Sizing a Packet Reassembly Buffer at a Host Computer in an ATM Network

Donald E. Smith and H. Jonathan Chao, Senior Member, IEEE

Abstract—This paper develops a queuing model of a buffer that collects cells for reassembly into packets for a protocol layer above the asynchronous transfer mode (ATM) layer. Whenever the buffer fills with all packets incomplete, a packet must be sacrificed to make room for others. The queuing model estimates the equilibrium fraction of packets sacrificed under one algorithm for selecting the packet to be sacrificed. The paper also uses simulation to compare three sacrifice algorithms. The model’s predicted packet loss probabilities bound from above the loss probabilities in the simulations of the different algorithms. Applications to sizing the buffer for a prescribed loss probability are given.

Index Terms—Asynchronous transfer mode, reassembly buffer, processor sharing M/G/1 queue.

I. INTRODUCTION

ONE APPLICATION envisioned for the public ATM network [1] is the interconnection of high-performance host computers or local area networks. Hosts typically engage in file transfers among one another and format their data into packets that may be much larger than ATM cells (e.g., up to 2^14 - 1 or 65,535 bytes of information payload in a packet versus 44 bytes in an ATM cell). To convert data packets into ATM cells and vice-versa, each host has an interface to the ATM network, called a broadband terminal adaptor (BTA)[2]. The BTA collects ATM cells it receives from the ATM network in a buffer called the virtual channel queue (VCQ) for reassembly into packets. The cells from all the virtual circuits active at any given time will enter the VCQ interleaved in time. This means the VCQ must be large enough to accommodate partially completed packets from many virtual circuits. Cell interleaving also dictates whenever the VCQ fills with incomplete packets, the BTA “sacrifices” one incomplete packet to make room for the cells of other packets. That is, the BTA reads out the cells of the sacrificed packet already in the VCQ and the remaining cells of the sacrificed packet as they come. The purpose of the sacrifice is to limit the loss to one packet each time the VCQ fills with incomplete packets.

Reference [2] evaluates two candidate architectures for the VCQ. This paper studies one of those architectures, a VCQ structure in which packets fully share buffer space. The objective is to calculate the equilibrium fraction of packets sacrificed as a function of the algorithm that selects the packet to be sacrificed, the VCQ size, and a packet traffic characterization. One can determine the minimum VCQ size required to attain a prescribed packet loss objective.

The sacrifice algorithm we considered initially discarded the packet with the most cells in the VCQ. At first sight, this algorithm would seem to keep packet losses low because each sacrifice frees up as much space as possible. The algorithm is also very similar to one that is analytically tractable, namely the algorithm that sacrifices the packet with the least number of cells remaining to be transmitted by the ATM switch. When all packets are of equal length, the two algorithms coincide. We analyze sacrifice by the fewest remaining cells, using as a tool, the completed work process. This counts the number of cells serving the host in the VCQ, belonging to unsacrificed packets currently in progress at the ATM switch. We compare completed work sample paths for finite and infinite VCQ’s to derive an upper bound for the fraction of packets sacrificed in terms of the infinite VCQ completed work process.

When we approximate the packet service discipline at the output link from the ATM switch to the host by the processor sharing discipline and packet arrivals by a Poisson process, the bound on the loss probability becomes a tail of the sojourn time distribution in the M/G/1 queue. We compute the bound using an asymptotic expression for the tail probability from renewal theory [3]. We then compare the predicted loss probabilities to the losses in simulations.

Sacrificing the packet with the fewest number of untransmitted cells cannot be implemented in practice because the length of the packet is not indicated in any of the partitioned cells of the packet (instead the last cell is marked “end-of-message”). A more natural sacrifice algorithm would discard the first packet to have a cell overflow the VCQ. We use simulations to compare the loss probabilities that use this algorithm to the loss probabilities that result when sacrifice is by the fewest remaining untransmitted cells. We find that loss probabilities are higher when the latter algorithm is used.

This finding raises two interesting issues. A theoretical question is first: Which sacrifice algorithm minimizes packet losses from the VCQ? To begin, we examine (with simulation), a third algorithm that sacrifices the packet with the maximum number of cells awaiting transmission to the VCQ. Instead of seeking to free up as much space as possible at the time of sacrifice, like sacrifice by fewest remaining cells, this algorithm seeks to reduce the rate at which cells enter the
and compares the completed work sample paths for finite and
infinite VCQ's. The upper bound on the fraction of sacrificed
packets is valid under general arrival processes. Section III
specializes the results of Section II to Poisson packet arrivals.
Section IV derives the asymptotic formula for the required
tail probabilities and describes validation of the analysis via
simulation. Section V presents numerical results of the model
and compares, by simulation, the loss probabilities under three
sacrifice algorithms. Section VI summarizes the paper and
outlines areas for further investigation.

II. A MODEL OF PACKET LOSSES
FROM THE VIRTUAL CHANNEL QUEUE

Fig. 1 displays a set of host computers connected through
an ATM network (metropolitan or wide area) that may consist
of many interconnected ATM switches. Local area networks
could readily be substituted for the hosts. Our interest is in
one host and its VCQ that collects cells of packets from other
hosts. The host is connected to one ATM switch, handling
traffic for it and other network nodes.

A. Modeling Assumptions

1) The Link Between the ATM Switch and the Host: The
ATM switch to which our host is attached is likely to simulta-
aneously serve several virtual channels for the host. Therefore,
cells of different packets will interleave on their way to the
VCQ. The degree of interleaving depends on the packet and
cell arrival processes and the service discipline at the switch.

To facilitate the exposition, we consider an output-buffered
ATM switch with parallel switching planes (see e.g., [8]).
Although cells in the output buffer that feeds the link to the
host are likely to be served first-in first-out, cell interleaving
will make the packets, order of service closer to round robin.
As our interest is in packets in the VCQ at the host’s end of
the link, we need to model the link and VCQ. We view the
link between the ATM switch and our host as the server in
a queueing system and view packets as customers. In reality,
packets do not enter the link’s buffer instantaneously. Instead,
packets are offered to a network entry point and subdivided
into cells that make their way across the ATM network.
Deterministic service times at network elements, however,
should cause the cells of a packet to arrive at the link buffer
at roughly a constant rate. Whenever there is a queue for
the link, the link should serve packets as if entire packets were
present in the link buffer rather than just a few cells of each
packet at a time. For this reason, we idealize the packet arrival
process so entire packets arrive to the link buffer at once.
As we are not concerned with overflows from it we assume the
link buffer is infinite. Finally, we assume that only a finite
number of packets can arrive in any finite interval of time.

