Simulation Study of Statistical Delays in an ATM Switch Using EDF Scheduling

Jeffrey J. Stuckey Yannis Viniotis George N. Rouskas

TR-99-11

June 24, 1999

Abstract

In this paper we study statistical delays in an ATM switch using EDF scheduling. In particular, our interests are in the behavior of the tails of cell delay functions and their proximity to their theoretical maximum delay bounds. We observed many factors which contribute to cell delay: source behavior, the degree of multiplexing, and the synchronization of transmissions. We discovered that by staggering source transmission instants, cell delays can be greatly reduced. Finally, of the source models we studied, the repetitive burst model produces the worst-case cell delays. Our work is a first step towards understanding statistical delay guarantees in an ATM environment and can be useful in the design of statistical CAC algorithms.

Keywords: Scheduling Rules, EDF, Statistical QoS, ATM

Department of Computer Science
North Carolina State University
Raleigh, NC 27695-7534

1 Introduction

Quality of Service (QoS) guarantees over high-speed, packet-based (e.g., IP, Frame Relay) or cell-based (e.g., ATM) networks have received wide attention recently. The existence of such QoS guarantees (expressed primarily on bandwidth, delay, and loss measures) enables new network services, such as, for example, real time, multimedia communications.

Most of the recent research efforts have focused on providing *absolute* delay guarantees. Applications such as real-time audio, video and mission-critical communications have provided the necessary motivation for such guarantees. The emergence of the ATM technology has provided the required network environment, with the introduction of CBR and VBR services. The seminal work of [10] provided the theoretical foundation for absolute delay guarantees. The work of [7] simplified the theoretical proofs, unified and extended the foundation for absolute delay guarantees. It also provided some preliminary theoretical results for statistical guarantees. In summary, the end-to-end delay, d_j, that the jth packet (fixed or variable size) of a connection would encounter in a network is bounded by:

$$d_j \leq \frac{\sigma + (K-1)\max_{1\leq k\leq j} l_k}{r} + \sum_{n=1}^{n=K} (\beta_n + \tau_n). \tag{1}$$

In the above expression, σ denotes the "burst", K is the number of routers/switches that the packet travels from entry to exit in the network, l_k is the size of the k-th packet, r is the "rate" allocated to the connection, β_n is a constant that depends on the link scheduler used in the n-th router/switch (this constant may also incorporate processing delays inside the n-th router/switch), and τ_n is the propagation delay on the n-th link.

There is a number of assumptions about the overall QoS architecture, that one must be careful about before applying inequality (1) for determining the delay bound. First, the routing path of the connection must be fixed through the lifetime of the connection, since not only K, but also β_n and τ_n must be known. In connection-oriented networks, such as ATM or Frame Relay, the routing path is fixed; however, in datagram networks, like IP, the routing path may change during the lifetime of the connection. Even if a pessimistic bound is used for K, such as the maximum number of hops, β_n and τ_n may not necessarily be known. Routing techniques such as the route pinning mechanisms made possible by the MPLS architecure can solve this problem. Second, the buffer management algorithm inside the router/switch must provide the connection its own buffer space, separate from others (known in ATM networks as "per-VC queueing"). In [6], the authors have calculated the exact amount, B, of buffer space required, provided that a shared buffer architecture is used:

$$B = NL_{max} + \sum_{n=1}^{n=N} \sigma_i, \tag{2}$$

where N is the number of connections supported in the router/switch, L_{max} is the maximum packet size and σ_i is the burst size of the ith connection. As the number of connections that demand absolute delay guarantees increases, the buffer requirements can become quite significant. Third, the link scheduler in each router/switch on the connection's path must guarantee the rate r to the connection. This requires some form of signaling (e.g., PNNI for ATM, RVSP for IP diffserv/intserv networks) or static provisioning through network management procedures.

It is quite "expensive", then, for the network infrastructure to provide such absolute delay guarantees. Signaling is widely recognized as a "heavy" protocol; in fact, one of the main objectives of the diffserv IP QoS framework was to avoid reliance on signaling. Per-VC queueing requires a lot of "bookkeeping" inside a router/switch and does not scale well. Class-based queuing (CBQ) aggregates multiple connections into one queue, in an effort to minimize this bookkeeping overhead. However, as we can easily see from inequality (1), the delay bound becomes looser with CBQ (since, for example, the burst size of the aggregate traffic increases). In addition to these disadvantages, providing absolute delay guarantees is also expensive in terms of bandwidth.

Another implicit assumption in the realm of absolute guarantees is that link or router/switch failures do not occur throughout the lifetime of the connection. Even though rare, such failures do happen. In recognition of all the shortcomings of providing absolute delay guarantees, the diffserv IP QoS architecture has introduced *statistical* delay guarantees. Such guarantees can be in general expressed as: "90% of the connection's packets are delivered to their destination with a delay not larger than 500 milliseconds".

Statistical guarantees provide the potential for significant network resource savings. Indeed, CBQ schemes can be used to avoid per-VC buffer management; buffer space need not be allocated to cover the worst-case conditions expressed in equation (2); signaling can be simplified; routing need not be fixed; traffic conditioning can be relaxed; and, finally, link and/or router/switch failures can be accomodated.

As such, statistical guarantees present a paradigm shift in network services. A number of new questions and challenges arise, both from the theoretical and practical point of view. The fundamental theoretical question is: what percentiles are achievable? Reference [1] addresses this issue. In this paper we focus on statistical delay studies for EDF schedulers. Our results reveal some interesting properties of such schedulers, as we discuss in Section 4.

Several researchers have performed various percentile analyses; however, each of their works is fundamentally different from our research. Schulzrinne *et al.* [11] consider network percentile delays; however, their focus is on a hop laxity scheduling algorithm, whereas we consider EDF. Chao [2] gives a very brief percentile analysis of 4 VCs; however, he uses a different "on-off" source model than ours. One of the source models that we use for our research on EDF is part of the ATM Forum standard and is also used by Wright [13]. Wright gives tail delay analyses; however, he considers WFQ scheduling. Zhang [14] performs percentile analyses; however, he uses a different scheduling algorithm and source model than ours. For simulation evaluations of statistical delays, under a variety of link schedulers, see also the work in [9, 15, 16, 12, 13].

The paper is organized as follows. In Section 2, we provide a brief overview of the EDF scheduling algorithm and some of its variations. In Section 3 we describe our simulation environment, including the source models used. In Section 4, we present and discuss the most important findings from our experimentation. Finally, in Section 5, we summarize our work and discuss some possible future research topics.

2 EDF Scheduling

EDF [5] is a dynamic priority algorithm, proposed initially for CPU scheduling. It has recently attracted attention as a link scheduling algorithm for a number of reasons. In the context of link scheduling, the algorithm can be summarized as follows. Each cell is assigned a "deadline" upon arrival at the switch. The deadline is the same for all cells of a certain connection. The "expected departure time" of a cell is calculated by adding the deadline to the arrival time of the cell. When the output link becomes free, the cell with the most immediate "expected departure time" is chosen for transmission. Ties are resolved arbitrarily.

