6} for $b = 16$, and 0 for $b = 32$. For all cases studied, $b = 64$ is sufficient to achieve ρ_a that is close to ρ_{max}.

Input-Buffered Scheme under Bursty Traffic

Figure 6 shows that the performance of the ordinary input-buffered scheme, the look-ahead scheme (with $w = 2$), and the input-port expansion scheme (with $s = 2$) are practically the same when $l = 8$. This is obvious, since bursty traffic conditions make it likely that the next packet has the same output destination as the HOL packet in each queue. Thus, letting the next packet from each queue compete for the outputs does not help much. The speed-up (with $v = 2$) scheme and output-port expansion (with $r = 2$) schemes use a different mechanism to achieve improvements: packets are simply cleared at the outputs at a higher rate. Thus, switch performance is improved for these schemes.

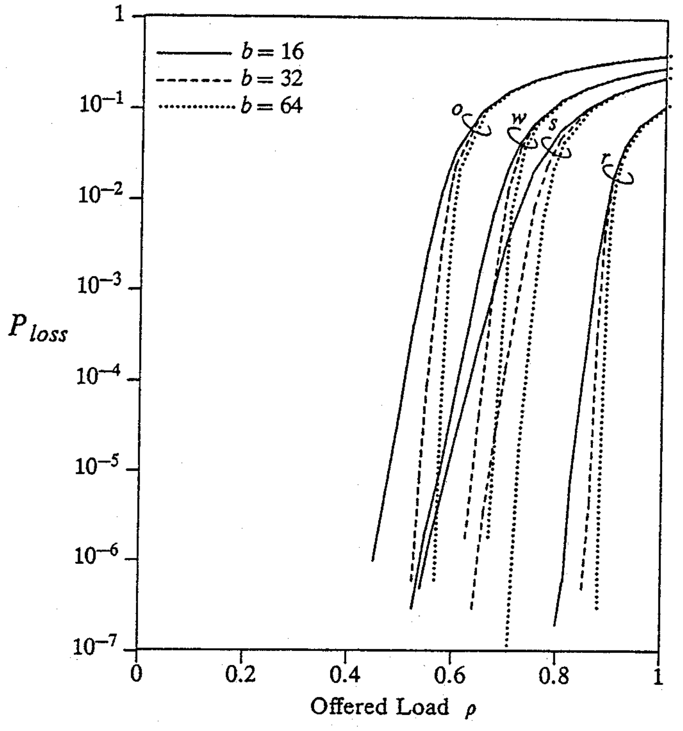

Fig. 5. P_{loss} vs. ρ for ordinary input-buffered scheme (o), look-ahead scheme ($w = 2$), input-port expansion scheme ($s = 2$), and output-port expansion scheme ($r = 2$), under uniform random traffic ($l = 1$).