Fig. 2 displays the link from the ATM switch to the host
computer and its queue of packets awaiting transmission on
the link. The figure groups cells (the small squares) belonging
to the same packet, adjacent to one another in the queue for
the link, and shows the link controller (the circle) transmitting
cells of the different packets in a round robin fashion. On the
right is the Virtual Channel Queue.
2) The Virtual Channel Queue: The VCQ at the host’s end of the link accepts and stores cells for reassembly into packets. The VCQ is partitioned into units that can hold an ATM cell and some overhead, such as a pointer to the next cell in a packet. Different packets share buffer space in the VCQ in the sense that any cell may be written into any free unit. Fig. 2 illustrates this aspect of the VCQ by placing the cells of different packets into the VCQ in their order of arrival. The numbers in the cells designate the packets to which the cells belong.

Packets can leave the VCQ for several reasons. Preferably, they leave as complete packets. This circumstance occurs when the End-of-Message (EOM) cell of an unsacrificed packet enters the VCQ. The BTA either begins reading the packet out or, if it is busy reading another packet, enqueues the packet to be read out later. If all the cells of the packet have arrived to the VCQ without errors, the BTA will pass the packet to the host successfully. We do not consider higher layers of the protocol in this paper and consequently assume packets to be free of errors.

No matter how large the VCQ, it will sometimes fill with all packets incomplete. When this happens, some packet is sacrificed to free up buffer space. Otherwise, cell interleaving would cause many packets in progress to be lost (a packet is lost if any of its cells are). The BTA reads the sacrificed packet out of the VCQ and passes it to the next layer of the protocol, which may or may not be able to salvage it despite the cells it is missing at the time of readout. The BTA also passes subsequently arriving cells of such a packet to the next higher protocol layer. If a sacrificed packet is unusable, the packet will require retransmission. Again, because this paper concerns the VCQ and ATM layers, such retransmissions will not be considered here. Any packets sacrificed will be counted as lost and will be cleared from the system with no further effects upon it.

Packets with cells in the VCQ can be of three types: complete, sacrificed, or neither sacrificed nor complete. An important characteristic of the BTA allows us to simplify our model by keeping track only of incomplete, unsacrificed packets. The useful property is at the rate at which the BTA reads cells from the VCQ is greater than or equal to the link speed. As a result, the net VCQ contents decreases (or may stay the same in the case of equality) whenever the BTA is reading from the VCQ. In particular, the VCQ cannot overflow while the BTA is reading a packet out of the VCQ. The only time the VCQ overflows is when all cells in the VCQ belong to incomplete, unsacrificed packets and the VCQ fills. At that time, a sacrifice is necessary. Following a packet sacrifice or completion, no sacrifices need occur until enough cells have arrived to fill the space vacated by the packet.

Since the next time the VCQ fills depends only on the space vacated by a departing packet and subsequent arrivals to the VCQ, we may treat departures of packets from the VCQ as instantaneous for the purpose of analyzing VCQ overflows. Accordingly, we model the BTA readout rate as infinite. As soon as a packet completes or is sacrificed, it leaves the VCQ. As a result, in our model, the VCQ contains only unsacrificed packets.

B. The Completed Work Process and Losses

It will be convenient to approximate the round robin service discipline for packets traversing the link between the ATM switch and the host by the processor sharing discipline. Under processor sharing, the server (link) works on all customers simultaneously at a rate equal to \(1/n\) times its capacity when \(n\) customers are present. Call a packet active if the server is working on it. Each packet \(i\) brings an amount of work \(X_i\) to the server (the ATM cell is the unit of work). At any time \(t\), an amount \(R_i(t)\) of work on packet \(i\) remains to be done. The function \(R_i\) equals \(X_i\) until packet \(i\) arrives at the link and piecewise decreases linearly (with slope the reciprocal of the total number of active packets), hitting zero when the packet completes (ceases to be active). We include the endpoints of the interval over which \(R_i\) is nonconstant in a packet’s active period. Then the set \(A(t)\) of active packets is unambiguously defined.

If packet \(i\) is active and not sacrificed, then \(X_i - R_i(t)\) is the amount of work of packet \(i\) in the VCQ at time \(t\). Summing over all packets in the VCQ gives the completed work defined by

\[
C(t) = \sum X_i - R_i(t).
\]

The function \(C\) is piecewise linear with jump discontinuities. The graph of \(C(t)\) has slope \(n/a\) if \(a\) packets are active at time \(t\) and \(n\) of the active packets are not sacrificed. If no packets are active at time \(t\), then \(C(t) = 0\). The function \(C\) jumps downward by the size of a packet when a packet completes, and by the number of cells of a packet in the VCQ at the time a packet is sacrificed.

Fig. 3 is a conceptual view of the link, its queue of packets, and the VCQ at some time \(t\). The vertical line represents the server. It sweeps to the left at rate \(1/n\) times the link capacity when \(n\) packets are in service. When a new packet arrives, its right hand end abuts the line. When a packet finishes transmission on the link, it disappears from the figure. The figure shows all active packets and separates them into unsacrificed and sacrificed packets. The part of a packet to the left of the line is the remaining work of a packet (the cells yet to be transmitted on the link). The part of a packet to the right of the line contributes to the completed work if the packet is not sacrificed. Whenever a packet is sacrificed, it moves from
the box labeled “VCQ Contents” to the box labeled “Sacrificed Packets.” Everything to the right of the vertical line would be the VCQ contents if the VCQ were infinite.

Sacrifices occur only at instants when $C(t)$ equals the VCQ size. Denote by $b$ the size of the VCQ in cells. We assume that $b$ is finite and larger than the maximum packet size. When $C(t) = b$, the VCQ is full and some packet, say packet $s$, is sacrificed to make room for cells of other packets. At this point, $C$ jumps downward by $X_s - R_s(t)$. If no EOM’s of unsacrificed packets arrive before $C(t)$ again reaches level $b$, another packet will have to be sacrificed. We will shortly define $C$ rigorously to make the following fact true.

**Fact 1:** The number of packets lost up to time $t$ equals the number of times in $[0, t]$ that the function $C$ hits the boundary at level $b$.

### C. Rigorous Definition of $C$

For the analysis to follow, we need a mathematical definition of $C$. We shall define $C(t)$ inductively in terms of an auxiliary sequence of functions $C_n(t)$. We assume in the construction that some algorithm for sacrificing packets is given.

Suppose the system starts at time zero with the VCQ not full (e.g., empty). Define

$$C_0(t) = \sum_{i \in A(t)} X_i - R_i(t).$$

The definition of $A(t)$ makes the function $C_0$ left-continuous. In addition, $C_0$ has right-hand limits, that is, a discrete jump set, and a jump discontinuity of size $X_i$ when packet $i$ completes. Note, for later use, that $C_0$ is the completed work at the ATM switch when the VCQ is infinite. Since there is no notion of sacrifice when the VCQ is infinite, the graph of $C_0$ is simpler than that of $C$. Whenever at least one packet is active, $C_0$ increases at rate 1 except at its downward jumps.