The motivation for using EDF comes from both theoretical and implementation advantages. From the theoretical point of view, [5] shows that non-preemptive EDF (NPEDF) is delay-optimal among all non-preemptive scheduling policies. They also show that with NPEDF for semi-flexible buffer allocation with N traffic streams into the system, a switch needs buffer space on the order of N^2 per traffic stream. From the implementation point of view, EDF is "easily" implementable, since it only requires an add operation (upon cell arrival instant) and a search for the smallest "expected departure time" (upon cell departure instant). This advantage comes at the (minor) expense of additional storage overhead, since "expected departure time" must be stored with each cell.

The challenging issue in employing EDF is use of a suitable connection admission control (CAC) algorithm. Link schedulers such as GPS require fairly simple CAC algorithms, that use a simple rate test for connection admission. CAC algorithms for EDF use more complex "schedulability conditions," that involve both the desired delay bounds and the allocated rates. Lieberherr *et al.* [8] derive a schedulability condition for NPEDF implemented in a single node. Elsayed and Perros [3] give schedulability conditions for the multiple node case. In their analysis they assume we have VCs i and j with requested delays d_i^{req} and d_j^{req}, respectively, where $d_i^{req} < d_j^{req}$ if $i < j$. Let σ_i represent the MBS of flow i and r_i represent its average rate. Let also $Pmax_k$ be the maximum packet size from connection k. Then, given that $\sum_{i=1}^{N} r_i < C$, where C is the link capacity, the schedulability condition is given by:

$$d_j^{req} \geq \frac{\sigma_j + \sum_{i=1}^{j-1}(\sigma_i - r_i d_i^{req}) + max_{k>j} Pmax_k}{C - \sum_{i=1}^{j-1} r_i}. \tag{3}$$

Inequality (3) is fairly complex to solve, especially when connections are added/removed dynamically. Firoiu [4] discusses a simplified CAC for EDF.

Our motivation in this study is to analyze the tail behavior of the cell delay distribution function. We want to quantify, a) how pessimistic the delay bound in inequality (1) is, under various operating conditions, b) what is the effect of system parameters such as, for example, conenection behavior and the degree of multiplexing, and, c) whether the tail exhibits an "exponential drop" behavior. Results of this nature can be used in the design of "statistical" CAC algorithms, that can replace, for example, algorithms based on inequality (3). Statistical CAC algorithms can be much simpler and offer the potential of admitting more traffic into the network, since provided delay guarantees are not strict.

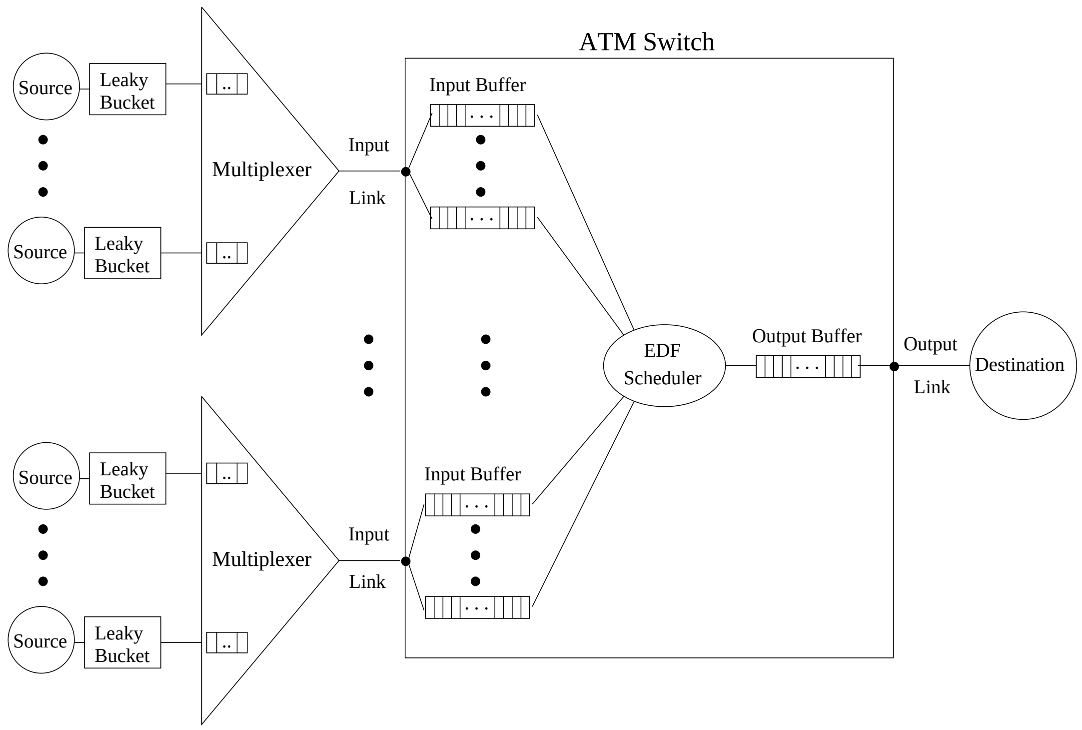

Figure 1: System topology used in simulations

3 Simulation Environment

We study a single stage of ATM multiplexing as shown in Figure 1. In our simulation, a number of sources transmit real-time traffic to a single destination. Only VBR-rt sources are considered. Both the number of sources and the number of input links are varied in our experiments. However, the input and output link rates are all fixed at the OC-3c payload rate of 149.76 Mbps.

The main parameter of interest is the delay *within* the ATM switch, defined as the amount of time between the instant the last bit of a cell arrives at an input port and the instant the last bit of the same cell has been transmitted onto the output link. We thus ignore the transmission time and propagation delay from the source to the ATM switch, and also from the ATM switch to the destination, since these components of the delay are constant. We also ignore the delay incurred due to the leaky bucket at the source, since this delay depends on the arrival statistics of the source process feeding the leaky bucket. Consequently, these delay components are not taken into account when calculating the delay bound for each cell. We also assume that before the simulator has begun, all sources have successfully established exactly one virtual circuit to the common destination. Thus, all sources are initially ready to send data cells (as opposed to connection establishment overhead cells). Similarly, sources never close their virtual circuit with the destination, and no new VCs may be established while the simulator is in progress. As a result, no overhead due to connection establishment or termination is incurred.

