ANALYSIS OF THE DISTRIBUTION OF CONSECUTIVE CELL LOSSES IN AN ATM SWITCH USING STOCHASTIC ACTIVITY NETWORKS *

Latha Kant and William H. Sanders

Center for Reliable and High-Performance Computing
University of Illinois at Urbana-Champaign
Urbana, IL 61801 USA

lkant@crhc.uiuc.edu and whs@crhc.uiuc.edu
(217) 333-0345

ABSTRACT

Advancements in fast packet switching technology have made possible ATM-based B-ISDNs and integration of diverse telecommunication services. High throughput requirements and diverse services place stringent quality of service (QoS) demands on the associated switches. The often computed average cell-loss probability (clp) is an interesting but not a sufficient measure, since it is averaged over both time and all switch inputs. There exist many applications that are sensitive to the *pattern* of cell loss, where despite a low average clp, consecutive cell loss implies insufficient QoS. Further, the cell loss pattern as seen by the switch and a specific port can differ even if the average clp for the two is the same. It is therefore important to distinguish between the loss behavior at the switch and at a port, especially when examining QoS as perceived by the users of a specific switch port. In this paper, we use stochastic activity networks (SANs) to analyze the distribution of consecutive cell loss, both with respect to the ATM switch as well as a specific port. To do this, we use *UltraSAN*, a SAN-based performance modeling and analysis tool to construct and solve the detailed Markov processes associated with the switch and a bursty workload. Our results provide useful information, both about the usefulness of SANs and *UltraSAN*, as well as the importance of sophisticated measures, such as the distribution of consecutive cell losses, when evaluating ATM switch designs.

Keywords: Asynchronous transfer mode, Broadband ISDN, Consecutive cell-loss probability, Fast packet switch, Stochastic activity networks, Stochastic Petri nets.

*This work was supported, in part, by a grant from Bell Communications Research.

I Introduction

Research in ATM switching has resulted in a variety of fast packet switch (FPS) designs. The high switching and routing speeds of the FPSs have provided a strong impetus in the proposed deployment of ATM-based B-ISDNs, which facilitate integration of diverse services like voice, data, and multimedia on a single transport network. The diversity of input traffic implies widely differing quality of service (QoS) demands from the underlying network. Real-time services like voice and video require very low delays but are more resilient to loss, while data applications demand the exact reverse.

The associated switches therefore need to provide high throughput together with low latency and cell loss. The approaches to designing high-speed switching fabrics thus involve a large degree of parallelism with routing performed at the hardware level. The vast amount of research in FPS architectures in the past decade has led to a variety of designs (e.g., [1, 22, 24, 31]). Further, the advancements in fiber optic technology have also motivated photonic fast packet switching [11]. The choice of a particular FPS architecture, however, is not simple, since there exist tradeoffs in the cost and performance of different FPS designs [8, 34].

In this paper, we consider switches with the following characteristics: (a) a fully interconnected structure to avoid internal blocking, (b) separate output ports to avoid HOL blocking [12] and render exact analysis of larger switch dimensions feasible,[1] and (c) synchronous input/output ports to avoid internal speed-ups, and, analyze performance of a generic FPS with the above features.

The measures employed to assess switch performance are extremely important. Considerable work in the past has been devoted to analyzing FPSs in terms of the average cell-loss probability (clp). This measure, though important, fails to shed light on the actual QoS perceived by diverse applications. This is because many real-time applications (e.g., recursively encoded video streams) are extremely sensitive to the *manner* or pattern of cell loss, with *consecutive* cell loss being very detrimental. In fact, even if the average clp requirement is met, satisfactory QoS may *not* be achieved if the losses are clustered for these applications.

While related work on loss behavior exists, no prior work has analyzed the distribution

[1] Though shared output buffers possess the potential to provide even better cell-loss probability, their analysis soon becomes intractable with bursty workloads typical in B-ISDNs [3, 9, 25].

of consecutive cell loss for a specific port in ATM switches. In particular, (a) [4, 5] address the issue of loss bunching with Poisson arrivals, (b) [6, 21, 23] provide general discussions of heavy load durations and cell loss compensation via simulation, (c) [27] presents a simulation study of alternative strategies to reduce successive cell loss in packet multiplexers, (d) [7] analyzes packet loss process in continuous-time systems with exponential service times and in discrete-time (slotted) systems with i.i.d. inputs, and (e) [2] provides an analysis of message loss process in an M/M/1/K queueing system. Also, in this context, [29] addresses cell-loss correlation but does not analyze the distribution of consecutive cell loss, while [30] discusses the loss and output process for a discrete-time queueing system but, again, does not address the distribution of consecutive cell loss for a specific port in an FPS, as also [15], who propose a packet drop strategy to minimize the average packet gap in ATM sessions.

This work considers the significance of loss clustering in ATM switches and analyzes the distribution of consecutive cell loss by computing the fraction of loss bursts (defined in Section II, Subsection C) of length m, where m is varied. (A loss burst is an uninterrupted period of cell loss.) We compute this measure *both* with respect to the switch and a specific input port (also referred to as a tagged port), since it differs considerably even with homogeneous traffic. The average clp, however, is the same for the two. As our results indicate, a high value for loss bursts at the switch does not necessarily imply poor switch performance (poor QoS), since the corresponding values for a specific port may be significantly lower, implying sufficient QoS from the perspective of the users of the particular switch port. We employ a bursty workload and use both homogeneous and heterogeneous switch inputs and vary the burst parameters of the workload over a wide range to capture the diverse application mix in a B-ISDN environment.

Evaluation of an FPS can be done by the construction and solution of a stochastic process associated with the switch and workload. Since the sizes of these detailed processes are large (on the order of tens to hundreds of thousands of states), a hand construction and solution of such processes is a formidable task. The problem is compounded further if alternative switch designs have to be evaluated. In this paper, we provide a methodology that can be used to evaluate alternative FPS designs. We use stochastic activity networks (SANs) [18, 19], a variant of stochastic Petri nets, to evaluate the FPS. The SAN formalism provides for a compact yet powerful way of modeling and analyzing alternative switch designs. It enables the automatic generation of the stochastic processes associated with the switch and workload, which can be solved numerically. We can therefore model a "class"

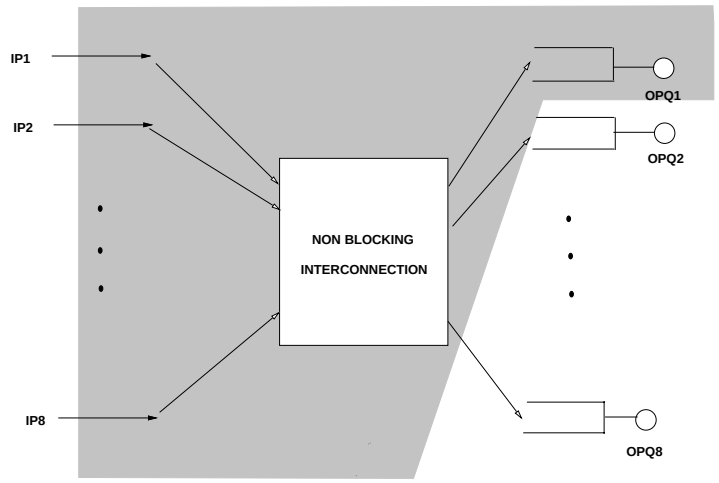

Figure 1: Switch model

of architectures and analyze specific FPS designs within the framework of SANs.

We use *UltraSAN* [28], a SAN-based performance modeling and analysis tool to automatically generate and solve the underlying Markov processes of the associated switch and workload. We are able to construct and solve detailed representations of the FPS and workload models relatively easily on a typical workstation. This permits much more accurate analysis than would have been possible if the stochastic process were to be constructed by hand. The results provide significant insights into the performance of a given FPS design, as well as indicating the power and usefulness of SANs for modeling and evaluating telecommunication switch architectures.

The remainder of the paper is organized as follows. Section II formulates the problem by characterizing the switch and workload models precisely and discusses the performance variables employed. Section III describes the details of SAN models of the FPS and workload, and it discusses model solution. Section IV discusses the results, and Section V concludes the paper.

II Problem formulation

A Switch model

The switch considered for analysis (see Figure 1) is characterized by a fully interconnected structure, with synchronous input/output ports and separate output buffering.