We next define inductively the sequence $\{T_n\}$ of times that $C(t)$ hits the boundary $b$. Define $T_0$ to be zero and

$$T_1 = \inf \{ t > 0 : C_0(t) = b \}$$

the first time $C_0$ hits $b$ after time zero. If $T_1 = \infty$, the construction ends here. Otherwise, let $s$ be the packet chosen for sacrifice at time $T_1$. Define $S_1 = \{s\}$, the set consisting of $s$ alone. $S_1$ will be the first of an increasing sequence of sets of sacrificed packets.

Define now $C_1(t) = C_0(t)$ for $t \leq T_1$ and

$$C_1(t) = \sum_{i \in A(t) - S_1} X_i - R_i(t), \text{ for } t > T_1.$$

For the rest of the construction, assume inductively that $T_n$, $S_n$, and $C_n$ are defined. Set

$$T_{n+1} = \inf \{ t > T_n : C_n(t) = b \}.$$ 

Provided $T_{n+1}$ is finite, let $s'$ be the packet sacrificed at that time and define

$$S_{n+1} = S_n \cup \{s'\}.$$ 

Finally, set $C_{n+1}(t) = C_n(t)$ for $t \leq T_{n+1}$ and

$$C_{n+1}(t) = \sum_{i \in A(t) - S_{n+1}} X_i - R_i(t), \text{ for } t > T_{n+1}.$$ 

If at any point in the construction, some $T_{n+1}$ is infinite ($C_{n+1}$ never hits $b$) then $C_{n+1}$ is not defined and the construction ends with $C_n$. Observe that $C_{n+1} \leq C_n$ for all $n$, i.e., the sequence $\{C_n\}$ is a decreasing sequence of functions bounded below by zero. Consequently, the sequence has a limit. 

**Definition of $C$:** $C(t) = \lim_{n \to \infty} C_n(t)$

The function $C$ is hard to analyze directly. As noted earlier, its rate of increase varies with the relative numbers of unsacrificed and active packets. Another difficulty in analyzing $C$ appears when one introduces probabilistic assumptions. The sequence $\{T_n\}$ of times when $C$ hits $b$ seems not to enjoy any special properties that allow one to relate the frequency of hits to the equilibrium distribution of $C$. In addition, the equilibrium distribution itself is hard to analyze.

The difficulties in analyzing $C$ directly prompt us to study $C$ by comparing it to the completed work function in the infinite buffer case. This function is the function $C_0$ defined above, however, we shall relabel $C_0$ with the more suggestive notation $C_0^\infty$ as a reminder of the infinite buffer case.

Our first result relating $C$ and $C_\infty$ asserts that when no sacrificed packets are active, $C$ and $C_\infty$ agree. For its statement, we need

**Definition:** The departure time $D_n$ of the $n$th sacrificed packet is the time the $n$th sacrificed packet completes service on the link from the ATM switch to the VCQ. Formally, if $s$ was the $n$th sacrificed packet, then

$$D_n = \inf \{ t > T_n : R_s(t) = 0 \}.$$ 

Observe that if $t > \max(D_1, \cdots, D_n)$, then $C_\infty(t) = C_n(t)$.

**Theorem 1:** At any point $t$ not contained in any interval $[T_n, D_n]$, $C(t) = C_\infty(t)$.

**Proof:** Given $t$, choose $m$ such that $T_m < t < T_{m+1}$. Since $t$ belongs to none of the intervals $[T_n, D_n]$ for $n \leq m$, no packet in $S_m$ is active at time $t$: $A(t) = A(t) - S_m$. As the left and right-hand sides are the index sets for the sums defining $C_\infty$ and $C_m$, it follows that $C_\infty(t) = C_m(t)$. By our choice of $m$, $C_m(t) = C(t)$.

### D. $C_\infty$ when Sacrifice is by Minimal Remaining Work

The comparison between $C$ and $C_\infty$ is manageable provided we assume whenever $C$ hits $b$, the sacrificed packet is that packet among those currently active having minimal remaining work $R$ (and is not already sacrificed). Under this algorithm, the packet sacrificed at time $T_{n+1}$ is that $s \in A(T_{n+1}) - S_n$ such that $R_s(T_{n+1})$ is minimal among $\{R_s(T_{n+1}) : s \in A(T_{n+1}) - S_n\}$.

Our main analytical result is that with selection by minimal remaining work, each reflection of $C$ from the boundary at $b$ corresponds to a jump of $C_\infty$ from above $b$. This result bounds the number of packets lost by the number of such jumps.

**Theorem 2:** For $t$ in the interval $[T_n, D_n]$, $C_\infty(t) > b$. 

Proof: Because the packet sacrificed at time \( T_n \) is that with minimal remaining work, every unsacrificed packet active at time \( T_n \) is still active at time \( t \). In other words, \( A(T_n) - S_{n-1} \subseteq A(t) - S_{n-1} \) for \( T_n < t \leq D_n \). Since \( A(t) - S_{n-1} \) is the index set for the sum defining \( C_{n-1} \) when \( t > T_{n-1} \), it follows that

\[
C_{n-1}(t) \geq \sum_{i \in A(T_n) - S_{n-1}} X_i - R_i(t)
\]

\[
= \sum_{i \in A(T_n) - S_{n-1}} [X_i - R_i(T_n) + R_i(T_n) - R_i(t)]
\]

\[
= \sum_{i \in A(T_n) - S_{n-1}} X_i - R_i(T_n) + \sum_{i \in A(T_n) - S_{n-1}} R_i(T_n) - R_i(t)
\]

\[
= C_{n-1}(T_n) + \sum_{i \in A(T_n) - S_{n-1}} R_i(T_n) - R_i(t)
\]

As \( C_{n-1}(T_n) = b \) and the differences of the remaining works in the sum are positive, we find \( C_{n-1}(t) > b \). Since \( C_{\infty} \geq C_{n-1} \), the result follows.

Remarks:

1) The function \( C_{\infty} \) may jump downward during the interval \((T_n, D_n)\) when any of several kinds of packets complete their service on the link. First, packets active at time \( T_n \) but sacrificed earlier (i.e., the set \( A(T_n) \cap S_{n-1} \)) may complete service on the link. The proof excludes such packets from consideration by working with \( C_{n-1} \). Second are packets active at time \( T_n \) but not yet sacrificed. The proof deals explicitly with these. Third are packets that begin and complete service on the link in the interval \((T_n, D_n)\). Such packets contribute to \( C_{n-1} \), but are dropped on the right hand side of the first inequality of the proof.