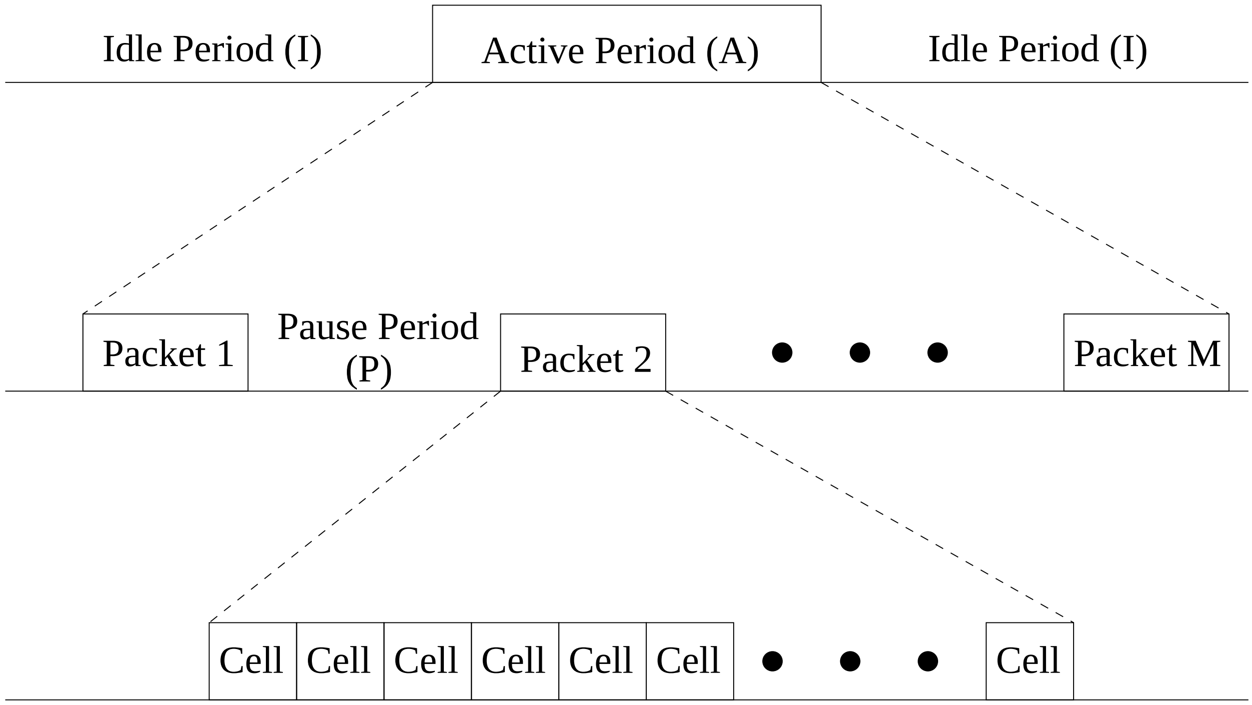

Figure 2: The three-state model for simulating bursty source behavior

3.1 Source Models

In our simulation study we have experimented with three different models for VBR sources. The first is the three-state model recommended by the ATM Forum. The other two models were chosen so as to create two different worst-case scenarios in terms of the delay experienced by the cells in the switch. Note that this delay is a function of the output process from the leaky buckets at the various sources (see also Figure 1). Thus, the source models are such that the corresponding output process from the leaky bucket exhibits an extreme behavior. Next, we describe our source modesl in more detail.

3.1.1 The Three-State Source Model

The ATM Forum recommends the following three-state source model for simulating VBR traffic. The model introduces the following concepts: active periods (A), idle periods (I), pause periods (P), the number of packets per active period (M), and the size of packets (S). For reference, see Figure 2.

Each source alternates between active and idle states. The length of an idle period is an exponentially distributed random variable of mean $\overline{I}$. The packet size S was set to 8 KB for all VCs. If r is the input link rate, the length of each active period is determined by the following expression:

$$A = M\frac{S}{r} + (M-1)P \tag{4}$$

M, the number of packets in the active period, is a geometrically distributed random variable.

According to the model, after generating each of the M packets, a source pauses for time P, where P is an exponentially distributed random variable with mean $\overline{P}$. The pause period takes into consideration the time that sources spend accumulating data into a packet. Intuitively, pause periods should be smaller than idle periods on average. Our choosing of their respective mean values reflects this intuition.

In initializing the three-state source model, we experimented with starting a source at the beginning of both an active period and an idle period. When we started all sources on active periods, each source sent the first cell from the first packet in the active period to its leaky bucket at a global time of zero. We refer to this mode of initialization as "aligned" mode, since all sources are synchronized to begin initial transmissions at the same instant in time.

In other experiments, we started sources at the beginning of an idle period. In this case, at simulation initialization, each source waits for a random idle period before becoming active and transmitting its first burst of cells. We refer to this mode of initialization as "staggered" mode, since all sources begin initial transmissions at different instants in time. We have observed that "staggered" mode experiments produce much more favorable worst-case cell delays than do similar "aligned" mode experiments.

An advantage of the three-state source model is that, by appropriately selecting the mean values of the various parameters, the sources can be very bursty. In measuring the ratio of observed peak cell rate to mean cell rate over 1-second intervals, we have found that for typical values of the source parameters, this ratio has maximum values on the order of 100. This means that these simulated sources can be extremely bursty. As a comparison, peak to mean ratios for the MPEG video traces we used were typically on the order of 10 to 20.

A disadvantage of the model is the choice of the exponential distribution for generating random variables for the idle and pause periods. Because of the "memoryless" property of the exponential distribution, each random variable is independent of its predecessor. This simplifying assumption, however, is not true in actual VBR sources. Typically, idle periods are correlated, as a source will likely have smaller idle periods when the transmitting application has data to send and will likely have longer idle periods when the application needs to wait for user input, disk I/O, etc. Despite this disadvantage, we chose this three-state model because of its ease in implementation as well as its conformance to the ATM Forum's recommendation for source modeling.

3.1.2 The Persistent Source Model

Under the "persistent" model, sources transmit cells continuously at a rate of PCR. Obviously, the vast majority of these cells are not conformant and are dropped by the corresponding leaky buckets. In this model, the output process is "worst-case" in the sense that the leaky buckets will accept individual cells and transmit them to the switch *as soon as* their respective parameters allow.

The pattern of cells at the leaky bucket output of a persistent source is shown in Figure 3. In this figure, we note that the first MBS cells are leaky bucket conformant. After the last cell in the first burst, cells are leaky bucket conformant at a rate of SCR.