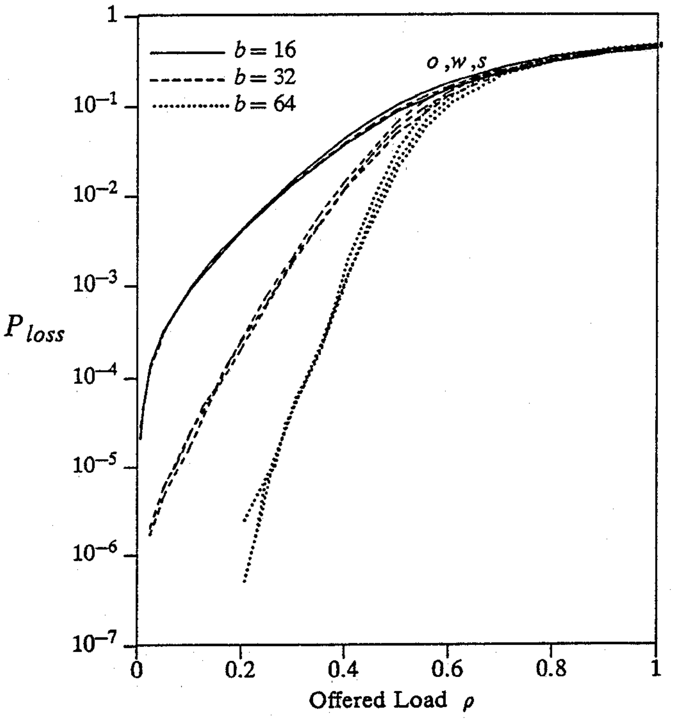

Fig. 6. P_{loss} vs. ρ for ordinary input-buffered scheme (o), look-ahead scheme ($w = 2$), and input-port expansion scheme ($s = 2$), under bursty traffic ($l = 8$).

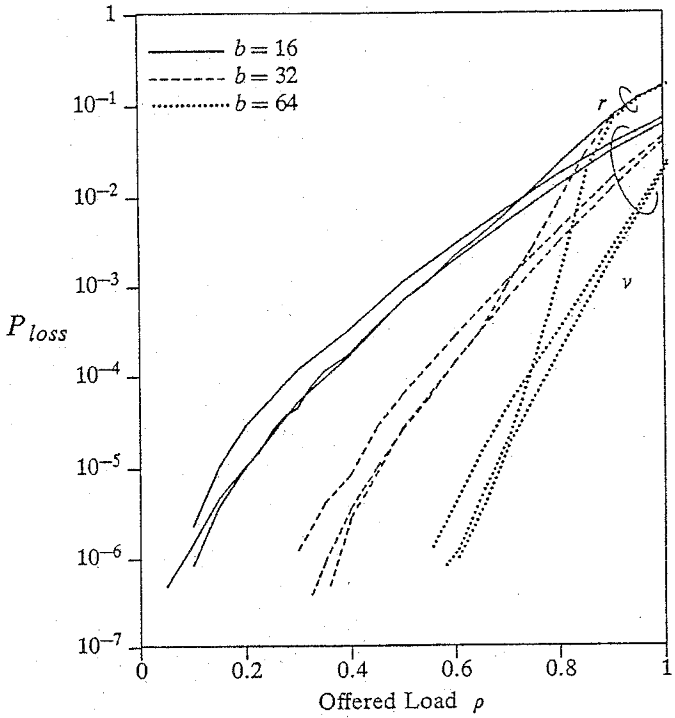

Fig. 7. P_{loss} vs. ρ for output-port expansion scheme ($r = 2$) and two alternative speed-up schemes ($v = 2$) under bursty traffic ($l = 8$).

Two variations for contention resolution in the speed-up scheme have been investigated. With $v = 2$ there are two rounds of contention in each time slot. For the first variation, called the single-sweep method, a new input port is randomly selected in each round as $Port_{top}$ (see Section III). For the second variation, called the double-sweep method, a new input port is randomly selected for the first round in each time slot, but for the second round, the priorities of the ports are reversed and ordered as $Port_{top+127 \pmod{128}}, Port_{top+126 \pmod{128}}, \cdots, Port_{top}$. The curves with the lower P_{loss} in Fig. 7 belong to the second variation.

The double-sweep method was simulated after a peculiar behavior of the single-sweep method was observed. Recall that we expected the speed-up scheme to perform better than the channel-grouping scheme. This is indeed the case under uniform random traffic. But, as shown in the figure, the channel-grouping scheme actually outperforms the single-sweep method for regions of small P_{loss}. To explain this, we hypothesize that under bursty traffic there may often be either no HOL packet or more than one HOL packet targeted for any given output.