2) The assumption that a packet with minimal remaining work is sacrificed is used to ensure that no unsacrificed packets active at time \( T_n \) complete service on the link, thereby causing \( C_{\infty} \) to jump downwards before time \( D_n \). These packets alone are enough to keep \( C_{\infty} \) above level \( b \) on the interval \((T_n, D_n)\).

The proof constructs a one-to-one mapping of the set of sacrificed packets into (not onto) the set of jump points of \( C_{\infty} \) from above level \( b \). The sacrifice at time \( T_n \) corresponds to the jump at time \( D_n \). The proof seems hard to generalize to other sacrifice algorithms because of the possibility of several sacrifices occurring before \( C_{\infty} \) jumps below level \( b \). One can map the first sacrifice to the first jump of \( C_{\infty} \) from above \( b \). It is not clear how to find a jump for the next sacrifice. The jump at the \( D \)-time of the second sacrificed packet may not be from above \( b \). In such a case, one cannot map the sacrifices injectively to the set of jump points above \( b \).

3) The mapping of sacrifices to jumps of \( C_{\infty} \) from above \( b \) is not onto (surjective). There may be jumps that do not correspond to sacrifices. For example, if a packet arrives between \( T_n \) and \( D_n \) and has a short enough work time that all its cells traverse the link before \( D_n \), then \( C_{\infty} \) will jump downward by the work time of the packet upon departure. The theorem shows, however, that \( C_{\infty} \) will not jump below \( b \). Note that the short packet may be sacrificed. If so, \((T, D)\) intervals will be nested.

Corollary 1: Under sacrifice by minimal remaining work, the only times that \( C_{\infty} \) jumps from above \( b \) are when \( 1 \) a sacrificed packet completes service on the link or \( 2 \) a packet arrives and completes service on the link within a \((T, D)\)-interval. The two events are not exclusive. Hence, the number of packets sacrificed up to time \( t \) is bounded above (asymptotically) by the number of jumps \( C_{\infty} \) makes from above \( b \).

The bound is asymptotic because \( C_{\infty} \) jumps sometime after the sacrifice.

III. THE DISTRIBUTION OF \( C_{\infty} \) WITH POISSON PACKET ARRIVALS

Let \( \{J_k : k = 1, 2, 3, \ldots\} \) be the sequence of times when \( C_{\infty} \) jumps downward (the times when the packet completes service on the link). According to Corollary 1, the number of packets sacrificed among the first \( K \) that leave the VCQ is, at most, the number of the \( C_{\infty}(J_k) - 1 \); \( k = 1, 2, 3, \ldots, K \) greater than \( b \) (the \(-\) in \( J_k \) emphasizes that we consider the value of the left-continuous function \( C_{\infty} \) just before the jump). Dividing the number of packets lost by \( K \) and letting \( K \to \infty \) gives the asymptotic fraction of packets lost.

The problem is in calculating this loss probability. In this section, we introduce probabilistic assumptions allowing us first to relate the distribution of \( C_{\infty} \) at jump times to the equilibrium distribution of \( C_{\infty} \) and then to compute the equilibrium distribution. With the equilibrium distribution in hand, it is possible to compute the packet loss probability numerically.

The packet arrival stream to the link between the ATM switch and the host is a superposition of traffic streams from individual users. The user population is large but at any given time, most users are not engaged in sessions. We might represent each user’s packet transmission process as a renewal process with frequent short interarrival times (modeling a session with the host) and rarer very long interarrival times (temporarily among the times between sessions). A large number of such arrival streams resembles the conditions of the Palm–Khinchin theorem [9], that asserts a limit of superimposed, sparse renewal streams is a Poisson process. However, the Palm–Khinchin theorem requires that each stream thins in a precise fashion as the number of streams becomes large. In our setting, it may be reasonable to thin streams by requiring longer periods of inactivity as the number of streams grows, but it is not correct to lengthen the interarrival times between packets during user sessions as required by the theorem.

Although the theorem does not strictly support the use of a Poisson limit of arrival streams, we will assume that packets bound our host arrival to the link according to a Poisson process. The two advantages of this assumption are its tractability and it does not impose an a priori bound on the number of active packets. Such a bound (if small) would simplify VCQ engineering considerably. The importance of the arrival process model is reduced in this paper by the fact that our more interesting conclusions are comparisons.
between sacrifice algorithms, which may not depend on arrival processes. In addition to assuming Poisson packet arrivals, we assume that packet lengths are independent and identically distributed. Denote by $X_i$ the length of packet $i$ and let $\lambda$ be the Poisson rate of arrivals of packets.

These and our earlier assumptions about cell interleaving at the output port of the ATM switch make the output link and its buffer a processor sharing M/G/1 queue. We assume the arrival rate and service times result in a total server occupancy less than 1 and the system is in equilibrium.

Departures of packets from the M/G/1 processor sharing queue form a Poisson process (see e.g., [10]). PASTA [11], therefore, relates the equilibrium and jump time distributions of $C_\infty$. Precisely

$$\lim_{k \to \infty} \frac{1}{k} \sum_{n=1}^{K} I[C_\infty(J_n+) > b] = \lim_{t \to \infty} \frac{1}{t} \int_{0}^{t} I[C_\infty(s) > b] ds$$

almost surely (the $I$'s are indicators). Because we assumed a total server occupancy $< 1$, the ergodic theorem or regenerative arguments allow us to identify the second time average with the equilibrium probability that $C_\infty$ is greater than $b$. Similarly, $C_\infty(J_n-) = C_\infty(J_n+)$ plus the work of the packet departing at time $J_n$. Therefore, $C_\infty(J-)$ has the distribution of $C_\infty + X$. The next task is to analyze the equilibrium distribution of $C_\infty$. For this, we will need the equilibrium distribution of the number of active packets on the link and the distribution of each active packet's completed work. We recall some results from [10]:

1) The total number $N$ of customers in an M/G/1 processor sharing queue has a geometric distribution

$$P\{N = n\} = (1 - \rho)\rho^n.$$ 

Here, $\rho$ is the server occupancy, given by $\lambda E(X)$.

2) Given $N$ customers in the M/G/1 processor sharing queue, the remaining works $R_n$ (or the completed works $X_n - R_n$, $n = 1, 2, 3, \ldots, N$) are independent of one another and and $N$ are identically distributed.