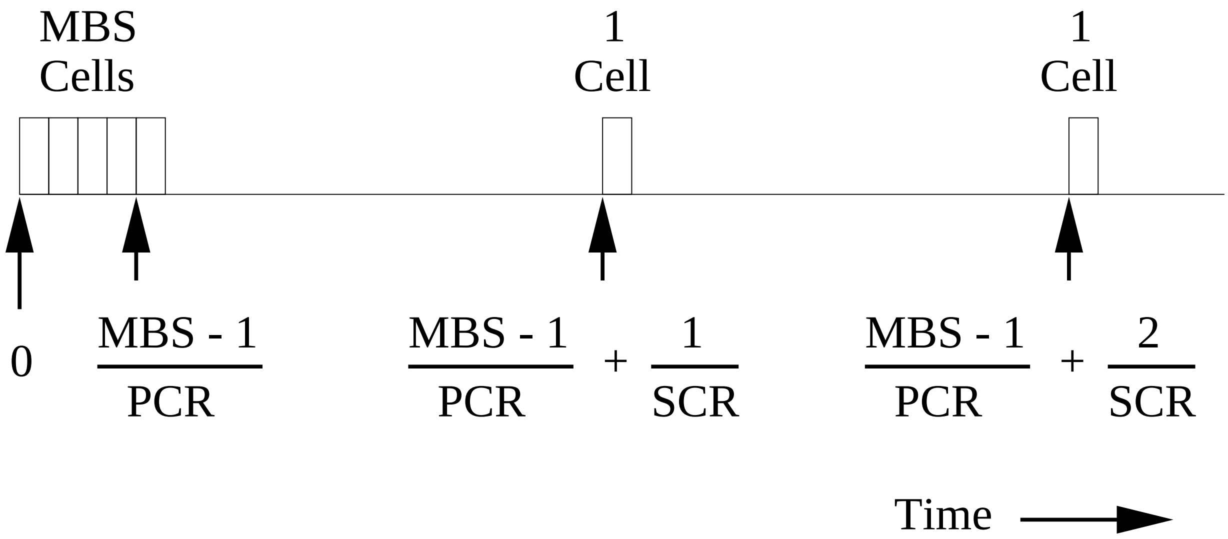

Figure 3: Leaky bucket output pattern from a persistent source

3.1.3 The Repetitive Burst Source Model

The "repetitive burst" source model represents another worst-case scenario in that the leaky buckets accept an entire burst of cells as soon as their parameters allow. In this model, a source repeats a cycle of sending a burst of MBS cells to its leaky buckets and then waiting before sending another burst of MBS cells. The wait time between each burst is just long enough (i.e., equal to $\frac{\mathrm{MBS}}{\mathrm{SCR}}$ units of time) that every cell of the burst will be conformant to the leaky buckets. The pattern of cells output from the leaky buckets is shown in Figure 4. Since all cells are conformant, the leaky bucket output process is identical to the departure process at the source, except that it is shifted in time.

3.2 Buffers

There are two types of buffers in the ATM switch under study: input buffers and output buffers. Specifically, there exists one separate input buffer dedicated to each VC and one common output buffer for all VCs (see Figure 1). Arriving cells are stored in an input buffer and are transferred to the output buffer, if it is not full. Cells are inserted into the output buffer in order of increasing deadlines. While the output buffer is non-empty, the switch transmits cells at the output link speed. If the output buffer is full, no cells are transferred from an input to and output buffer. Therefore, in our simulations cells will never be dropped in an attempt to enqueue in the output buffer. Although it never occurred in our simulations, cells could, in theory, be dropped at their input port if their input buffer is full. In this light, we can view the output buffer as merely a shared extension of the input buffers.

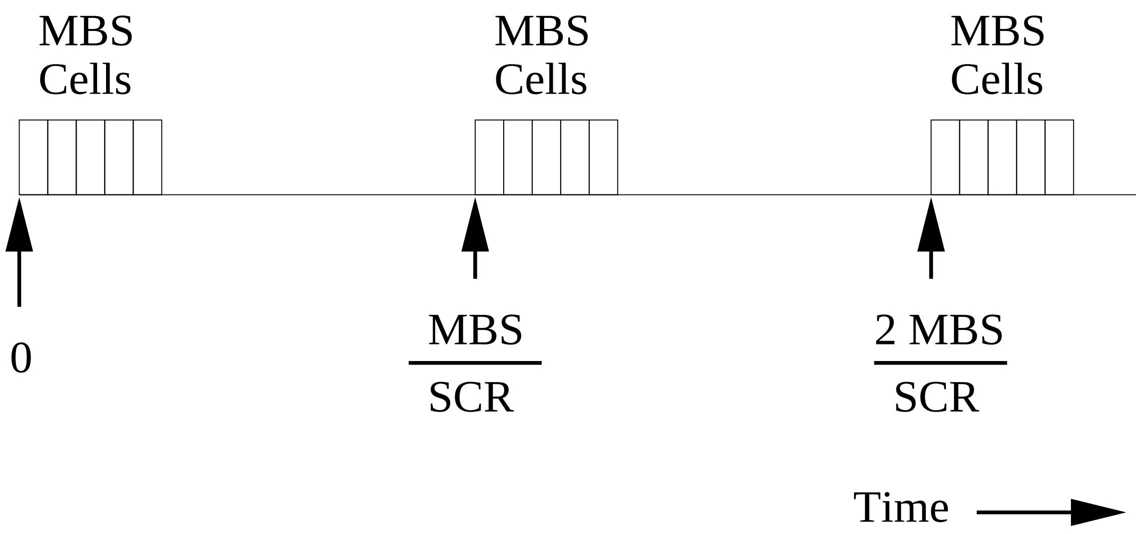

Figure 4: Leaky bucket output pattern from a repetitive burst source

4 Simulation Results and Discussion

In this section we present simulation results of cell delays in the switch of Figure 1. All experiments were run until a total (over all sources) of 10,000,000 cells were transmitted on the output link of the switch.

We focus our research on worst-case cell delays and thus the tails of cell delay distribution functions. Our interest in worst-case cell delay is coupled with our interest in EDF's behavior in these worst-case scenarios. Through our experimental results, we provide a barometer for both future research in this field and also in ATM switch development using EDF scheduling.

An advantage of our worst-case approach is that the results are independent of the actual network traffic. The disadvantage of this approach is that while these worst-case scenarios are possible, they would rarely occur in an actual ATM network. It should therefore be kept in mind that the results from our worst-case experiments are more pessimistic than would likely be observed under average network conditions.

To simulate worst-case scenarios, in most of our experiments we aligned the sources such that they start transmitting cells at simulation time $t = 0$. When all sources start transmiting bursts of cells simultaneously at the beginning of the simulation, worst-case delays result since the switch is flooded with cells. We use the term "warm-up period" to denote that time from simulation onset until a switch has recovered from the initial flood of cells in the aligned sources case. After its warm-up period, the switch begins serving cells with much less delay. There is no exact measure of the warm-up period as warm-up periods vary greatly depending on the switch topology, initial burst sizes, and subsequent burst sizes.

In our simulation we varied several parameters. The number of input ports was varied from 1 to 64 in powers of two. By varying the number of input ports and keeping the total number of VCs constant, we were able to observe the effects of the degree of multiplexing of sources onto links. The degree of multiplexing has a very profound impact on the delay values we obtain, in that the greater its value, the smaller the worst-case delays observed [1].