If we consider a tagged output address, the speed-up scheme will have an advantage over the channel-grouping scheme only when there is one and only one HOL packet destined for the output. In this case the channel-grouping scheme can only clear one packet at the tagged output, whereas the speed-up scheme may be able to clear an additional packet when new packets move to the heads of input queues in the second round of the same time slot. The hypothesis basically says that this advantage does not exist most of the time. To see why speed-up could actually be worse, we need to look into the contention-resolution process. With more than one HOL packet destined for a given output address, the channel-grouping scheme will always select two packets from different inputs for clearance at the output. But the single-sweep method may clear two adjacent packets from the same input instead; bursty traffic increases this likelihood since two adjacent packets are likely to be destined for the same output. The unfair service discipline of the single-sweep method may cause uneven distribution of queue lengths, resulting in higher P_{loss} than the channel-grouping scheme.

The double-sweep method guarantees that when there is more than one HOL packet for a given output addresses, the cleared packets will be taken from different inputs. Figure 7 shows that the double-sweep method has about the same performance as the channel-grouping scheme, further validating our hypothesis.

Finally, unlike the channel grouping scheme, neither speed-up scheme has maximum throughput limitations, as can be seen from the fact that the curves for different b do not converge at $\rho = 1$. In summary, we observe the following for the simulations performed:

1. the look-ahead and input-port expansion schemes are not effective under bursty traffic;
2. the advantage of the speed-up scheme over the channel-grouping scheme is negligible under bursty traffic;
3. under bursty traffic, the contention probability is high, and contention-resolution strategies that guarantee fairness are more important than under the uniform random traffic; and
4. for all schemes, burstiness degrades ρ_a significantly, and a buffer size of 64 packets is not sufficient to make ρ_a close to ρ_{max}.

B. Output Queueing

Effects of Bursty Traffic on Ordinary Output-buffered Scheme

As shown in Fig. 8, bursty traffic degrades the performance of the output-buffered switch even more than that of the input-buffered switch. For a buffer size of 64 packets per port, ρ_a when $l = 8$ is less than 0.1! Although there is no throughput limitation, and P_{loss} can be made arbitrarily small with sufficiently large b, it appears that the required b will be prohibitively large in order for ρ_a to be, say, 0.8. A case not shown in Fig. 8 is when b is increased to 128. Our simulation also reveals that simply increasing the buffering space by eight times is not enough to compensate for the increase in traffic burstiness by eight times. For instance, fixing P_{loss} at 10^{-6}, the allowable load for $l = 1, b = 16$ is 0.65, but the allowable load for $l = 8, b = 128$ is only 0.35.

Output-bandwidth Expansion

Figure 9 shows the improvement in performance under bursty traffic ($l = 8$) when the output bandwidth is dou-

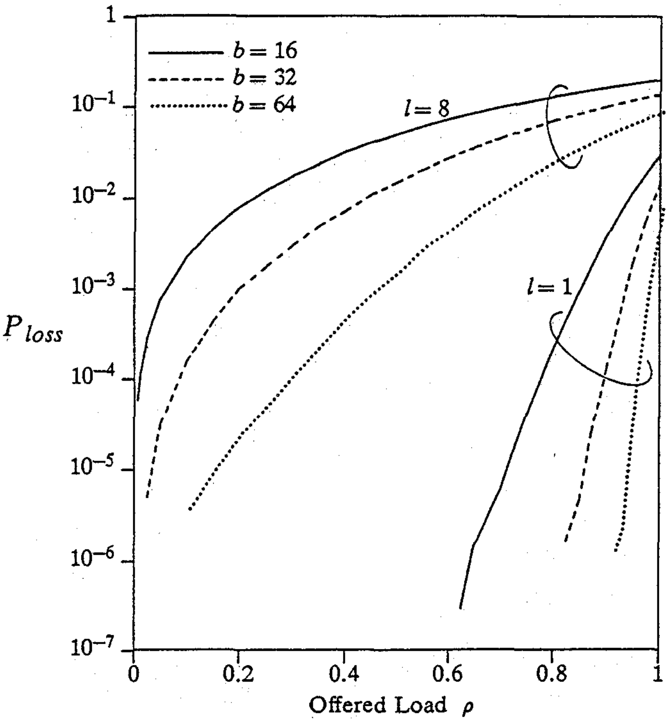

Fig. 8. P_{loss} vs. ρ for ordinary output-buffered scheme.