3) If $F$ is the distribution function of the $X_n$, then the $R_n$ or $X_n - R_n$ have equilibrium distribution function

$$G(x) = 1/E(X) \int_{0}^{x} (1 - F(u)) du.$$ 

By items 1 and 2, $C_\infty$ has the distribution of the sum of $N$ independent remaining works:

$$C_\infty = \sum_{n=1}^{N} R_n.$$ 

Remark: The preceding analysis for Poisson arrivals goes through for more general arrival processes, e.g., a finite source model (a two–node network consisting of a processor sharing node and an infinite server node with a finite number of customers in the network.) The sojourn time at the infinite server node may be general. One need only replace the geometric distribution by the distribution of customers at the processor sharing node.

IV. NUMERICAL ANALYSIS AND VALIDATION

The bound for the fraction of packets sacrificed is given by the tail probability $P(C_\infty + X > b)$. In this section, we calculate this tail probability approximately. The method interprets $X + C_\infty = X + \sum_{n=1}^{N} R_n$ as a terminating renewal process and estimates its tail probabilities asymptotically.

A. An Asymptotic Estimate

If we view $X$ and the $R_n$ as interarrival times, then $X + C_\infty$ is the epoch of termination for a terminating renewal process [3]. The exceptional term $X$ is the first interarrival. Since its distribution $F$ differs from that of the later interarrivals, the process is delayed, in the terminology of [3]. The later interarrival times have a common defective distribution function $L(x) = \rho G(x)$ with probability $1 - L(\infty) = 1 - \rho$ of being infinite and terminating the process.

An easy extension of Feller’s arguments shows the function $Z(x) = P\{X + C_\infty \leq x\}$ satisfies the renewal (6.4) of [3], XI.6 with $z(x) = (1 - \rho)F(x)$. Theorem 2 of [3], XI.6 states that the asymptotic behavior of $Z$ is

$$\mu_\# e^{-\kappa z} [Z(\infty) - Z(x)] \to \frac{1 - \rho}{\kappa}$$

$$+ \int_{0}^{\infty} e^{\eta y} [z(\infty) - z(y)] dy.$$ 

In this formula, $\kappa$ is the root of the equation

$$\int_{0}^{\infty} e^{\eta y} L(dy) = 1.$$ 

Equivalently, if we define

$$\Phi(s) = \int_{0}^{\infty} e^{\eta y} F^c(y) dy$$

then $\kappa$ is the solution to $\lambda \Phi(s) = 1$. The constant $\mu_\#$ is given by

$$\mu_\# = \lambda \Phi'(\kappa) = \lambda \int_{0}^{\infty} ye^{\eta y} F^c(y) dy.$$ 

Both $\kappa$ and $\mu_\#$ are most easily found integrating by parts in the definition of $\Phi$ to get

$$\Phi(s) = \phi(s)/s - 1/s$$

where

$$\phi(s) = \int_{0}^{\infty} e^{\eta y} F(dy).$$ 

Differentiating and using the definition of $\kappa$ leads to

$$\mu_\# = \lambda \phi'(\kappa)/\kappa = \lambda \phi'(\kappa)/\kappa - 1.$$ 

The final form of the asymptotic expression for $Z$ is

$$P\{X + C_\infty > x\} \sim \frac{(1 - \rho)(\kappa^{-1} + \lambda^{-1}) e^{-\kappa x}}{\mu_\#}.$$ 

To implement the asymptotic method, we first solved the equation $\lambda \Phi(s) = 1$ for $\kappa$ by the Newton–Raphson method. The remainder of the calculations were then routine.
TABLE I
MODEL AND SIMULATION LOSS PROBABILITIES AT 90% LINK UTILIZATION

<table>
<thead>
<tr>
<th>p1</th>
<th>p2</th>
<th>p3</th>
<th>VCO Size</th>
<th>Model</th>
<th>Sim 1</th>
<th>Sim 2</th>
<th>Sim 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>16</td>
<td>4.17e-02</td>
<td>4.18e-02</td>
<td>4.24e-02</td>
<td>4.16e-02</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>32</td>
<td>1.52e-03</td>
<td>1.65e-03</td>
<td>1.37e-03</td>
<td>1.65e-03</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>48</td>
<td>5.52e-05</td>
<td>6.13e-05</td>
<td>6.88e-05</td>
<td>3.15e-05</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>60</td>
<td>4.59e-06</td>
<td>1.05e-06</td>
<td>7.48e-06</td>
<td>4.11e-06</td>
</tr>
<tr>
<td>.33</td>
<td>.33</td>
<td>.34</td>
<td>16</td>
<td>2.49e-02</td>
<td>2.28e-02</td>
<td>2.37e-02</td>
<td>2.21e-02</td>
</tr>
<tr>
<td>.33</td>
<td>.33</td>
<td>.34</td>
<td>32</td>
<td>5.89e-04</td>
<td>5.53e-04</td>
<td>5.69e-04</td>
<td>6.58e-04</td>
</tr>
<tr>
<td>.33</td>
<td>.33</td>
<td>.34</td>
<td>48</td>
<td>1.39e-05</td>
<td>3.13e-05</td>
<td>1.04e-05</td>
<td>1.32e-05</td>
</tr>
<tr>
<td>.33</td>
<td>.33</td>
<td>.34</td>
<td>60</td>
<td>8.40e-07</td>
<td>6.93e-07</td>
<td>6.94e-06</td>
<td>0.0e-06</td>
</tr>
<tr>
<td>.75</td>
<td>.25</td>
<td>.25</td>
<td>16</td>
<td>3.12e-02</td>
<td>2.11e-02</td>
<td>2.20e-02</td>
<td>2.12e-02</td>
</tr>
<tr>
<td>.75</td>
<td>.25</td>
<td>.25</td>
<td>32</td>
<td>9.86e-04</td>
<td>6.08e-04</td>
<td>1.13e-03</td>
<td>8.34e-04</td>
</tr>
<tr>
<td>.75</td>
<td>.25</td>
<td>.25</td>
<td>48</td>
<td>3.11e-05</td>
<td>2.90e-05</td>
<td>2.87e-05</td>
<td>8.50e-06</td>
</tr>
<tr>
<td>.75</td>
<td>.25</td>
<td>.25</td>
<td>60</td>
<td>2.33e-06</td>
<td>3.00e-07</td>
<td>5.70e-07</td>
<td>4.12e-06</td>
</tr>
</tbody>
</table>