One of the most critical parameters in our study was the number of VC classes using the switch. In the homogeneous cases (i.e., when all VCs are identical, and thus, have the same theoretical delay bound in (1), we observed the tail of the cell delay function to drop drastically. However, in the heterogeneous cases, we have noticed that the competition between the various VC classes can lead to peaks at the tails of the cell delay functions.

In presenting our simulation results, we focus on the VC from each class for which its maximum delayed cell is also the maximum delayed cell over all VCs in its class. In explaining some of the results, a parameter we call the "EDF threshold" will be useful. The EDF threshold, T_i, for a given cell from VC$_i$ with theoretical maximum delay bound, d_i^{max}, is defined as:

$$T_i = d_i^{max} - max_{j:d_j^{max} < d_i^{max}}\{d_j^{max}\} \tag{5}$$

Let a cell from VC$_i$ arrive at the switch at time t. The EDF threshold is the maximum amount of time before this cell's priority, due to an imminent deadline, becomes higher than that of any newly arriving cell. In other words, if this cell has not been served by time $t + T_i$, then it will be served (i.e., exit the switch) before any other cell arriving at time greater than $t + T_i$.

In the rest of this section we discuss details of our simulation results organized according to source model.

4.1 Persistent Sources

4.1.1 The Aligned Case

In our first experiment we consider 256 sources (VCs) and a switch with 64 input links. The VCs are partitioned in two classes, with odd-numbered VCs belonging to class A, and even-numbered VCs belonging to class B. The VCs within each class have the same leaky bucket parameters (MBS, SCR, PCR). These parameters were chosen so that class A cells have a lower delay bound (and, therefore, higher priority) than class B cells. Whenever the EDF scheduler has to select among two or more cells of the same priority, it selects cells in order of increasing VC number.

In Figure 5 we plot the delay CDF for the two VCs, one for each class, that experience the worst case cell delay (because of the tie-breaking rule we used, it is not surprising that these are the two VCs with the highest number). We observe that cell delays come very close to reaching their theoretical maximum delay bounds. However, the cells with the highest delays are those transmitted during the initial burst of

[1]As more VCs are multiplexed onto an input link, the bulk of the delay that a cell encounters during the switch's warm-up period tends to shift from occurring inside the switch to occurring inside the multiplexer. The inverse is true as the degree of multiplexing is decreased. From a user's perspective, where the delay occurs along a cell's path is irrelevant. However, our focus on delays is from the perspective of the switch which is only concerned with delays incurred while it buffers the cells. It is to the benefit of the user if a large degree of multiplexing can be achieved, since this will allow the switch to accept more VCs while still providing the same QoS to the user at a reduced cost.

the respective source. The reason these warm-up period cells incur much larger delays than subsequent cells is due to the contention arising within the switch at the start of the simulation (refer to Figure 3). After the initial bursts of cells have been served, the switch stabilizes and is able to serve cells arriving at their SCR with very minimal delays, as evidenced by the large peaks of cells at very low delay values in Figure 5.

We also observed that the worst-case delay of class B cells occurs for the last cell of the first burst of VC 256, since this cell must wait until many class A cells and all other class B cells in its same burst have been served. The worst-case delay for class A cells occurs for those cells which arrive at the switch just after the first class B cells reach their EDF threshold (sinse at that time, class B cells already in the switch have higher priority, by the definition (5)).

The relatively even distributions of cells over large delay ranges in Figure 5 is due to the round-robin service within a VC class. In the figure we also note an increase in the number of cells from VC #256 that experience a delay higher than the VC's EDF threshold which is equal to 83,777 μs. Cells from VC #256 which incurred more delay than this threshold are those which were competing with the initial flood of class A cells. The number of these cells is relatively large since, at the start of the simulation class A cells have higher priority and all arriving class B cells are buffered. Those cells from VC #256 which incurred less delay than this threshold arrived after the initial flood of class A cells had exited the switch. They, therefore, competed largely with other class B cells as well as the relatively few class A cells which continued to arrive at SCR.

4.1.2 The Staggered Case

In Figure 6 we present delay CDF results similar to those in Figure 5, except that in this experiment there are 256 VCs in four VC classes, and the sources are staggered such that at the start of simulation each VC is idle for a random period of time. After this initial idle period, a source persistently sends cells to its leaky buckets. As we can see, the maximum observed cell delays in this case are extremely small compared to the theoretical delay bounds. This result can be explained by observing that each VC's first burst of cells only compete with cells from VCs which had shorter initial idle periods and whose initial burst has mostly been served already. Consequently, even though the leaky buckets are exhibiting the same worst-case behavior as in Figure 5, simply staggering the sources such that they do not have worst-case starting behavior results in drastic improvements in worst-case cell delays.

4.2 Repetitive Burst Sources

4.2.1 The Aligned Case

In Figure 7 we present results for 256 repetitive burst sources (as in Figure 4) in two VC classes, and a switch with 64 input links. We observe that worst-case cell delays are relatively close to respective theoretical maximum delay bounds, as in the persistent source case in Figure 5. Comparing to the latter figure, however, we notice two important differences. One one hand, the maximum delays observed are lower in Figure 7.

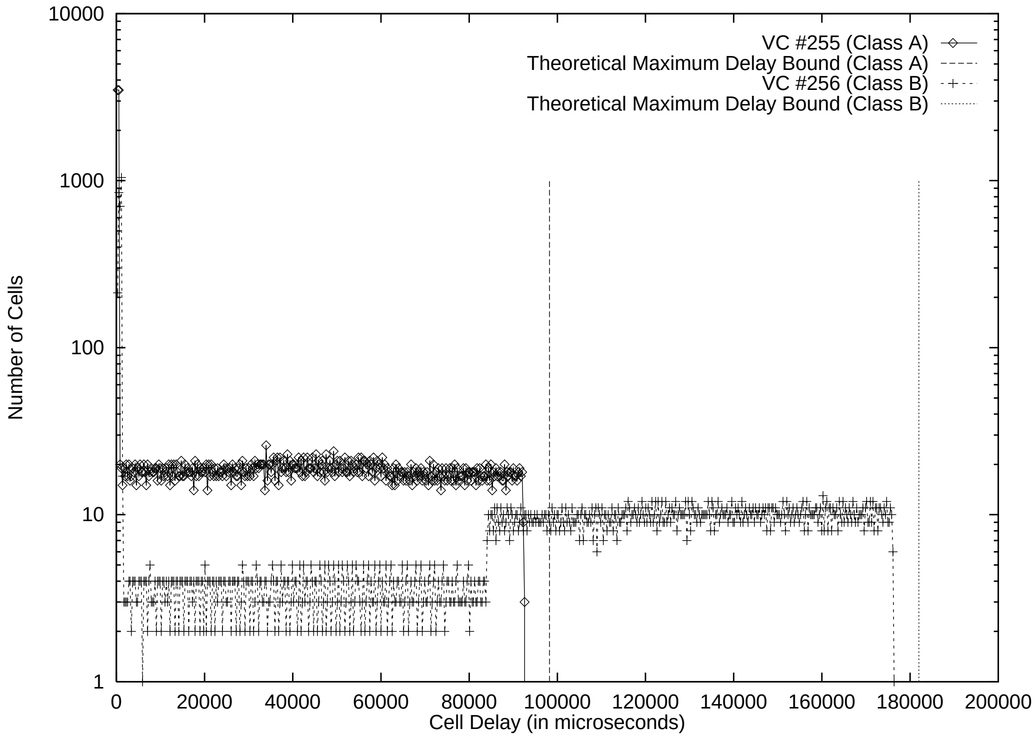

Figure 5: Cell delays with 256 persistent VCs in 2 VC classes, 64 input links, aligned mode