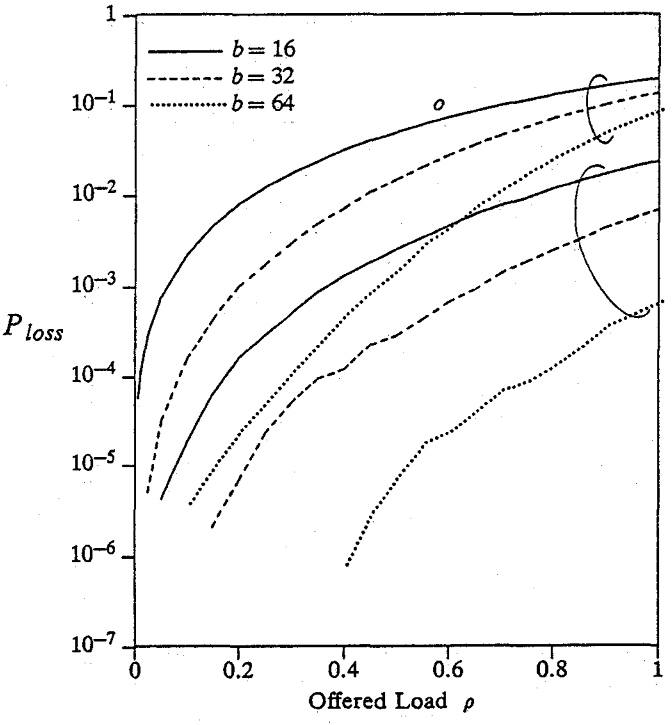

Fig. 9. P_{loss} vs. ρ for ordinary output-buffered scheme (o) and output-bandwidth expansion scheme (r = 2) under bursty traffic.

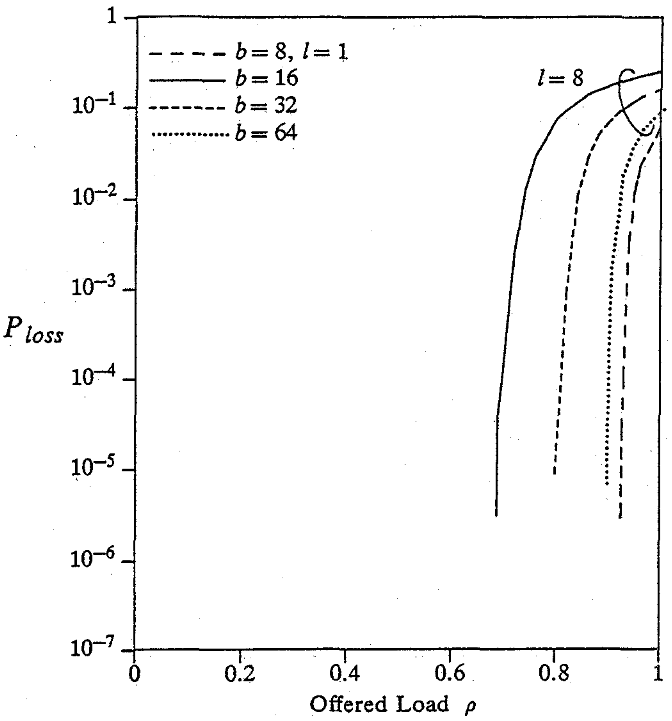

Fig. 10. P_{loss} vs. ρ for output-buffer sharing scheme.

bled, either by doubling the number of output ports per output address or by doubling the output port speed. For $b \leq 64$, although there is some improvement in ρ_a, it is still below 0.4.