The behavior of the FPS is modeled by examining the precise sequence of events that

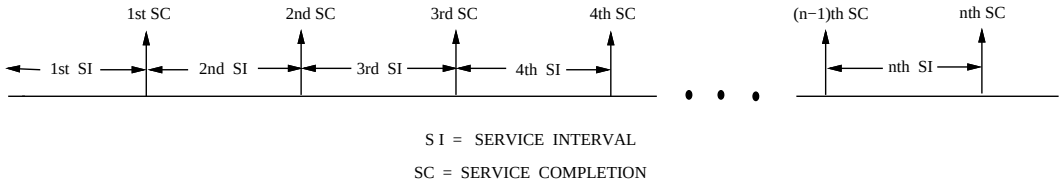

Figure 2: Timing diagram

occurs upon cell service. The synchronous input/output ports together with deterministic service times permit us to view the operation of the switch in a time-slotted fashion as shown in Figure 2. A slot corresponds to the time taken to serve an ATM cell and is denoted by SI (service interval) in Figure 2. The end of a slot is marked by SC (service completion).

To enumerate the sequence of events at each port formally, consider an arbitrary time point n corresponding to a service completion. The sequence is:

1. The cell in service at each output queue during the nth SI is removed.
2. High-speed discarding decisions are performed, and the newly arriving cells are accommodated at their respective output queues.
3. The $(n+1)$th SI begins, representing the service times for the cells at the head of the output queues.

In the above, we assume that cell service is non pre-emptive and the high-speed discarding decisions are performed in negligible time, both of which are reasonable assumptions for an FPS.

Proceeding with the switch model, since the output queues are not shared (i.e., separate queue at each output port), we may restrict our attention to a single output port. Denoting the state of the system in the nth SI by X_n, we have, $X_n = (A_n, D_n, Q_n)$, where

- A_n denotes the number of arrivals at the beginning of the nth SI, i.e.,

$$A_n = f(\textit{workload state in nth SI})$$

- D_n denotes the number of cells discarded in the nth SI. This depends both on the discarding mechanism (if any) employed within the interconnection fabric (e.g., loss at the shift register unit (SRU) of the Gauss switch [32] or at the knockout concentrator in a knockout switch [33]) and loss due to lack of space at the output queue Q, i.e.,

$$D_n = g(\textit{interconnection discarding policy}) + \textit{loss at } Q$$

- Q_n denotes the state of the output queue at the nth SI and depends on the state of the output queue after the $(n-1)$th SC and discarding decisions both within the interconnection fabric and the output queue, i.e.,

$$Q_n = h(Q_{n-1} + outcome\ of\ discarding\ decisions)$$

where $f(\cdot)$, $g(\cdot)$, and $h(\cdot)$ are appropriately defined functions for the selected workload and switch architecture. Subsection C discusses the details of each of the above functions for the FPS in Figure 1, together with the performance variables.

B Workload model

A number of bursty traffic models exist in the literature. They range from among a variety of Markovian models as described in [10, 13, 14, 17, 26], and the references they contain, to the more recent study based on self-similarity [16] and long-range traffic models [20]. The choice of an "appropriate" model to characterize the diversity of traffic in B-ISDNs, however, is very much an arguable matter, due to the difficulty of accurately specifying the traffic mix to an FPS. Further, most of the good traffic models are very application specific. Since our aim is to study loss behavior of the FPS with correlated traffic in general, and not focus on any specific application, we employ a two-state MMPP as in [14, 17]. However, to capture the *effect* of multiplexing diverse applications, we vary the burst parameters over a wide range and consider both homogeneous and heterogeneous inputs.

The switch inputs therefore switch between idle and active states representing spurts of silent and active periods, respectively. When active, a cell is emitted with probability q, and no cell is emitted when it is silent (idle). The transition matrix for this workload is given by

$$\begin{bmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{bmatrix},$$

where subscripts 0 and 1 stand for the idle and active states, respectively. The holding time in each of the two states is geometrically distributed.

To characterize the bursty workload, we use the activity fraction (AF) and burst size (BS) as in [14, 17], together with q and ON times, each of which are defined below.

First, let $\underline{\pi}$ denote the steady-state probability distribution vector, i.e.,

$$\pi_0 = \frac{p_{10}}{(p_{01} + p_{10})} \ \ and \ \ \pi_1 = \frac{p_{01}}{(p_{01} + p_{10})} \tag{1}$$

Next, the burst parameters are defined as follows:

- Average time spent in active period denoted by E_{act}, and also referred to as ON time, defined by

$$E_{act} = \frac{1}{p_{10}} \tag{2}$$

- Average burst size (BS) defined by

$$BS = E_{act} \times q \tag{3}$$

- Activity fraction (AF), the fraction of time the workload is active, defined by

$$AF = \pi_1 \tag{4}$$

The average offered load ρ is given by

$$\rho = \pi_1 \times q = \frac{p_{01}}{(p_{01} + p_{10})} \times q \tag{5}$$

Thus, the burst parameters AF and BS together with q can be varied to denote the traffic "peakedness" and long spurts of congestion despite a fixed average load. Equations 2, 3, and 4 therefore help in selecting appropriate Markov chain parameters while analyzing switch performance with correlated inputs.

C State descriptor and performance measures

For an NxN switch with characteristics as in Figure 1 and a bursty workload model as discussed in Subsection B, the functions $f(\cdot)$, $g(\cdot)$, and $h(\cdot)$ for A_n, D_n, and Q_n may be represented as follows.

$\underline{A_n:}$

$$A_n = C_r^{nai} \times (P_{cell})^r \times (1 - P_{cell})^{nai-r} \qquad 0 \leq r \leq nai \tag{6}$$

where nai is the number of active inputs at the beginning of the nth SI, and P_{cell} denotes the probability of a cell emission when the workload is active (i.e., q). C_y^x is the function x *choose* y. To calculate nai, we have:

$$nai = a_0 + a_1 \tag{7}$$

where a_0 represents inputs that were idle in the $(n-1)$th SI but become active in the nth SI, and a_1 represents inputs that remain active in the nth SI. The expressions for a_0 and a_1 are:

$$a_0 = C_k^{ni} \times (P_{i-a})^k \times (1 - P_{i-a})^{ni-k} \qquad 0 \le k \le ni \tag{8}$$

$$a_1 = C_l^{N-ni} \times (P_{a-a})^l \times (1 - P_{a-a})^{N-ni-l} \qquad 0 \le l \le (N - ni) \tag{9}$$

where ni represents the number of idle sources in the $(n-1)$th SI, P_{i-a} denotes the probability that the workload transitions from idle to active states (p_{01}), and P_{a-a} represents the probability that it remains active (p_{11}) in the nth SI.

$\underline{D_n:}$ Since the FPS is fully interconnected with no discarding within the interconnection fabric, the expression for D_n becomes:

$$D_n = A_n - (Q_{max} - (Q_{n-1} - 1)) \qquad A_n > (Q_{max} - (Q_{n-1} - 1)) \tag{10}$$

$$D_n = 0 \qquad otherwise \tag{11}$$

$\underline{Q_n:}$ The number of cells at the output queue depends on the number enqueued in the $(n-1)$th SI less one (to denote the cell that completed its service) and the number of arriving and discarded cells.

$$Q_n = Q_{n-1} - 1 + A_n - D_n \tag{12}$$

This completes the state descriptor X_n.

Next, since the system is viewed at fixed time slots and state X_{n+1} depends only on X_n, the process $\{ X_n \mid n \in I\!N \}$ is a discrete-time Markov chain (DTMC).

Regarding the performance variables, we proceed as follows. To calculate clp, define clp_n, the cell-loss probability at time n, as the fraction of number of arriving cells discarded. We have

$$clp_n = \frac{E[D_n]}{E[A_n]} \tag{13}$$

where $E[\cdot]$ denotes the expectation. The expectations are obtained by solving for the state occupancy probabilities $\pi_{i,j,k}^{(n)}$, where $\pi_{i,j,k}^{(n)}$ represents the probability that the system is in state $X_n = (i, j, k)$ at time n. As the DTMC under consideration is finite state, irreducible, aperiodic, and time homogeneous, the limiting probabilities exist and are independent of the initial state. Thus, $\pi_{i,j,k} = \lim_{n\to\infty} \pi_{i,j,k}^{(n)}$, and equation (13) becomes

$$clp = \frac{\sum_{i,j,k} j\pi_{i,j,k}}{\rho} \tag{14}$$

since the denominator in (1) is the traffic intensity ρ. Equation (14) gives the clp as seen at the switch.