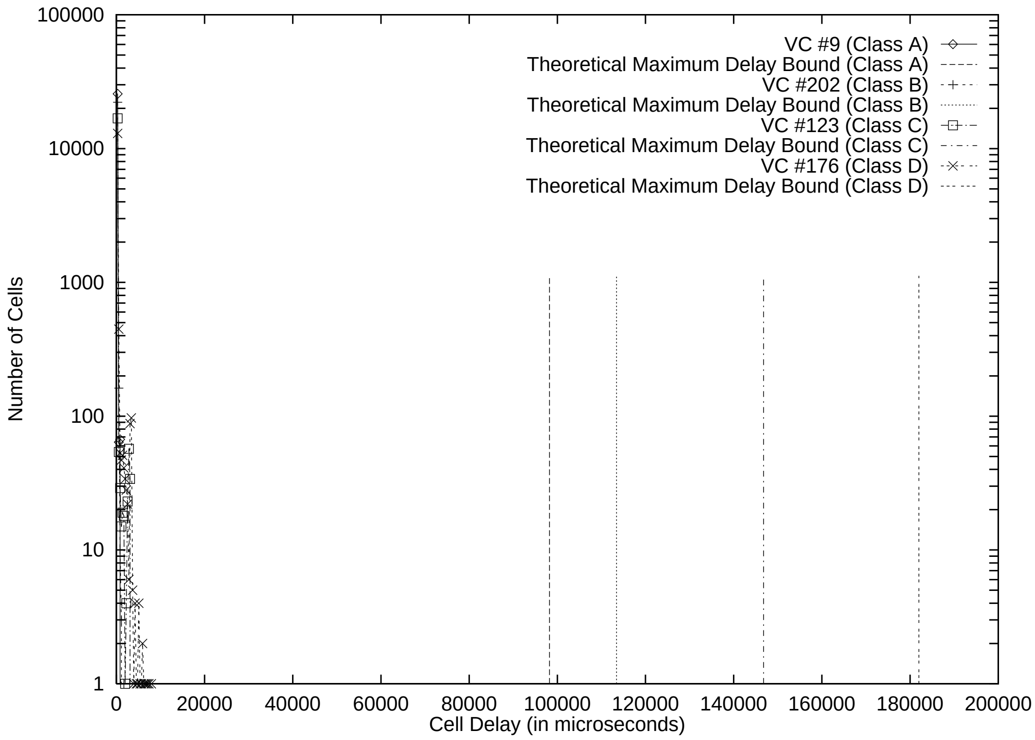

Figure 6: Cell delays with 256 persistent VCs in 4 VC classes, 64 input links, staggered mode

Also, the number of cells incurring very high delays (close to the theoretical bound) is lower than in Figure 5 (e.g., compare the mass of the delay distribution for values higher than 60,000μs for class A cells and 120,000μ s for class B cells). On the other hand, the number of cells experiencing minimal delays is very small under the repetitive source model (e.g., compare the mass of the distribution in the two figures for values less than 20,000μs). In other words, most cells experience some significant delay, unlike the persistent source model where post-warm-up period cells encountered small delays which resulted in large peaks at the beginning of the delay functions. Since cells arrive in bursts which have synchronized arrival times within each VC class, the switch is not able to serve all of these cells before the next bursts arrive. Thus, in the case where all repetitive burst sources are aligned, the switch is never able to overcome the effect of the warm-up period.

4.2.2 The Staggered Case

In Figure 8 we show results of an experiment with 256 repetitive sources in four classes, similar to that used in Figure 6. Comparing to the latter figure, we note that the observed maximum cell delays are higher than with persistent sources. However, these maximum delay values are significantly lower than the theoretical delay bounds than in the aligned source case of Figure 7. As in the persistent source experiments, we again see that alignment of the sources at the beginning of the simulation has a severe impact on the cell delays.

4.3 Three-State Source Model, Homogeneous VCs

In this section we present results for sources that follow the three-state model recommended by the ATM and described in Section 3.1.1. We first consider the homogeneous case where all VCs belong to the same class [2]; we discuss experiments with multiple VC classes in the next subsection. In all experiments, sources where in aligned mode, starting with an active period (see Figure 2) at simulation time $t = 0$.

4.3.1 The Effect of the Degree of Multiplexing

In Figures 9 and 10 we plot the cell delays for the two VCs whose cells experience the maximum delay. The two experiments whose results are plotted in the figures were identical (involving 256 VCs of the three-state type with the same parameters) except that in the experiment of Figure 9 (respectively, Figure 10) the 256 VCs were multiplexed onto 64 (resp., 2) input links with 4 (resp., 128) VCs each.

We observe that, in both experiments, the vast majority of cells incur very small delays, as evidenced by the large peaks at the left-most edge of the delay functions. We also observe that after these large peaks, the delay functions decrease dramatically after which a long tail is produced. Cells which contribute to the tails of these functions are those which depart during the switch's warm-up period. After this warm-up period, source transmissions are more dispersed in time, thus subsequent cells encounter less competition with other

[2] It is important to note that in the homogeneous VC experiments, cells are served in the order of their arrivals and the EDF scheduler behaves identical to a FIFO scheduler. However, the conclusions drawn from these experiments also apply to the heterogeneous VC case.