Output-buffer Sharing

Figure 10 shows that for buffer sharing, the P_{loss} vs. ρ curves are practically vertical when ρ is below certain values (which depends on the buffer size). Consequently, very low loss probabilities can be achieved simply by controlling the offered load with network-level controls such as routing and congestion control. However, it is possible for a few queues in our simulations to take over the entire buffer space under high-load situations, resulting in worse performance than having separate buffers for separate queues. For instance, the simulation results show that for $b = 16$, $l = 8$, and $\rho = 1$, P_{loss} is 0.260 with buffer sharing and 0.202 without buffer sharing. This points out that a fair buffer-allocation strategy is needed to 1) ensure uniform loss probabilities among the output ports, and 2) obtain a lower overall average loss probability.

C. Output Queueing vs. Input Queueing

Without buffer sharing, when the traffic is bursty, our simulation results indicate that the loss probability of input queueing can be lower than that of output queueing for $\rho < \rho_{max}$. Figure 11 compares the ordinary input-buffered scheme with the ordinary output-buffered scheme under bursty traffic (l = 8). For each of the buffer size, there is an offered load ρ' for which input queueing is better than output queueing for all $\rho < \rho'$. This is so even if b is increased to 128 (not shown in Fig, 11), although the

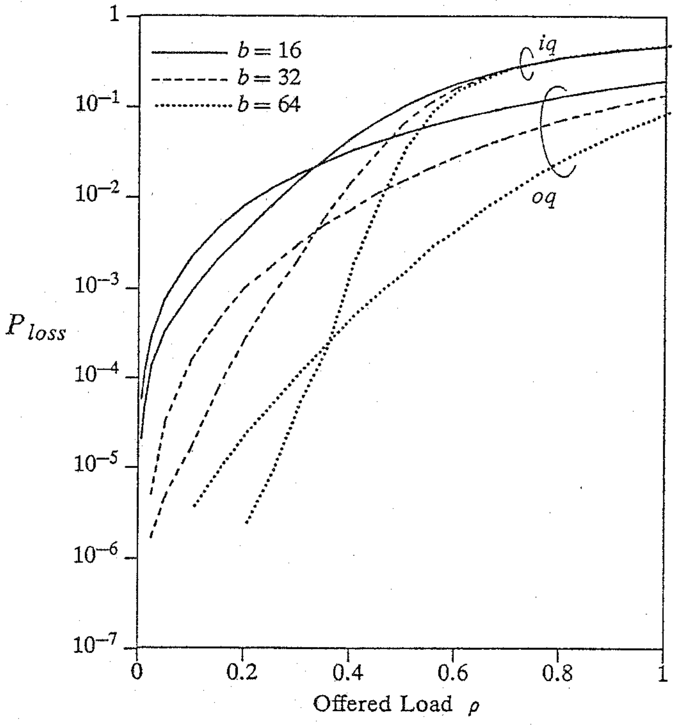

Fig. 11. P_{loss} vs. ρ for ordinary input-buffered scheme (iq) and ordinary output-buffered scheme (oq) under bursty traffic ($l = 8$).

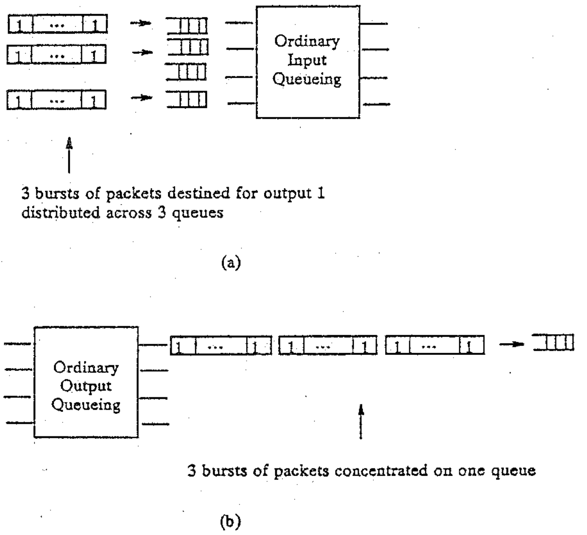
1 ··· 1
Ordinary Input Queueing
3 bursts of packets destined for output 1 distributed across 3 queues
(a)
Ordinary Output Queueing
1 ··· 1
3 bursts of packets concentrated on one queue
(b)

Fig. 12. Output queueing vs. input queueing when multiple bursts of packets arrive for the same output ports.

crossover point is now at a higher ρ of 0.35 and a lower P_{loss} of 10^{-6}. The same qualitative results as above is also observed if we compare the curves for $r = 2$.