To calculate the loss seen at a particular port, a refinement of the state space is needed to tag the distinguished port. Using superscript t to denote the tagged port, the state of the system in the nth SI has A_n^t and D_n^t in addition to A_n, D_n, and M_n in its state description and is represented by X_n^t. Similar to equation (13), we have

$$clp_n^t = \frac{D_n^t}{A_n^t} \tag{15}$$

Since the limiting probabilities exist, the limiting clp as seen by the tagged port, clp^t, is calculated similar to equation (14) with the denominator being ρ^t instead and the numerator being only those cells discarded from the tagged stream.

Next, to study the distribution of consecutive cell loss, we proceed as follows. First, we define a "loss burst" of length m as the loss of at least one cell in m consecutive slots. Next, we compute loss bursts of length m, where m is varied. To do this, define LB_n to be a random variable denoting the length of a loss burst observed at time n, and $P(LB_n = m)$ as the probability that at least one cell loss has occurred in m preceding consecutive slots when the system is observed at time n. Thus,

$$\begin{aligned}
P(LB_n = m) &= P(D_n > 0, D_{n-1} > 0, \cdots, D_{n-(m-1)} > 0, D_{n-m} = 0) \\
&= P((A_n = i_0, D_n = j_0, Q_n = qmax), (A_{n-1} = i_1, D_{n-1} = j_1, Q_{n-1} = qmax), \\
&\quad \cdots, (A_{n-(m-1)} = i_{m-1}, D_{n-(m-1)} = j_{m-1}, Q_{n-(m-1)} = qmax), \\
&\quad (A_{n-m} = i_m, D_{n-m} = j_m, Q_{n-m} = k)) \\
&= P(X_n = (i_0, j_0, q_{max}), X_{n-1} = (i_1, j_1, q_{max}), \cdots \\
&\quad \cdots, X_{n-(m-1)} = (i_{m-1}, j_{m-1}, q_{max}), X_{n-m} = (i_m, j_m, k))
\end{aligned} \tag{16}$$

where $i_0, i_1, \cdots, i_{m-1} \geq 2$, $i_m \geq 0$, $1 \leq j_0 \leq (i_0 - 1)$, $1 \leq j_1 \leq (i_1 - 1)$, $\cdots$, $1 \leq j_{m-1} \leq (i_{m-1} - 1)$, $j_m = 0$, and $k \leq qmax$ with $qmax$ being the capacity of the output queue.

Since the models of the switch considered are ergodic, the limit of $P(LB_n = m)$ as $n \to \infty$ exists and is the probability that m consecutive slots have incurred at least one cell loss when the system is observed at some random time in steady state.

Buffer size	*Number of states*
30	9704
60	19424
90	29144
120	38864
150	48584

Table 1: Variation of state-space size with buffer capacity with a tagged port

Next, we can define the fraction of loss bursts of a particular length r, FLB_r, as

$$FLB_r = \frac{P(LB = r)}{\sum_{p=1}^{\infty} P(LB = p)} \quad r \geq 1 \tag{17}$$

Finally, to study the distribution of consecutive cell loss, we compute the fraction of loss bursts of different lengths by varying r. The results obtained via these calculations are presented in Section IV. The above measure definition is with respect to the switch. The definition with respect to a tagged port is similar to that for the switch except for the following differences: (a) we use the augmented state descriptor X_n^t, and (b) the random variable D_n^t is either zero or one, since either no cells or only one cell arrives from the tagged port during a slot. Thus, the definition of a loss burst of length m for a tagged source is the loss of one cell in each of m consecutive slots.

III SAN representation and model solution

Having formulated the problem, the next step is to construct the appropriate Markov chains to compute the distribution of consecutive cell losses. Since these Markov processes are large (on the order of tens of thousands of states, see Table 1), we use *UltraSAN* [28], a SAN-based performance modeling and analysis tool, to generate the needed Markov processes, rather than construct them by hand. The SAN can precisely represent the FPS architecture, workload, and performance variables specified abstractly in the previous section. Figure 3 gives a SAN representation of the FPS and bursty workload model. While space does not permit a detailed description of the SAN formalism, we introduce the various SAN components as we describe the model. For further information on SANs, see [18, 19].

SANs consist of four primitives: *places, activities, input gates,* and *output gates.* Places

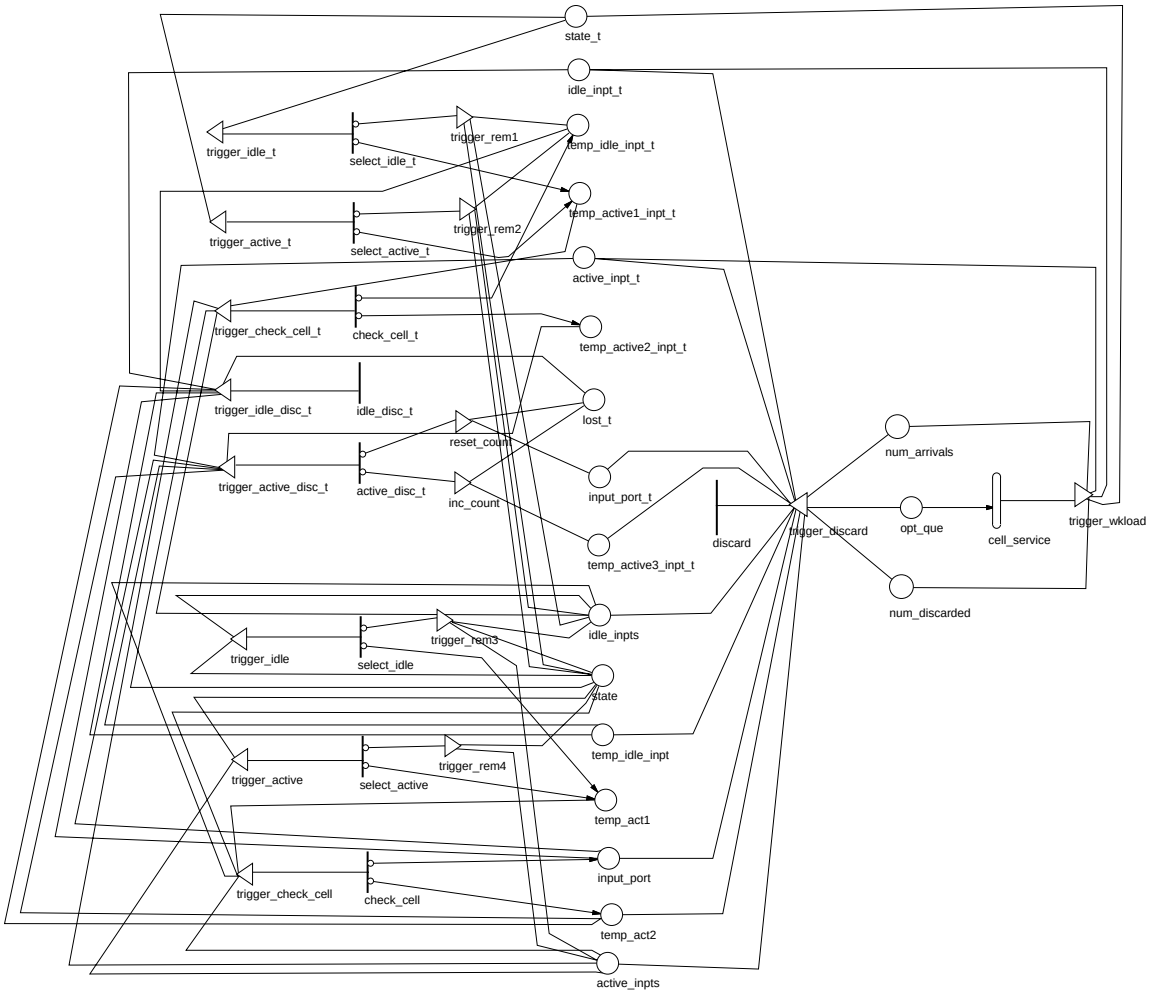

Figure 3: SAN model of switch with bursty workload and a tagged input

are represented by circles (e.g., *opt_que, num_arrivals* in Figure 3). They are used to represent the "state" of the system and may contain tokens. Activities represent actions of the modeled system and are of two types: (a) timed and (b) instantaneous. Timed activities are represented by a hollow vertical bar and denote actions in the system that take time to complete. For example, *cell_service* in Figure 3 denotes a timed activity to represent the service time for an ATM cell. Instantaneous activities are represented by a solid vertical line and denote actions that complete in a negligible amount of time relative to the other activities of the modeled system. In Figure 3, *discard* is an instantaneous activity that models the high-speed routing and discarding decisions performed by the FPS.