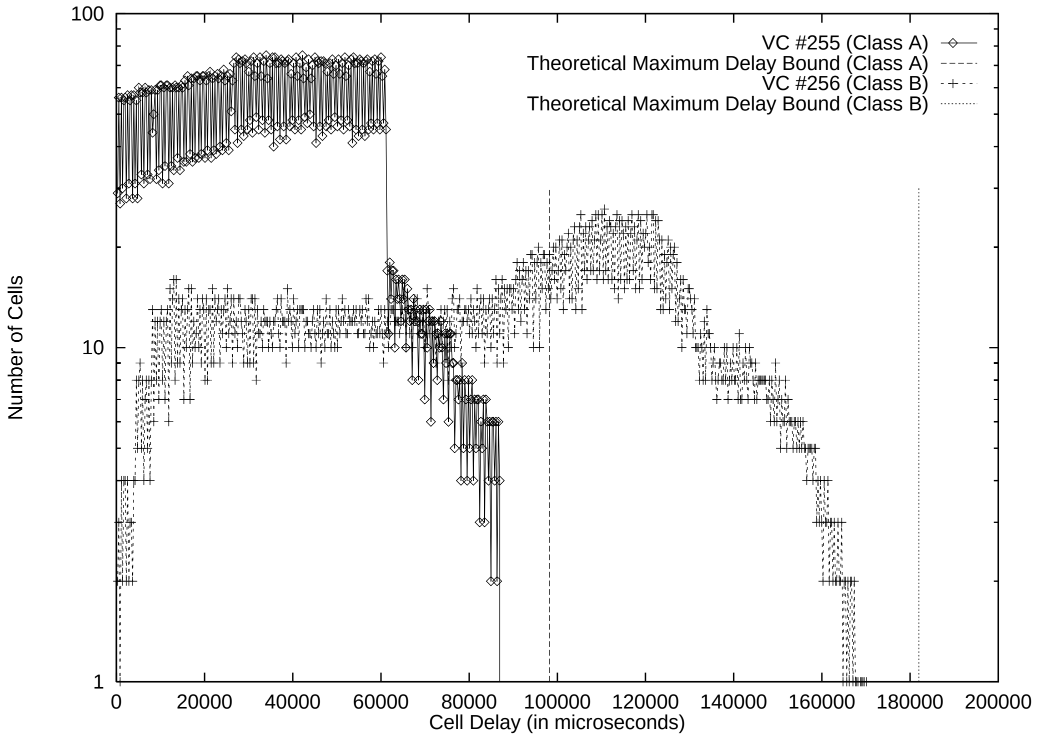

Figure 7: Cell delays with 256 repetitive burst VCs in 2 VC classes, 64 input links, aligned mode

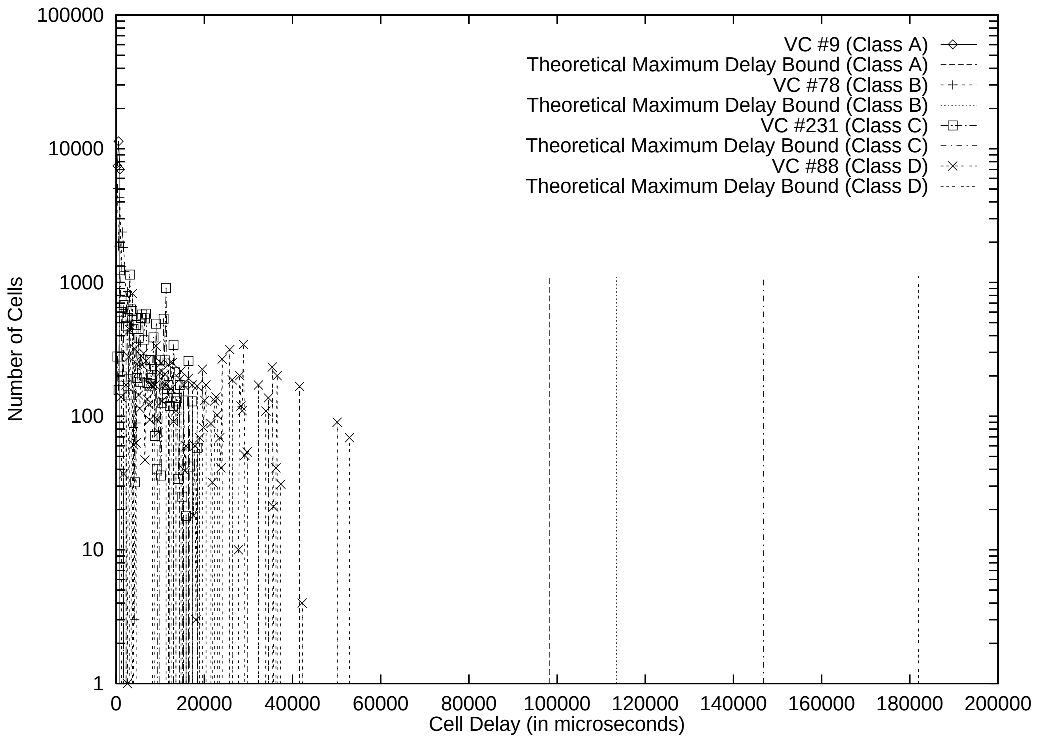

Figure 8: Cell delays with 256 repetitive burst VCs in 4 VC classes, 64 input links, staggered mode

cells inside of the switch and smaller delays result. Comparing to Figures 5 and 7 corresponding to the persistent and repetitive burst sources, respectively, we note that the heavy tails of the delay distribution observed there are mainly due to the worst-case behavior of those sources.

Returning to Figures 9 and 10 we observe that the degree of multiplexing has a profound effect on maximum delay, which drops from more than 120,000μs when the 256 VCs are multiplexed onto 64 input links (see Figure 9) to about 70,000μs when the same number of VCs are multiplexed onto just two input links (Figure 9). Put another way, a decrease in the number of input links from 64 to 2 results in a decrease of the maximum delay from a value close to the theoretical bound to a value almost half that bound. This result can be explained that the theoretical bound is derived under, among others, the assumption that cells from all sources arrive at the switch at exactly the same time. This assumption is only true when each input link carries exactly one VC, but it is not valid when several VCs are multiplexed onto a single ling. The larger the number of VCs multiplexed onto a link, the smaller the number of cells that can potentially arrive at the switch simultaneously, and the lower the maximum delay encountered in the switch.

4.3.2 The Effect of Statistical Delay Guarantees

On a per-VC basis, we define an $n\%$ delay guarantee as the amount of delay $D(n)$ such that $n\%$ of the VC's cells have encountered a delay of at most $D(n)$. In this section we study the relationship between $D(n)$ and the theoretical delay bound for different values of n. In Figures 11 and 12 we plot $D(n)$ (expressed as a percentage of the theoretical delay bound) as a function of the number of VCs multiplexed onto an input link (the number of VCs is kept constant at 256 for all experiments, thus the points along the x-axis in the two figures represent switched with different numbers of input links). Figure 11 presents results for a 97% output link reservation level and Figure 12 for a 65% reservation level. (An output link reservation level of $x\%$ means that the rate r in (1) allocated to each VC was selected such that the total rate allocated to the 256 VCs equals $x\%$ of the output link bandwidth.)

In Figure 11 we observe that with a low degree of multiplexing, providing a 100% or 99.99% delay guarantee results in delays which closely approach the theoretical maximum delay bound (also note that the value for 100% guarantee and 4 VCs per input link corresponds to the maximum delay in Figure 9). At these high delay guarantee levels, the degree of multiplexing has a significant effect on the value of $D(n)$ as we noticed in Figures 9 and 10. On the other hand, lowering the delay guarantee to 99.5% or 99% has a profound impact on the value of $D(n)$, which drops to about 50% or 20% of the theoretical delay bound, respectively. In other words, the vast majority of cells encounter delays significantly lower than the theoretical bound; in fact, from the 95% curve we see that 95% of all cells encounter delays in the order of 2% of the theoretical bound. Similar observations can be made from Figure 12, but the corresponding values for $D(n)$ are significantly lower. This result is expected, since a lower reservation level implies a lower load at the switch, and thus a lower level of contention for the output link leading to lower overall delays [3].