TABLE II
MODEL AND SIMULATION LOSS PROBABILITIES AT 80% LINK UTILIZATION

<table>
<thead>
<tr>
<th>p1</th>
<th>p2</th>
<th>p3</th>
<th>VCO Size</th>
<th>Model</th>
<th>Sim 1</th>
<th>Sim 2</th>
<th>Sim 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>16</td>
<td>1.38e-03</td>
<td>1.39e-03</td>
<td>1.35e-03</td>
<td>1.44e-03</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>24</td>
<td>4.31e-05</td>
<td>4.13e-05</td>
<td>2.95e-05</td>
<td>4.05e-05</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>32</td>
<td>1.37e-06</td>
<td>0.0e-07</td>
<td>7.00e-07</td>
<td>5.00e-07</td>
</tr>
<tr>
<td>.33</td>
<td>.33</td>
<td>.34</td>
<td>16</td>
<td>4.84e-04</td>
<td>5.17e-04</td>
<td>4.20e-04</td>
<td>4.87e-04</td>
</tr>
<tr>
<td>.33</td>
<td>.33</td>
<td>.34</td>
<td>24</td>
<td>1.00e-05</td>
<td>7.85e-06</td>
<td>2.85e-05</td>
<td>4.80e-06</td>
</tr>
<tr>
<td>.33</td>
<td>.33</td>
<td>.34</td>
<td>32</td>
<td>2.08e-07</td>
<td>0.0e-07</td>
<td>1.00e-07</td>
<td>0.0e-07</td>
</tr>
<tr>
<td>.75</td>
<td>.25</td>
<td>.25</td>
<td>16</td>
<td>7.63e-04</td>
<td>5.21e-04</td>
<td>5.40e-04</td>
<td>4.77e-04</td>
</tr>
<tr>
<td>.75</td>
<td>.25</td>
<td>.25</td>
<td>24</td>
<td>2.12e-05</td>
<td>1.57e-05</td>
<td>2.06e-05</td>
<td>6.29e-05</td>
</tr>
<tr>
<td>.75</td>
<td>.25</td>
<td>.25</td>
<td>32</td>
<td>5.89e-07</td>
<td>0.0e-07</td>
<td>0.0e-07</td>
<td>4.40e-06</td>
</tr>
</tbody>
</table>

B. Simulation of Sacrifice by Minimal Remaining Work

Our theoretical result is the steady state fraction of packets lost, when the packet with minimal remaining work is sacrificed, is bounded by the fraction of \( C_{\infty} \) jumps from above level \( b \). This section addresses the tightness of the bound with simulation. We find that the bound is very tight. In some simulation runs, the measured fraction of packets lost is greater than the bound. The simulations, however, verify the bound statistically.

The simulation represents the link by a server that processes the cells of any packets in service in a round robin fashion. The simulation tracks which active packets are sacrificed and measures the number of cells in the VCO belonging to active, unsacrificed packets. The packet arrival process is Poisson.

We use three different distributions of packet lengths. The packet layer protocol permits a maximum packet length of \( L = 64 \) kilobytes. The nature of host-to-host communication (large file transfers) suggests a sizable fraction of packets will be of maximal length. Therefore, our packet length distributions have substantial mass at \( L \) and some spread over smaller values. The three distributions are as follows:

- **X:** all packets are of length \( L = 64 \) kilobytes.
- **Y:** 1/3 of the packets have length near 1 kilobyte, 1/3 are of length \( L \), and 1/3 are uniformly distributed between 1 kilobyte and \( L \).
- **Z:** 75% of packets are near 1 kilobyte and 25% are of maximal length.

"Near 1 kilobyte" is achieved with a two-stage Erlangian distribution with a mean of 1 kilobyte.
event at least 16 out of 21 times is about 1%, which leads us to reject the hypothesis.

Fig. 4 also shows the normalized spread of the distribution of loss probabilities grows with the VCQ size. To gain an intuitive grasp on this, note that packet losses occur bunched together in "episodes." In a typical episode, the completed work function $C$ hits the boundary at $b$ several times over a time interval much shorter than the period during which $C$ makes large excursions away from $b$. Let $\tau$ be a generic inter-episode time. The larger the VCQ, the larger $\tau$ tends to be. In addition, because the occurrence of an episode requires more conditions to be met when the VCQ is large (for example, a large number of packets arriving within a short interval), we would expect the relative variability of $\tau$ to increase with the VCQ size. That is, the coefficient variation of $\tau$, $CV(\tau)$, increases with VCQ size.

As a heuristic model, suppose the successive times $\tau_n$, $n = 1, 2, \ldots$, between episodes are independent and identically distributed. Denote by $M$ the number of episodes in an interval of length $T$ (e.g., the length of a simulation run). The central limit theorem for the number of renewal epochs [3], XLO states, for large $T$, $M$ is approximately normally distributed with coefficient of variation $CV(M)^2 \approx CV(\tau)^2 E(\tau)/T$. Next, let $Y_m$ be the number of packets lost in the $m$th episode. Then $S_M = \sum_{m=1}^{M} Y_m$ is the number of packets lost in time $T$.

As a surrogate for the ratio whose log appears in Fig. 4, use the coefficient of variation $CV(S_M)$ of $S_M$. Assuming $M$ and $Y_m$ independent and the $Y_m$ identically distributed, $CV(S_M)^2 = CV(Y)^2/EM + CV(M)^2$ or $CV(S_M)^2 \approx (CV(Y)^2 + CV(\tau)^2) E(\tau)/T$. Since both $E(\tau)$ and $CV(\tau)$ increase with the VCQ size, so does $CV(M)$ as long as $CV(Y)$ does not decrease.

V. NUMERICAL RESULTS

Packet losses from the VCQ depend on four factors: the packet arrival process, the packet length distribution, the VCQ size, and the sacrifice algorithm. The first subsection below explores the dependence of losses on the VCQ size, the server occupancy, and the packet length distribution when sacrifice is by minimal residual work. The next subsection compares losses under this sacrifice algorithm to those when the packet that overflows the VCQ is the one sacrificed and those when the packet with maximal remaining work is sacrificed.

A. Packet Losses Under Sacrifice by Minimal Remaining Work

The preceding section showed the tightness of our bound on packet loss rates. Thus, we can use the model to predict loss probabilities. This section explains the dependence of losses on the packet length mix and the link utilization. Figs. 5 and 6 present results in terms of the mean number of active packets, the relation between the mean number of active packets, and the link utilization is $E(N) = \rho/(1 - \rho)$.