Activities can have *case probabilities* associated with them. Case probabilities, repre-

sented by tiny circles on the right side of an activity, represent uncertainty associated with the completion of that activity, with each case denoting a possible outcome. In Figure 3, *select_idle* with its two case probabilities represent the probabilities p_{00} and p_{01}. Gates are represented by triangles and are of two kinds, *input* and *output*. Input gates are used to enable activities, while both input and output gates help change the state of the system upon activity completion. In Figure 3, *trigger_idle* is an example of an input gate while *trigger_wkload* denotes an output gate.

The functioning of the SAN model is as follows. All places named with an "*_t*" represent a tagged input. The portion of the SAN to the left of activity *discard* represents the workload model, and that to the right, the FPS. At the beginning of every SI, the number of tokens in *input_state_t* is checked. Depending on the number of tokens in it (one token implies that the tagged input was idle in the $(n-1)$th SI, and two tokens imply that it was active), input gates *trigger_idle_t* or *trigger_active_t* enable activities *select_idle_t* or *select_active_t*, respectively. These activities along with their four case probabilities implement the four transition probabilities of the bursty workload. If an input either remains or becomes idle, no cell is emitted, and a token is placed in *temp_idle_t*. On the other hand, if it becomes or remains active, a token is placed in *temp_active1_t*. Input gate *trigger_check_cell_t* then enables *check_cell_t*, which decides upon the presence or absence of a cell via its two case probabilities.

Input gate *trigger_check_cell_t* also triggers the workload for the remaining $(N-1)$ inputs, by placing either one or two tokens in *state*. It places one token in *state* if at least one of the remaining $(N-1)$ inputs was idle, else it places two tokens. On a similar note, if the tagged input had transitioned to the idle state, then output gates *trigger_rem1* and *trigger_rem2* perform similar checks to place the appropriate number of tokens in *state* to trigger the remaining $(N-1)$ inputs.

The functioning of the gates *trigger_idle, trigger_active,* and *check_cell* along with their activities *select_idle, select_active,* and *check_cell* is similar to that described in the case of the tagged input. Essentially, these gates and activities check the remaining $N-1$ inputs for the presence or absence of a cell. After the $(N-1)$ inputs have all been examined, then, depending on whether the tagged input has a cell arrival or not, activities *active_disc_t* or *idle_disc_t*, respectively, are enabled. If it has an arrival, the two case probabilities of *active_disc_t* decide whether or not the tagged cell is discarded. If it is discarded, output gate *inc_count* increments the count in *lost_t*, which represents the consecutive cell loss for

the tagged source. (A maximum count on *lost_t* is imposed to maintain finiteness of the generated state space.) Otherwise, *lost_t* is set to zero, and a token is placed in *inpport_t* by output gate *reset_count*. If no cell was emitted by the tagged input, then activity *idle_disc_t* is enabled, which resets *lost_t*.

Input gate *trigger_discard* then enables *discard*, which together implement the high-speed routing and discarding decisions performed by the FPS. The total number of cell arrivals over the current SI is stored in *num_arrivals*. If this number exceeds queue capacity at place *opt_que*, the excess cells are discarded, the count in *num_discarded* is incremented to reflect this loss, and the number of tokens in *opt_que* is updated to its maximum queue size. If not, *num_discarded* loss is set to zero, and the count in *opt_que* is incremented accordingly.

The timed activity *cell_service* denotes the fixed time to serve an ATM cell. Upon its completion, a cell is removed from the output queue, and the output gate *trigger_wkload* resets *num_arrivals* and *num_discarded* and triggers the workload by placing either one or two tokens in *state_t*, depending, respectively, on whether or not the tagged input was idle.

Due to space limitations, we have only provided the specifications for one input gate, output gate, and case probability in Tables 2, 3, and 4, respectively. In these tables, $MARK$ and $GLOBAL\_S$ are SAN keywords. $MARK$ followed by a place name in parentheses represents the marking (number of tokens) in that place. $GLOBAL\_S$ is used to declare global variables of type short. (A similar variable for doubles also exists.) Global variables in a SAN allow for model parameterization.

Once the SAN models of the FPS and workload are specified, the next step is the construction and solution of the underlying Markov process. This is done using *UltraSAN*, which automates the generation of the underlying continuous-time Markov process (CTMP). Note, however, that the switch and workload models formulated in Section II were represented by a discrete-time Markov chain (DTMC). But, it can easily be shown that the steady-state solution of a DTMC with a single timed (deterministic) activity is equivalent to that of the CTMP with the deterministic activity replaced by an exponential activity. In particular, if the time associated with the single deterministic activity is t, with the transition probability matrix for the DTMC denoted by $\mathbf{P}$ and the transition rate matrix for the CTMC by $\mathbf{Q}$, we have $\mathbf{P} = \mathbf{Q} \cdot t + \mathbf{I}$, establishing the equivalence between the two.

This equivalence allows us to use *UltraSAN* to generate the associated Markov process. The state-space size for the process depends on the switch size, buffer size, the maximum

Gate	*Definition*
trigger_discard	Predicate *MARK(input_port_t) ==1 || MARK(temp_active3_inpt_t) ==1* *|| MARK(idle_inpt_t) ==1 &&* *(MARK(input_port)+MARK(temp_act2)+MARK(temp_idle_inpt) ==* *GLOBAL_S(NTINPTS))*
	Function *MARK(num_arrivals) = MARK(input_port)+MARK(input_port_t);* *MARK(idle_inpts) = MARK(temp_idle_inpt);* *MARK(active_inpts) = MARK(temp_act2)+MARK(input_port);* *if((GLOBAL_S(QMAX)−MARK(opt_que)) >= MARK(num_arrivals)) {* *MARK(num_discarded) = 0;* *MARK(opt_que) + = MARK(num_arrivals);* *}* *else {* *MARK(num_discarded) =* *MARK(num_arrivals) − (GLOBAL_S(QMAX)−MARK(opt_que));* *MARK(opt_que) = GLOBAL_S(QMAX);* *}* *if(MARK(input_port_t) > 0 || MARK(temp_active3_inpt_t) > 0) {* *MARK(active_inpt_t) = 1;* *MARK(input_port_t) = 0;* *MARK(temp_active3_inpt_t) = 0;* *}* *else* *MARK(active_inpt_t) = 0;;* *MARK(input_port) = 0;* *MARK(temp_act2) = 0;* *if(MARK(opt_que) == 0)* *MARK(opt_que) + +;*

Table 2: Enabling predicate and function for input gate *trigger_discard*

Gate	*Definition*
trigger_wkload	*MARK(num_arrivals) = 0;* *MARK(num_discarded) = 0;* *if(MARK(idle_inpt_t) > 0) {* *MARK(state_t) = 1;* *MARK(idle_inpt_t) = 0;* *}* *else {* *MARK(state_t) = 2;* *MARK(active_inpt_t) = 0;* *}*

Table 3: Gate function for output gate *trigger_wkload*

Activity	*Case*	*Probability*
active_disc_t	1	*double z = 0.0;* *if((MARK(temp_act2)+1)>GLOBAL_S(QMAX)−MARK(op_que)) {* *z=((double)((MARK(temp_act2)+1)−* *(GLOBAL_S(QMAX)-MARK(opt_que)))) /* *((double)(MARK(temp_act2)+1)); }* *else {* *z = 0.0; }* *return(1.0−z);*
	2	*double z = 0.0;* *if((MARK(temp_act2)+1)>GLOBAL_S(QMAX)−MARK(opt_que)) {* *z=((double)((MARK(temp_act2)+1)−* *(GLOBAL_S(QMAX)-MARK(opt_que)))) /* *((double)(MARK(temp_act2)+1)); }* *else {* *z = 0.0; }* *return(z);*

Table 4: Case probability definitions for activity *active_dist_t*

consecutive loss length desired, and whether or not a particular input is tagged. Table 1 gives the state-space size as a function of the buffer capacity for an 8x8 switch with a tagged port and a maximum consecutive loss length of 5.