The intuitive explanation for the above is that there are some inherent buffer-sharing effects in the input-buffered scheme not found in the output-buffered scheme. As illustrated in Fig. 12, when multiple bursts arrive for the same output, packets from all these bursts go to the same output queue in the output-buffered scheme, and the queue can overflow easily. With input queueing, however, the packets from these bursts are distributed across several input queues, and therefore the queues do not overflow as easily. In short, one cannot claim that output queueing is better than input queueing under all traffic situations.

V. Conclusions

We have investigated the effectiveness of various input-buffered and output-buffered switches for dealing with bursty traffic. The results of this study and the insights derived are summarized as follows:

- Without buffer sharing, bursty traffic decreases the slope of the P_{loss} vs. ρ curve so that simply decreasing the offered load is not effective in lowering P_{loss}. This implies simply controlling the offered load at the network level with a simplistic admission-control strategy will not be very useful. With input queueing, burstiness reduces the maximum throughput ρ_{max} only slightly. However, ρ_{max} is not a good performance metric under bursty traffic, since it is necessary to operate the switch at loads much below ρ_{max} in order to obtain small P_{loss}.
- For input queueing with uniform random traffic, switch designs that improve ρ_{max} generally also decrease P_{loss}. Under bursty traffic, the look-ahead and input-port expansion schemes have negligible effects on P_{loss} and ρ_{max}.
- For input queueing, speeding up the switch operation results in lower P_{loss} than expanding the number of output ports per output address under uniform random traffic. However, depending on the contention-resolution scheme assumed, the reverse could actually happen under bursty traffic. Our work suggests that the specific contention-resolution scheme we use is a more important performance factor under bursty traffic than it is under the uniform random traffic.
- Buffer-sharing for output queueing is a very effective way for dealing with bursty traffic. Specifically, the slope of the P_{loss} vs. ρ curve becomes practically vertical below a certain offered-load threshold. As a result, very small P_{loss} can be achieved by decreasing the offered load slightly below the threshold. With a steep P_{loss} vs. ρ curve, however, tight control is needed in order to keep the offered load below the threshold, or else the performance could degrade very quickly with small increases in the offered load. Therefore, buffer sharing, if adopted, must be taken into account in higher level network protocols that permit congestion control.

Sharing buffers can result in higher P_{loss} than not sharing buffer if the loading sequence of input packets into the shared buffers is static. This typically happens at high offered loads. Regardless of the offered load, it is important to make sure that some queues are not always favored over others when buffers are shared, not only to guarantee fairness, but also to obtain a lower overall P_{loss}.

Under bursty traffic and without buffer sharing, output queueing could have higher P_{loss} than input queueing. Input queueing is relatively more robust with respect to bursty traffic because of the certain degree of inherent buffer-sharing effects: although buffers are actually not shared, simultaneous packet arrivals with a common destination are automatically distributed across several buffers in input queueing. The loss probability can be further improved if the arbitration process gives priority to the longest queues. This could be achieved easily with contention resolution based on a sorting network [3]. A few extra bits corresponding to the queue length are simply attached to the destination address of the packet header. The robustness of this strategy for bursty traffic remains an issue to be studied.

In conclusion, this investigation indicates that many qualitative results that are true for random uniform traffic are not necessarily true for bursty traffic. With bursty traffic, unless buffers are shared or very large buffers provided, it is difficult to lower the loss probability by methods that are designed to increase throughput for uniform random traffic.