Fig. 5 shows the loss probability as a function of VCQ size (in units of $L$ bytes) for two different load levels: four virtual circuits up on average ($\rho = 0.8$) and 9 virtual circuits up ($\rho = 0.9$). In Fig. 6, the mean number of active packets is the independent variable and loss probability corresponding to two buffer sizes appear. Three features are obvious from the figures:

- The loss probabilities are highest when all packets are of maximum length (case X).
- The loss probabilities are lowest in case Y, the 1/3, 1/3, 1/3 mix.
- The VCQ size required to attain a given loss probability (e.g., $10^{-4}$) is very sensitive to the mean number of active packets but relatively insensitive to the packet length mix.
TABLE III

<table>
<thead>
<tr>
<th>$p_1$</th>
<th>$p_2$</th>
<th>$p_3$</th>
<th>VQ Size</th>
<th>Min</th>
<th>Rem</th>
<th>Overflow</th>
<th>Ratio</th>
<th>Mix</th>
<th>Rem</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>16</td>
<td>4.18e-02</td>
<td>1.91e-02</td>
<td>2.2</td>
<td>2.63e-02</td>
<td>1.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>32</td>
<td>1.65e-03</td>
<td>5.96e-04</td>
<td>2.8</td>
<td>1.81e-03</td>
<td>1.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>48</td>
<td>6.15e-05</td>
<td>1.63e-05</td>
<td>2.7</td>
<td>3.31e-05</td>
<td>1.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.33</td>
<td>.33</td>
<td>.34</td>
<td>16</td>
<td>2.28e-02</td>
<td>6.87e-03</td>
<td>3.3</td>
<td>8.18e-03</td>
<td>2.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.33</td>
<td>.33</td>
<td>.34</td>
<td>32</td>
<td>5.53e-04</td>
<td>1.08e-04</td>
<td>5.0</td>
<td>1.79e-04</td>
<td>3.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.33</td>
<td>.33</td>
<td>.34</td>
<td>48</td>
<td>3.18e-05</td>
<td>5.96e-06</td>
<td>5.3</td>
<td>8.38e-06</td>
<td>3.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.75</td>
<td>0</td>
<td>.25</td>
<td>16</td>
<td>2.11e-02</td>
<td>4.16e-03</td>
<td>5.1</td>
<td>5.51e-03</td>
<td>3.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.75</td>
<td>0</td>
<td>.25</td>
<td>32</td>
<td>6.07e-04</td>
<td>7.51e-05</td>
<td>8.1</td>
<td>1.34e-04</td>
<td>4.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.75</td>
<td>0</td>
<td>.25</td>
<td>48</td>
<td>2.89e-05</td>
<td>2.72e-06</td>
<td>10.6</td>
<td>4.90e-06</td>
<td>5.9</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The first point is true because the deterministic distribution concentrated at $L$ gives a value for $P\{C_{\infty}(J-) > b\}$ greater than or equal to the value corresponding to any length distribution bounded above by $L$. Recall that a random variable $X_k$ is stochastically smaller than $X_k$ if $P\{X_1 > a\} \leq P\{X_2 > a\}$ for all $a$. It is straightforward to show that if $X_l$ is a random variable bounded above by $L$, then the remaining work belonging to $X$ is stochastically smaller than the remaining work belonging to the deterministic random variable $L$. It follows that $C_{\infty}$ for $X$ is stochastically smaller than $C_{\infty}$ for $L$ (assuming equal $\rho$).

The implication of the first point is that to engineer the VCQ for a prescribed loss probability, consideration of the distribution in which all packets have length $L$ suffices. Still, it is of interest to examine other distributions to see whether over-engineering occurs. For example, to realize a loss probability under $10^{-9}$ when an average of four packets are active, the VCQ must be of size 32 when all packets are at maximum, as opposed to size 29 in case $Y$. The corresponding required VCQ sizes, when the mean number of active packets is nine, are 66 and 59. Note that losses in this paper are losses of packets, not ATM cells, and packets can contain over a thousand cells. This explains loss objectives several orders of magnitude greater than the ATM cell loss probability of $10^{-9}$ commonly used as a target.

The second and third bullet items can be explained by Theorem 3.8 in [10], which states, given the number of active packets, the probability that a packet is of a specified type (which distribution in the mix it comes from) is proportional to the product of fraction of packets of that type and the mean length of that type. Thus, in the 75%/25% mix, the 64–1 ratio of mean packet lengths makes the results very close to those for deterministic packet lengths. The VCQ requirements for the 1/3, 1/3, 1/3 mix are smaller because the shorter-than-maximal packets from the uniform distribution comprise on average 1/3 of the active packets. In any event, maximal packets dominate enough so the differences in VCQ requirements are only a few maximum packet lengths. Therefore, worst-case engineering of the VCQ is not excessive for the packet length distributions we have used.

The figures indicate that the VCQ size requirements are far more sensitive to the mean number of active packets (or server occupancy) than to the packet length distribution. This is attributable in part to the Poisson arrival assumption, which permits an unbounded number of active packets. For values of $\rho$ near 1.0, the tail of the (geometric) distribution for the number of active packets is long. For example, when $\rho = 0.9$, the probability that at least 40 packets are active is about 0.015. Even though the Poisson traffic model we have used lacks the burstiness that might be expected in the ATM setting (though the Poisson arrivals are packets, not cells), the resulting tail behavior of the distribution of $N$ forces a large VCQ.

The sensitivity of loss probabilities to load points to a need for a better traffic characterization is discussed below.

B. Comparison of Sacrifice Algorithms

The analytical model of Sections II and III bounds packet loss probabilities when the packet sacrificed has with minimal work remaining at the ATM switch. A more realistic sacrifice algorithm would purge cells of the packet that causes the VCQ to overflow. The theoretical calculation of packet loss probabilities seems difficult under this algorithm. We estimate the loss probabilities using simulation, therefore, and compare them and the resulting VCQ size requirements to the corresponding quantities under sacrifice by minimal remaining work.

We next compare the two sacrifice algorithms to a third, which sacrifices the packet with maximal remaining work. The reason for this choice is that sacrifice by minimal remaining work seems to be too “greedy,” as evidenced by loss rates about several times higher than the loss rates with sacrifice by overflow. Thus, we investigate the other extreme in which the packet using the most VCQ space in the future is discarded.

1) Sacrifice by Minimal Remaining Work and Overflow: The principal conclusions about the comparison between sacrifice by minimal remaining work and sacrifice by overflow is that losses are higher with sacrifice by minimal remaining work.