After the states are generated, the steady-state probability distribution vector is obtained numerically using successive over-relaxation (SOR). A stopping criterion of 10^{-9} is used, i.e., the SOR algorithm is stopped when the difference between the most recent and previous iteration is less than 10^{-9}. This is used to determine the clp and the distribution of the consecutive cell loss. The results obtained in this manner are described below.

IV Discussion of results

We present our results in two subsections. Subsection A discusses performance with homogeneous traffic, and Subsection B discusses behavior with heterogeneous traffic, both for an 8x8 switch unit with uniform loading across the output ports. Since loss behavior is very significant, we focus on the distribution of consecutive cell loss, but we present some results on the average clp as well, wherever pertinent, to maintain brevity. While studying distribution of consecutive cell loss, we look at loss bursts and present results for the fraction of the number of occurrences of loss bursts of length m, with m a parameter. For example, a probability 0.2 for a consecutive loss length 4 means that when the system is observed at some random time in steady state, 20% of the losses observed have occurred over 4 successive slots.

The choice of realistic values for the burst parameters of the workload is difficult. This is because the AFs for a given average load may range from very low (peaked) to large (smooth) values based on the relative mix of the number and type of applications and the port speed. The work by [13] also addresses a similar problem of workload characterization in LANs. Their results indicate the smoothening effect due to aggregation at high loads on the overall LAN traffic parameters, with the bursty nature prevailing more strongly at low loads.

While the issue of "appropriate" parameters with bursty workload is very much debatable, it is apparent that these parameters will vary significantly depending on the relative mix of the various traffic types. Therefore, to capture such a mix, we vary the burst parameters over a wide range in our analysis, and we use both homogeneous as well as heterogeneous inputs. Regarding the cell drop mechanism, observe that both the consecutive

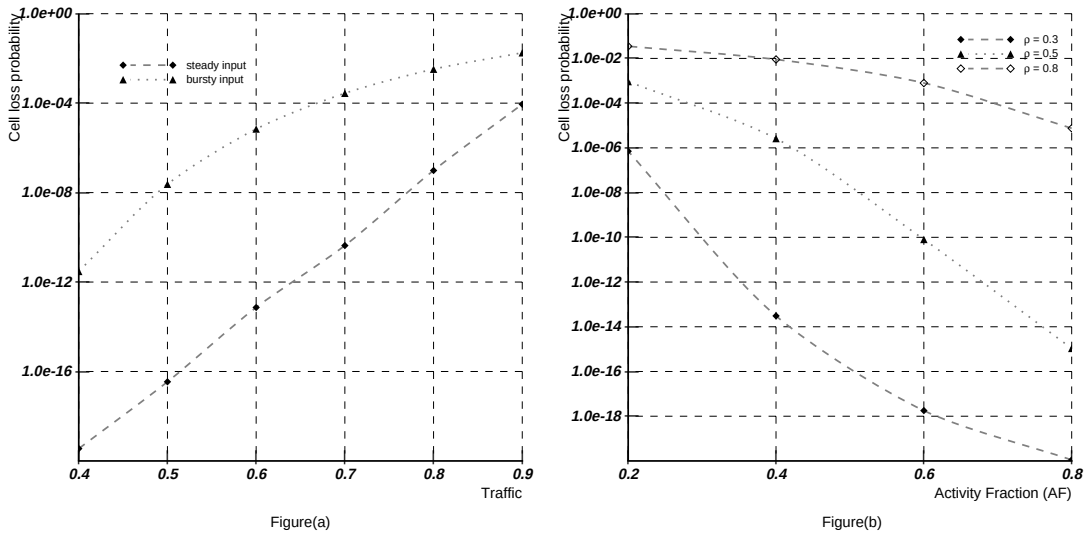

Figure 4: Effect of bursty traffic (Figure(a)) and AF (Figure(b)) on clp

cell loss behavior and its effects in terms of the reconstructed signal will be influenced by the particular cell drop mechanism. However, since the focus of this work in not on comparing the various cell drop mechanisms (e.g., push-out, head, tail, random) or, coding techniques (which also influences signal recovery), we use a tail dropping mechanism that can be implemented fairly easily and without any overheads in fast packet switches. Further, we do not assume any particular encoding structure. Our results thus provide useful insights into the trends that occur and the FPS robustness with varying burstiness and mixes of input traffic, rather than absolute measures of performance.

A Performance with homogeneous inputs

In this subsection, the burst parameters of all the inputs are the same. The output queue size is set to 30 except in Figures 7(a) and (b).

Effect of traffic bursts and AF on clp Figure 4(a) gives the clp vs. the average offered load ρ for both steady and bursty traffic. With bursty traffic, AF = 0.5 and BS = 8. q is varied in the case of bursty traffic to vary ρ. Figure 4(a) is included to illustrate the significance of correlated input vs. i.i.d. Bernoulli (steady) input. Observe that for the same average load, performance with bursty traffic is much worse than with steady inputs. This underscores the importance of analysis with bursty correlated inputs, as opposed to

the often used i.i.d. Bernoulli traffic.

Figure 4(b) illustrates the effect of AF on clp. The AF is varied from 0.2 to 0.8 for three average loads with BS fixed at 8 to provide the effect of multiplexing different types and numbers of sources. The lower AFs reflect the multiplexing of a small number of large bandwidth applications, for example, motion video streams on a DS3 port, while the higher AFs reflect the reverse, for example, multiplexing many low-quality video conference applications at 384 kbps on a DS3 port.

Observe that for a fixed average load, the clp varies significantly as a function of the AF, illustrating the importance of differing AFs despite the same average load. Another important point illustrated by these curves is that low AFs (peaked traffic) by themselves do not imply poor performance. (Notice the orders of magnitude variation in clp with AF = 0.2 and $\rho = 0.3$, 0.5, and 0.8.) Further, observe that while a combination of high loads and low AFs can be very detrimental as expected, due to the high bandwidth demands at extremely bursty intervals, the other extreme is a low load with a large AF. The latter implies almost smooth traffic with very low utilization and hence produces very low clp, as expected. This suggests that efficient admission control and policing mechanisms that admit and police based on an AF-load pair rather than AF or load alone may help achieve good performance. In such a case, a larger swing may be permitted with low AFs (peaked traffic) and low loads. Finally, since homogeneous inputs are used in this section, the average clp for both the switch and a tagged port are the same.

Effect of AF on the distribution of consecutive cell loss Figures 5(a) and (b) illustrate the fraction of loss bursts of particular lengths for a tagged port and switch, respectively. The AFs are varied from 0.2 to 0.8 for a $\rho = 0.8$ and BS = 8.

The curves in Figures 5(a) and (b) illustrate several important points. First, they demonstrate the detrimental effects of peaked traffic (low AFs) on the loss process, since lower AFs produce significantly longer consecutive loss lengths for the same average load. Next, the slopes of curves with small AFs are much shallower than their higher AF counterparts, indicative of a rapid deterioration in consecutive losses. Specifically, as illustrated in Figure 5(a), observe that while consecutive loss lengths ≥ 5 with AFs ≥ 4 are small ($\leq 10^{-4}$), they become significant ($> 10^{-2}$) with AFs ≤ 2. This is reflected as a severe degradation in QoS and suggests the need for efficient traffic shaping and rate regulation mechanisms.