Acknowledgements

Suggestions and Comments by Tony Lee, Howard Lemberg, and Kevin Lu are much appreciated.

References

[1] M. G. Karol, M. J. Hluchyj, and S. Morgan, "Input versus Output Queueing on a Space-Division Packet Switch," *IEEE Trans. on Commun.*, vol. 35, Dec. 1987, pp. 1347–1356.

[2] M. G. Hluchyj and M. J. Karol, "Queueing in High-Performance Packet Switching," *IEEE J. on Selected Areas in Commun.*, vol. 6, no. 9, , Dec. 1988, pp. 1587–97.

[3] J. Y. Hui and E. Arthurs, "A Broadband Packet Switch for Integrated Transport," *IEEE J. on Selected Areas in Commun.*, Oct. 1987, pp. 1264 – 1273.

[4] Y. Yeh, M. J. Hluchyj, and A. S. Acampora, "The Knockout Switch: A Simple Modular Architecture for High-Performance Packet Switching," *IEEE J. on Selected Areas in Commun.*, vol. 5, Oct. 1987, pp. 1274–1283.

[5] S. C. Liew and K. W. Lu, "A Three-Stage Architecture for Very Large Packet Switches," *International J. of Digitanl and Analog Communications Systems,* vol. 2, 1989, pp. 303–316.

[6] A. E. Eckberg and T. C. Hou, "Effects of Output Buffer Sharing on Buffer Requirements in an ATDM Packet Switch," *Proc. of INFOCOM '88*, March 1988.

[7] S-Q Li, "Performance of a Non-Blocking Space-Division Packet Switch with Correlated Input Traffic," *Conf. Record, Globecom '89*, vol. 3, pp. 1754–1763.

[8] S. C. Liew and K. W. Lu, "Comparison of Buffering Strategies for Asymmetric Packet Switch Modules," *IEEE. J. on Selected Areas in Commun.*, vol. 9, April 1991.

[9] A. Pattavina, "Multichannel Bandwidth Allocation in a Broadband Packet Switch," *IEEE J. on Selected Areas in Commun.*, vol. 6, no. 9, Dec. 1988, pp. 1489-1499.

[10] Y. Oie *et al.*, "Effect of Speedup in Nonblocking Packet Switch," *Conf. Record, ICC '89*, vol. 1, pp. 410–415.

[11] H. Kuwahara *et al.*, "A shared buffer memory switch for an ATM exchange," *Conf. Record, ICC '89*, vol. 1, pp. 118–122.

[12] A. Descloux, "Contention Probabilities in Packet Switching Networks with Strung Input Processes," *Proc. of the ITC 12*, 1988.

Soung C. Liew received the S.B., S.M., E.E., and Ph.D. degrees in electrical engineering from Massachusetts Institute of Technology, Cambridge, in 1984, 1986, 1986, 1988, respectively. From 1984 to 1988, he was a Research Assistant in the Local Communication Networks Group at the M.I.T. Laboratory for Information and Decision Systems, where he investigated fundamental design problems in high-capacity fiber-optic networks. He was also a Teaching Assistant for a graduate course on data communication networks.

In March 1988, he joined Bellcore, Morristown, New Jersey, where he has been a Member of Technical Staff in the Network Systems Research Laboratory. He is currently taking a leave of absence from Bellcore and is Senior Lecturer in the Chinese University of Hong Kong. He has conducted research and published actively in various areas related to broadband communications, including wavelength-division-multiplexed optical networks, high-speed packet-switch designs, system-performance analysis, routing algorithms, network-traffic control, and reliable and survivable networks.

His current research interests include interconnection networks, broadband network control and management, distributed and parallel computing, fault-tolerant networks, and optical networks. Dr. Liew is a senior member of the IEEE and a member of Sigma Xi and Tau Beta Pi.