Table III illustrates loss rates under the different sacrifice algorithms. Each line of the table is the output of three simulation runs differing only in the sacrifice algorithm (in particular, the random number streams generating arrival times and packet lengths are the same in each pair of runs). The column after “overflow” labeled “ratio” is simply the ratio of the sacrifice by minimal remaining work loss rate to the sacrifice by overflow loss rate. Since the results in Table III are based on single simulation runs, we cannot draw conclusions from them without further work. However, multiple, independent simulation runs established that the differences between the losses in each algorithm were significant. First of all, such runs established that the losses in the sixth line of the table displayed the most variability from run-to-run. In that line, for sacrifice by minimal remaining work (min rem), the mean over eight independent runs was 1.39e-05 and the standard deviation was 9.94e-06. The mean for sacrifice by overflow was 2.50e-06, with standard deviation 1.76e-06. We ran standard statistical tests for paired data (since runs were paired with the same random number seeds). The 99% confidence interval for the mean difference of loss probabilities ranged from 21% of the mean to 178% of the mean. In particular, since the interval did not include zero, and the cases on this line were most variable, we conclude the tests demonstrated statistically, the loss probabilities under sacrifice by minimal remaining work were indeed larger than with sacrifice by overflow.
The reason for the differences in loss probabilities in cases Y (1/3, 1/3, 1/3) and Z (0.75, 0, 0.25) is fairly easy to see. When sacrifice is by minimal remaining work and there is a large difference in packet lengths, the packets sacrificed tend to be short. When sacrificed, they free up relatively little space in the VCQ, thereby necessitating another sacrifice soon. Fig. 7 displays cumulative distributions of the lengths of packets sacrificed using the two sacrifice algorithms (case Y). Under sacrifice by minimal remaining work, about 22% of sacrificed packets have a length under 3600 bytes and 42% of sacrificed packets are of maximal length. In contrast, when sacrifice is by overflow, fewer than 1% of the sacrificed packets are shorter than 3600 bytes, whereas 68% of all sacrificed packets are at maximum.

Fig. 8 shows the corresponding cumulative distributions for case Z. With sacrifice by minimal remaining work, some 60% of sacrificed packets are about 3600 bytes or less and 40% are of maximum length. With sacrifice by overflow, about 6% of sacrificed packets are under 3600 bytes and about 94% are of maximum length.

The reasons for the differences in loss probabilities in case X (all packets of maximum length) are less transparent. Here, we observe that about two or three times as many packets are sacrificed when the minimum remaining work algorithm is used. An intuitive explanation of the difference is as follows: Whenever a packet completes, it frees up 64 kilobytes of VCQ space without causing any losses. The sacrifice by the minimum remaining work algorithm selects the packet that will free up the most space. In so doing, it discards the packet that would complete next were the need to sacrifice not present. In this sense, sacrifice by minimum remaining work is locally, but not globally, optimum. In contrast, sacrifice by overflow selects a "random" packet for sacrifice, increasing the chances for a whole packet, with the minimum remaining work, to complete without being sacrificed.

2) Sacrifice by Maximum Remaining Work: We compared sacrifice by maximum remaining work to the other two sacrifice algorithms. Table III shows loss probabilities (max rem) and the ratios of the loss probabilities with sacrifice by minimum remaining work to the loss probabilities with sacrifice by maximum remaining work (last column). Over multiple runs in the sixth line of Table III, the mean loss probability was 4.58e-06 and the standard deviation was 2.33e-06. Again, statistical tests of paired data (sacrificed by overflow and maximum remaining work) established the significance of the differences in means at the 99% level. The 99% confidence interval for the mean of the losses under sacrifice by maximum remaining work minus losses under sacrifice by overflow, normalized by the mean, was [0.35, 1.64]. Again, as the interval did not contain zero, the conclusion is that losses under sacrifice by maximum remaining work lie between the losses seen with sacrifice by minimum remaining work and sacrifice by overflow.

Even though sacrifice by minimum remaining work is too greedy, the other extreme is not optimal. The fact that sacrifice by minimum remaining work is the worst algorithm in terms of packet loss leads us to ask whether our analytical model provides a bound on loss probabilities for all sacrifice algorithms.

VI. SUMMARY AND FUTURE WORK

This paper studied a buffer that collects ATM cells for reassembly into packets. The problem was formulated for a host computer communicating over an ATM network with other computers, but the buffer sizing problem is generic to any data communications network in which one protocol layer "partitions" or "divides up" the data units from a higher layer for transport. Each time such a buffer fills with incomplete packets, some packet must be sacrificed from the buffer. The
paper used the completed work process as a tool to count the equilibrium fraction of packets sacrificed. Under sacrifice by minimal remaining work, we compared the sample paths of the completed work processes for finite and infinite buffers to bound the number of sacrifices by the number of jumps the infinite buffer completed work makes from above the finite buffer size.

The bound is valid regardless of the nature of the packet arrival process or the packet length distribution. However, numerical computations require additional assumptions. To illustrate our results, we assumed Poisson arrivals and independent, identically distributed packet lengths, making the ATM switch into a processor sharing M/G/1 queue. We then drew upon renewal theory and the theory of symmetric queues to calculate our bound on packet losses.

Sacrifice by minimum remaining work has a number of drawbacks. It cannot be implemented if the VCQ does not know the packet's length. Moreover, the motivation for this algorithm (when all packets are of maximum length) was to free up as much space as possible in the VCQ. This strategy appears to be too myopic, sacrificing the packet that will complete next seems to result in too many sacrifices. The evidence is the smaller packet loss probabilities corresponding to sacrifice of the packet that overflows the VCQ and sacrifice by most remaining work. Our simulation results suggest that sacrifice by minimal remaining work is the worst sacrifice algorithm from the point of view of packet losses and lead us to ask whether our analytic model provides an upper bound on loss probabilities for all sacrifice algorithms. The issue is important because if the bound is general, then the model provides the means to do worst case engineering.

Our results and conjectures depend on the Poisson packet arrival assumption. Further work will require better characterizations of host and LAN traffic. A recent analysis of ethernet traffic [12] shows the aggregate traffic on a local area network can display similarity on time scales of many different orders of magnitude. Queueing theory has not yet assimilated such traffic processes, however, much less the way passage through a network alters such traffic streams. General Markovian Arrival Processes [14] are also candidates to model the traffic offered to an ATM network. However, the processor sharing, general service time, single-server queue has not been analyzed for such arrival processes. Therefore, more realistic arrival processes would have to be analyzed by simulation. The Poisson process considered here is a first approximation that suggests directions for future efforts.

REFERENCES


Donald E. Smith received the B.A. degree in mathematics from Princeton University and the Ph.D. degree in mathematics from the University of Chicago.

He is a Member of the Technical Staff at Bellcore, Red Bank, NJ, in the Robust Networks Design, Analysis, and Assurance Department. He has worked on congestion control in Common Channel Signaling and ATM, ATM performance, switch and signaling transfer point analysis, and operator services. He currently works on broadband performance analysis.

Dr. Smith was awarded the Leonard G. Abraham Prize Paper Award for his paper, "Effects of feedback delay on the performance of the transfer-controlled procedure in controlling ccs network overloads," published in IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS in April, 1994.

H. Jonathan Chao (S'84-M'85-SM'95) for a photograph and biography, please see p. 659 of this issue.