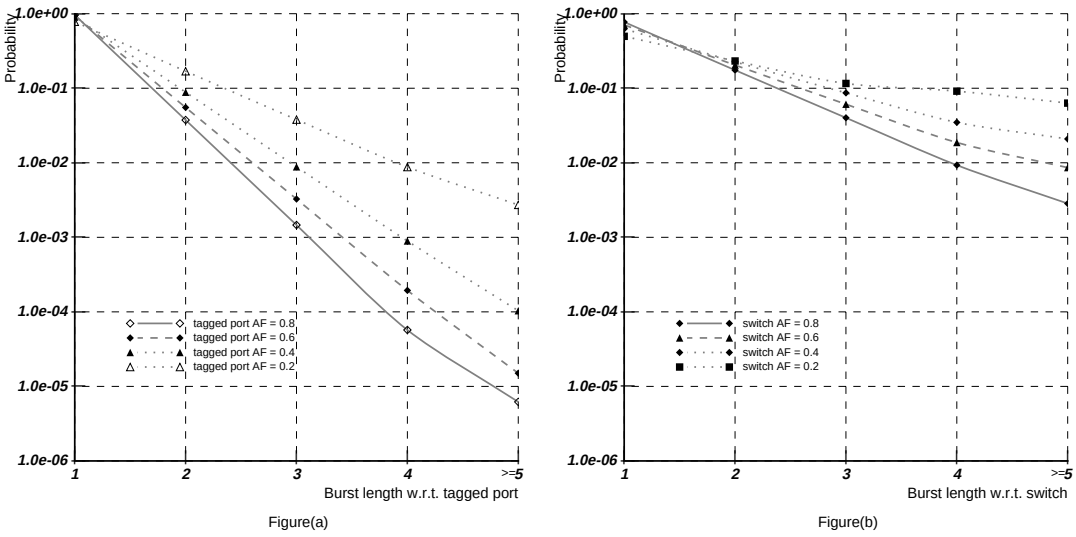

Figure 5: Effect of AF on the distribution of consecutive cell loss at a tagged port (Figure(a)) and at switch (Figure(b))

Finally, while the average clp for both the switch and a tagged port is the same with homogeneous inputs (Figure 4(b)), the fraction of loss bursts of particular lengths for the two differ significantly despite homogeneity. Figure 5(b) illustrates this. Observe the orders of magnitude difference between the switch and a tagged stream for loss lengths greater than 4 for a fixed AF and ρ. This reflects as a wide variation in the QoS as perceived by the switch and a tagged port. However, though the values for burst lengths for the switch are high (on the order of 10^{-1} and 10^{-2}), this does not reflect poorly on the FPS. This is because the values at the switch are those averaged over all the inputs. The values as seen at a tagged port, however, are more relevant, since this is what a specific switch stream sees. This information is helpful to switch designers and in admission control schemes. Thus, in the following discussions, we examine the fraction of loss bursts of particular lengths for a specific (tagged) switch port.

Effect of BS on the distribution of consecutive cell loss Figures 6(a) and (b) illustrate the effect of different lengths of congestion (BS varying from 8 to 128) with a fixed average load (0.8) and two different AFs (0.3 and 0.8, respectively). In Figure 6(a), this results in varying mean ON times from 24 to 384 cell times, while in Figure 6(b), it results in mean ON times varying from 64 to 1024 cell times, for the same average load.

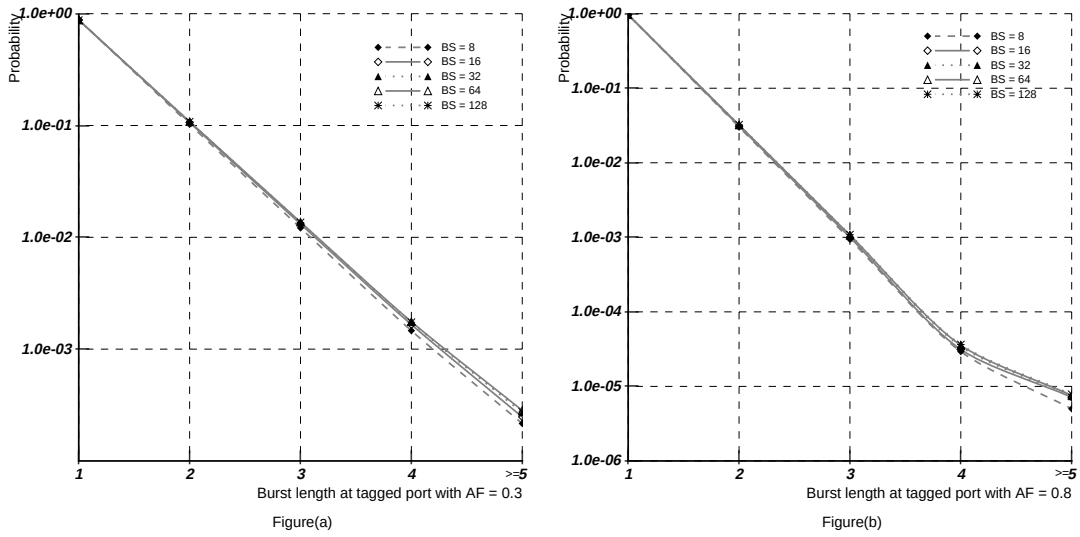

Figure 6: Effect of BS on the distribution of consecutive cell loss at a tagged port with AF = 0.3 (Figure(a)) and AF = 0.8 (Figure(b))

The clustering of the curves for varying ON times in both these figures indicates that the degradation in the fraction of loss bursts of given lengths is not very severe for the range of BSs examined and with ρ and AF fixed. (Contrast this with the degradation in Figure 5(a) where ρ and BS were fixed and AF was varied.) To gain a better insight, we therefore use two AFs, a large AF of 0.8 (Figure 6(a)) to represent fairly smooth traffic and a small AF of 0.3 (Figure 6(b)) representative of much burstier traffic.

Comparing Figures 6(a) and (b), we observe that with a lower AF the fraction of loss bursts ≥ 5 with all five burst sizes is much higher than the corresponding values with a larger AF. (Note that the scales in Figures 6(a) and (b) are different.) This is as expected since Figure 6(a) represents much burstier traffic. However, notice that with smoother traffic, the slopes with different BSs become less steep when the fraction of loss clusters become ≥ 5 indicating the onset of a saturation effect (Figure 6(b)). Observe also that the saturation becomes relatively more pronounced as the burst sizes increase. Such a saturation effect, however, is not the case with a lower AF (Figure 6(a)), which in turn implies poorer performance. Finally, note that the high values for the fraction of loss bursts ≤ 2 imply that most of the loss bursts are either isolated, or occur in pairs, which is indeed a good sign, especially for those applications that are sensitive to large loss clusters.

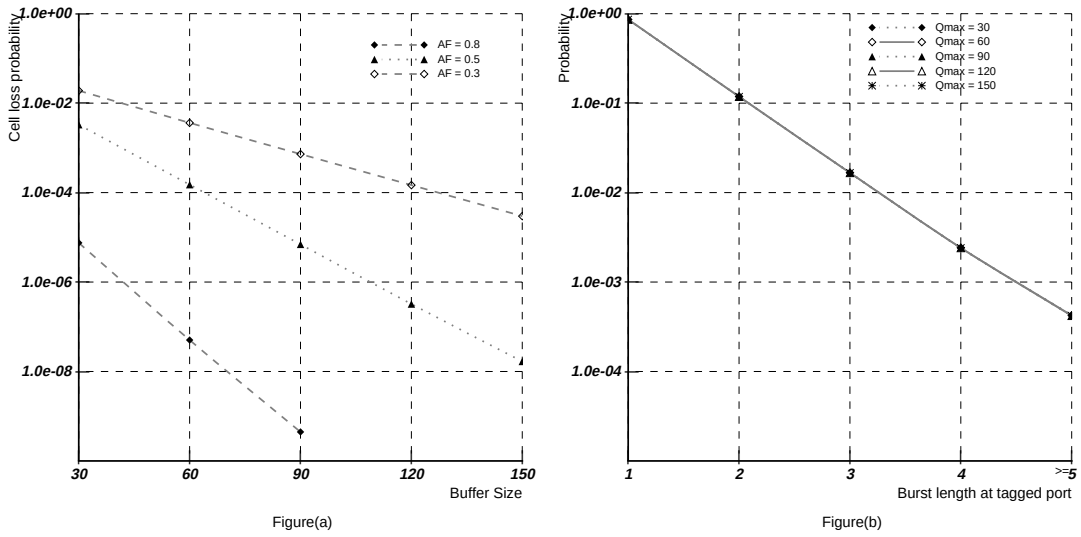

Figure 7: Effect of buffer size on the average clp (Figure(a)) and on the distribution of consecutive cell loss of a tagged port (Figure(b))

Effect of buffer size on the distribution of consecutive cell loss Figures 7(a) and (b) display the effect of buffer size on the average clp and the fraction of loss bursts of given lengths, respectively, for a tagged port, with $\rho = 0.8$ and BS $= 8$. In Figure 7(a), three AFs, a high (0.8), a medium (0.5), and a low (0.3), are used, while Figure 7(b) displays results for one AF (0.8).

While the average clp behavior improves with increasing buffer size (Figure 7(a)), observe that it plays no role on the fraction of loss bursts of particular lengths (Figure 7(b)). This is precisely due to the fact that large buffers imply that more cells may be queued, hence fewer discarded, resulting in a lower average clp. However, the fraction of loss bursts of a certain length is not at all affected, because once the buffer is full, losses occur till the buffer is drained regardless of its absolute size. Bigger buffers mainly imply that the onset of congestion, so to speak, is delayed. This indicates the need for hybrid queueing together with prioritized discarding to improve loss behavior. Finally, as seen from Figure 7(a), the gains, even for the average clp with increasing buffer sizes, become much smaller as the AFs decrease.[2] This motivates efficient traffic shaping and congestion control techniques together with buffer sizing to achieve good QoS.

[2]Since cell-loss probabilities are on the order of 10^{-9} with buffer size 90 for AF $= 0.8$ in Figure 7(a), we do not increase buffer size any further for this AF.

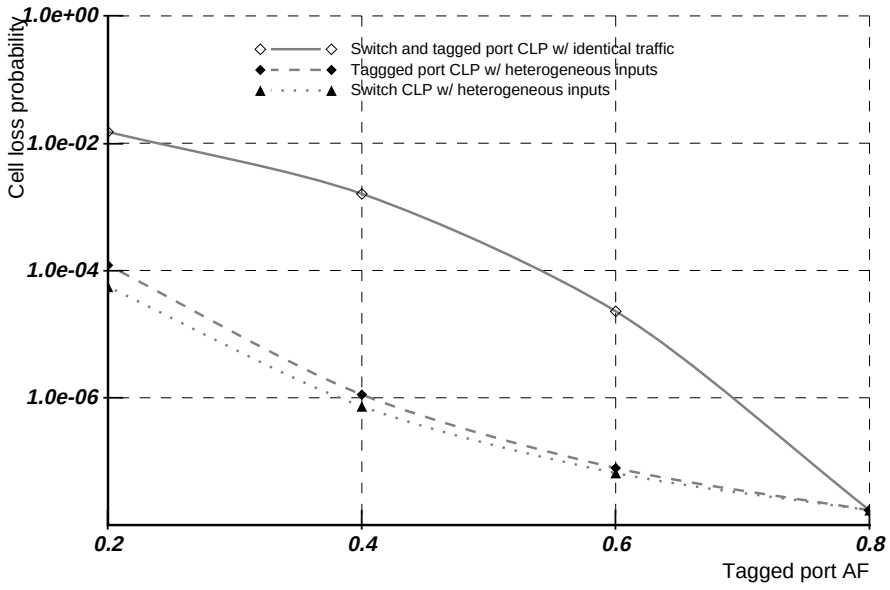

Figure 8: Effect of AF with heterogeneous inputs on clp

B Performance with heterogeneous inputs

In this subsection, we use one set of burst parameters for 7 ports and a different set for the tagged port, and we vary the tagged port's burst parameters. To maintain brevity, we present results only for the case with varying AFs and fixed ρ and BS.

Effect of heterogeneous inputs on the average clp Figure 8 represents the effects of heterogeneous inputs on the clp as seen at the switch and at the tagged port. The upper solid curve represents the clp experienced by the switch and a tagged port when all the AFs are the same and vary between 0.2 to 0.8, as indicated. The lower two curves are the clp for the tagged input and the switch, respectively, when the AF of the tagged input is varied from 0.2 to 0.8 and the AFs of the remaining inputs are fixed at 0.8.

The curves in this figure illustrate what happens when one input is significantly burstier than the other. Consider the first set of values when tag AF = 0.2. When all AFs = 0.8, the clp is on the order of 10^{-7}. However, when the tag AF alone becomes 0.2 with the others still at 0.8, the switch clp degrades to around 10^{-4}. The tag clp, however, which is around 10^{-2} when all AFs = 0.2, benefits when it alone is bursty (0.2) and the others are not (AFs = 0.8), since its clp is now around 10^{-4}. Thus, the remaining $(N-1)$ "well-

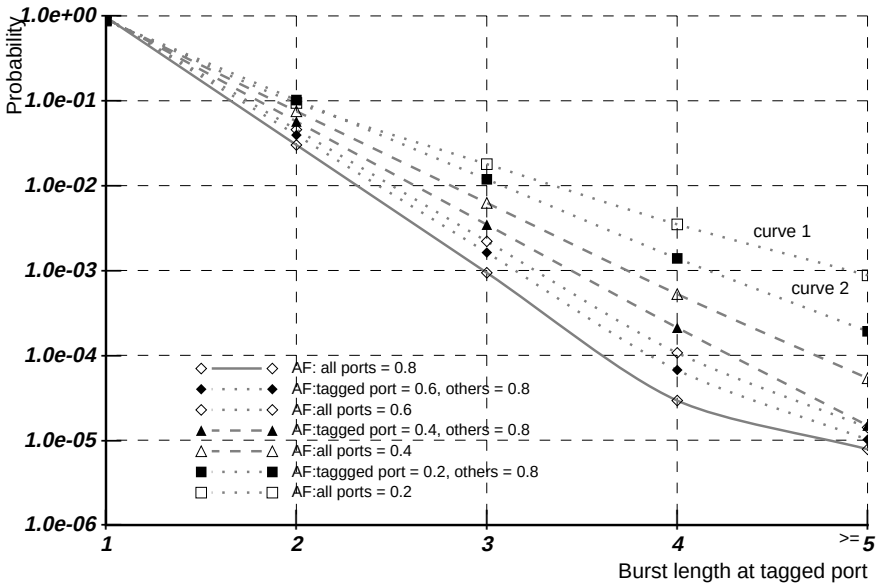

Figure 9: Effect of AF with heterogeneous inputs on distribution of consecutive cell loss for a tagged port

behaved" (smooth) inputs seem to have lost considerably in performance, while the one "not-so-well-behaved" (bursty) input has gained at their ($(N-1)$ inputs) expense! Notice also that as the disparity between the tagged port AF and the others decreases, so does the "gain" in clp of the tagged input. Thus, such anti-social behavior by even a few "ill-behaved" users can cause serious problems to the remaining "well-behaved" users, especially since the former can hog up network resources at the expense of the latter. However, if the average network usage alone is monitored, this behavior may go unnoticed, thereby hurting the QoS for many "well-behaved" users. These curves therefore provide useful insights into the behavior with heterogeneous traffic underscoring the need for efficient policing and traffic control mechanisms.

Effect of heterogeneous inputs on the distribution of consecutive cell losses In Figure 9, ρ is fixed at 0.8, BS at 8, and M at 30. The AF of the tagged input is varied from 0.2 to 0.8, while the AF of the remaining inputs is kept at 0.8. Also provided for comparison are the curves when all the inputs have AFs of 0.6, 0.4, and 0.2.

To begin, consider the curves labeled "curve 1" and "curve 2" in Figure 9. Notice again the behavior when one input alone is very bursty (curve 2). The bursty input is seen

to benefit in terms of the consecutive cell losses at the expense of the remaining smooth inputs by almost an order of magnitude, i.e., the fractions of loss lengths ≥ 5 in curve 2 are lower than their counterpart in curve 1. The other two pairs of curves (AF: tagged port = 0.4, others = 0.8 and AF: tagged port = 0.6, others = 0.8) demonstrate a similar effect. These curves again confirm the problems of mixing heterogeneous inputs *and* maintaining adequate QoS for *all* inputs, emphasizing the need for efficient traffic shaping and policing mechanisms.

V Conclusions

In this paper, we emphasize two important issues: (a) performance modeling of ATM switch architectures and (b) choice of appropriate performance measures while examining switch behavior with correlated input. Issue (a) requires the development of efficient techniques/tools that can capture the details of a given FPS architecture without sacrificing accuracy and help obtain an accurate mathematical description of the switch. To achieve this, we use stochastic activity networks (SANs) and *UltraSAN* to demonstrate the power offered by the SAN formalism. The FPS was modeled very conveniently using SANs, and the detailed Markov processes associated with the switch and workload were generated automatically and solved numerically with *UltraSAN*.

Regarding issue (b), we study the distribution of consecutive cell losses instead of the commonly studied average clp, since many applications may not receive adequate QoS if losses occur consecutively despite a low average clp. In particular, we compute and present results for the fraction of loss bursts of length m (as defined in Section II, Subsection C) for varying m. Our results provide useful insights into switch behavior with correlated and heterogeneous inputs, both to switch designers and network engineers. Specifically, large BSs or low AFs alone do not necessarily imply poor performance (both clp as well as distribution of consecutive cell loss). It is the combination of low AFs and large loads that cause problems, indicating the need for efficient CAC strategies based on the AF-load pair combination.

Further, while the average clp with homogeneous traffic for the switch and a tagged input are the same, the fraction of loss bursts of particular lengths for the two differ considerably despite homogeneity. However, as the values seen at the switch are averaged over all sources, the higher loss bunching at the switch does not imply poor QoS from the switch, with the

tagged stream exhibiting a much lower loss bunching. This indicates the need for examining behavior with respect to a tagged port even while assessing switch performance.

The fraction of loss bursts of particular lengths were seen to be affected significantly by varying AFs despite a fixed average load. This emphasizes the need for efficient traffic shaping mechanisms at the source to reduce traffic "peakedness" in order to achieve good performance. With respect to buffer sizing, though the average clp improved with increasing buffer sizes, the fraction of loss bursts of given lengths was unaffected. This indicates that increasing buffer size alone to improve loss performance is not profitable and perhaps requires hybrid queueing together with efficient congestion control and priority discarding mechanisms.

Finally, the detrimental effect of one "ill-behaved" (very bursty) input on the remaining "well-behaved" (smoothened) inputs was demonstrated by employing heterogeneous inputs. The gain in performance of one bursty source at the expense of the remaining smoother sources was seen to increase as the disparity in their burstiness increased. This provides useful insights into problems while providing adequate QoS across heterogeneous inputs, calling for efficient policing and traffic control mechanisms.

Acknowledgments We are grateful to the anonymous reviewers for their helpful suggestions.

REFERENCES

[1] **Ahamadi, H and Denzel, W E** "A survey of modern high-performance switching techniques," *IEEE Journal on Selected Areas in Communications*, Vol 7 No 7 (1989) pp 1091-1103.

[2] **Altman, E, and Jean-Marie, A** "The loss process of messages in an M/M/1/K queue," *INFOCOM*, (1994), pp 1191-1198.

[3] **Bianchi, G and Turner, J S** "Improved queueing analysis of shared buffer switching networks," *IEEE/ACM Transactions on Networking*, Vol 1 No 4 (August 1993) pp 482-490.

[4] **Bondi, A B** "On the bunching of cell losses in ATM-based networks," *GLOBECOM*, (1991), 14.1.1-14.1.4, pp 444-447.

[5] **Bondi, A B, and Lai, W-S** "The influence of cell loss patterns and overheads on retransmission choices in broadband ISDN," *Computer Networks and ISDN Systems*, Vol. 26, (1994), pp 585-598.

[6] **Cidon, I, Khamisy, A, and Sidi, M** "On packet loss process in high-speed networks," *INFOCOM*, (1992), 2C.3.1-2C.3.10, pp 242-251.

[7] **Cidon, I, Khamisy, A, and Sidi, M** "Analysis of packet loss processes in high-speed networks," *IEEE Transactions on Information Theory*, Vol 39, No 1, (1993), pp 98-108.

[8] **Denzel, W E, Engbersen, A P J, Iliadis, I and Karisson G** "A highly modular packet switch for Gb/s rates," *ISS*, Vol 2 A8.3 (October 1992) pp 236-240.

[9] **Gianatti, S and Pattavina, A** "Performance analysis of ATM Banyan networks with shared queueing - Part 1: Random offered traffic," *IEEE/ACM Transactions on Networking*, Vol 2 No 4 (August 1994) pp 398-410.

[10] **Heyman, D P and Lakshman, T V** "Source models for VBR broadcast-video traffic," *INFOCOM*, (1994) pp 664-671.

[11] **Jajszczyk, A and Mouftah, H T** "Photonic fast packet switching," *IEEE Communications Magazine*, (February 1993) pp 58-65.

[12] **Karol, M J, Hluchyj, M G, and Morgan, S P** "Input versus output queueing on a space-division packet switch," *IEEE Transactions on Communications*, Vol 35 No 12 (1987) pp 1347-1356.

[13] **Khalil, K M, Lue, K Q, and Wilson, D V** "LAN traffic analysis and workload characterization," *Proceedings of the 15th Conference on Local Computer Networks*, (1990) pp 112-122.

[14] **Khalil, K M and Sun Y S** "The effect of bursty traffic on the performance of local area networks," *GLOBECOM*, (1992) pp 597-603.

[15] **Lee, C W, and Andersland M S** "Minimizing consecutive packet loss in real-time ATM sessions," *GLOBECOM*, (1994), pp 935-940.

[16] **Leland, W E, Taqqu, M S, Willinger, W, and Wilson, D V** "On the self-similar nature of Ethernet traffic (Extended Version)," *IEEE/ACM Transactions on Networking*, Vol 2 No 1 (February 1994) pp 1-15.

[17] **Li, S-Q, and Mark, J W** "Traffic characterization for integrated services networks," *IEEE Transactions on Communications*, Vol 38 No 8 (1990) pp 1231-1243.

[18] **Meyer, J F, Movaghar, A, and Sanders, W H** "Stochastic activity networks: Structure, behavior, and, application," *Proceedings of the International Workshop on Timed Petri nets*, (1985) pp 106-115.

[19] **Movaghar, A and Meyer, J F** "Performability modeling with stochastic activity networks," *Proceedings of the Real-Time Systems Symposium*, (1984), pp 215-224.

[20] **Norros, I** "On the use of Fractional Brownian Motion in the theory of connectionless networks," *IEEE Journal on selected areas in Communications*, Vol 13 No 6 (August 1995) pp 953-962.

[21] **Ohta, H and Kitami, T** "Simulation study of cell discard process and the effect of cell loss compensation in ATM networks," *Transactions of IECE*, Vol E73 No 10 (October 1990) pp 1704-1710.

[22] **Oie, Y, Suda, T, Murata, M, Kolson, D, and Miyahara, H** "Survey of switching techniques in high-speed networks and their performance," *INFOCOM*, (1990) pp 1242-1251.

[23] **Osterbo, O** "Duration of heavy load sttes in an ATM network," *Queueing, Performance and Control in ATM, ITC-13*, (1991) pp 91-95.

[24] **Pattavina, A** "Nonblocking architectures for ATM switching," *IEEE Communications Magazine*, (February 1993) pp 91-95.

[25] **Pattavina, A and Gianatti, S** "Performance analysis of ATM Banyan networks with shared queueing – Part II: Correlated/unbalanced offered traffic," *IEEE/ACM Transactions on Networking*, Vol 2 No 4 (August 1994) pp 411-424.

[26] **Pitts, J M, Sun, Z, Cosmas, J P, and Scharf, E M** "Burst-level teletraffic modelling: applications in broadband network studies," *Third IEE Conference on Telecommunications*, (1991) pp 348-352.

[27] **Ramaswami, V and Willinger, W** "Efficient traffic performance strategies for packet multiplexers," *Computer Networks and ISDN Systems*, Vol. 20, (1990), pp 401-407.

[28] **Sanders, W H, Obal, W D, Qureshi, M A, and Widjanarko, F K** "The *UltraSAN* modeling environment," *Performance Evaluation Journal, Special Issue on Performance Modeling Tools*, Vol 24 No 1 (1985) pp 89-115.

[29] **Schulzrinne, H, Kurose, J F, and Towsley, D F** "Loss correlation for queues with single and multiple input streams," *SUPERCOM/ICC*, (June 1992) pp 219-224.

[30] **Takine, T, Suda, T, and Hasegawa, T** "Cell loss and output process analysis of a finite-buffer discrete-time ATM queueing system with correlated arrivals," *INFOCOM*, 10c.3.1-10c.3.11 (1993) pp 1259-1269.

[31] **Tobagi, F A** "Fast packet switch architectures for broadband integrated services digital network," *Proceedings of the IEEE*, Vol 78 No 1 (January 1990) pp 133-167.

[32] **de Vries, R J F** "Gauss: A simple high performance switch architecture for ATM," *Proceedings of SIGCOM*, Vol 20 No 4 (September 1990) pp 126-134.

[33] **Yeh, Y S, Hluchyj, M G, and Acampora, A S** "The knockout switch: A simple, modular architecture for high-performance packet switching," *IEEE Journal on Selected Areas in Communications*, Vol 5 No 8 (1987) pp 1274-1283.

[34] **Zegura, E W** "Architectures for ATM switching systems," *IEEE Communications Magazine*, (February 1993) pp 28-48.