[3] It is important to keep in mind that the results in Figures 11 and 12 where obtained for a total of 10,000,000 cells transmitted on the output link of the switch. Because the largest cell delays occur during the warm-up period, an increase in the total number of cells simulated minimizes the effect of the warm-up period cell delays on the entire

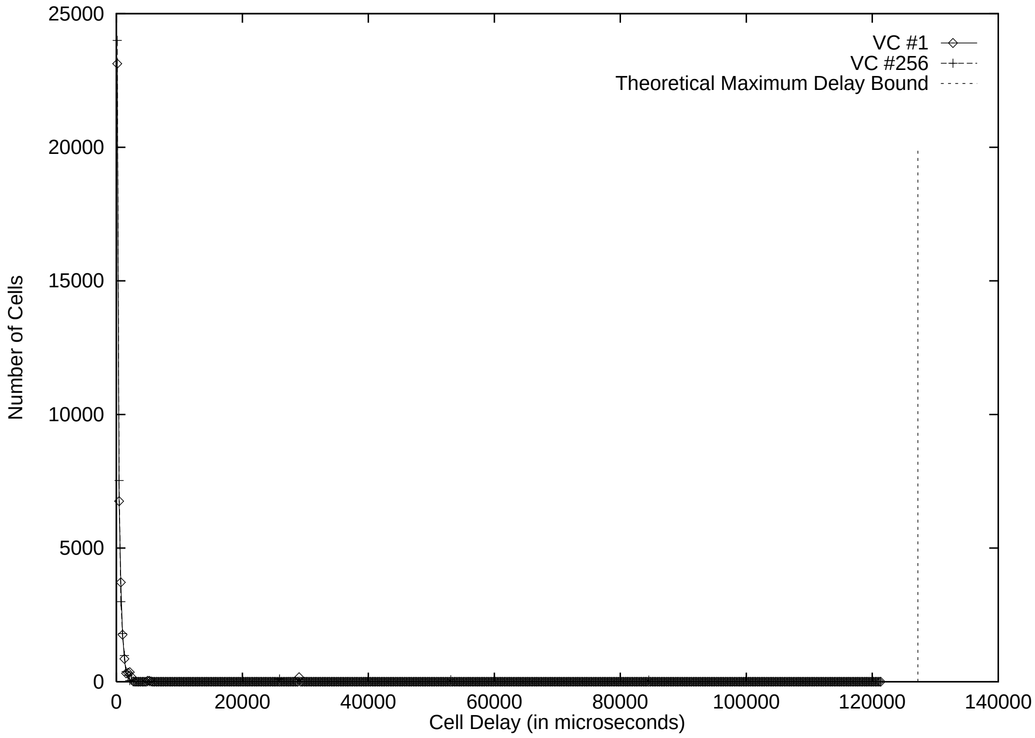

Figure 9: Cell delays with 256 VCs, 64 input links

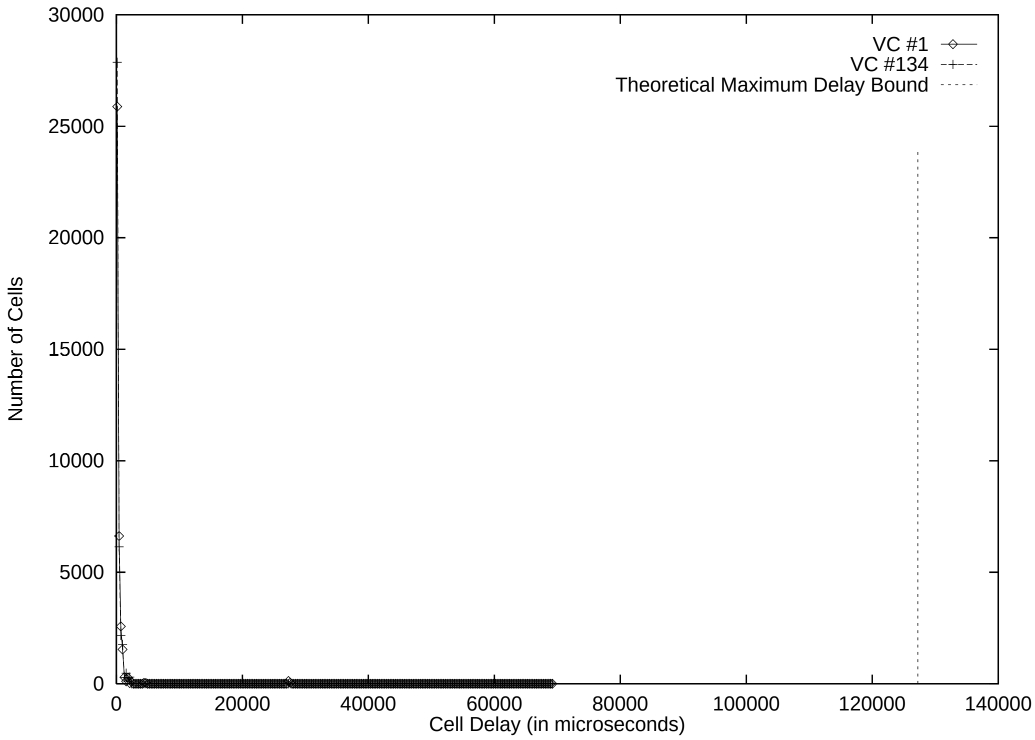

Figure 10: Cell delays with 256 VCs, 2 input links

The results in Figures 9 to 12 indicate that a CAC algorithm using the theoretical bound in (1) can be quite conservative when (a) the degree of multiplexing is relatively high, and/or (b) statistical guarantees are to be provided. Although this observation has been made by other researchers, our work is a first step towards quantifying the impact of the degree of multiplexing and of statistical guarantees. The issue of designing statistical CAC algorithms for EDF schedulers will be addressed in future research.

4.4 Three State Source Model, Heterogeneous VCs

We again consider sources following the three-state model, but we now discuss experiments with mutiple VC classes. This time our objective is to explore the potential for worst-case behavior for these sources by explicitly manipulating the various parameters of the three-state model in Figure 2.

4.4.1 Second Burst Behavior

Motivation for this experiment is provided by Figure 13. In the figure we plot the cell delays for two VCs, VC #97 of class A and VC #34 of class B for a simulation of 256 VCs in two classes multiplexed onto 32 input links. In this figure we indicate the delay experienced by the first few bursts of each VC by labeling the first two bursts of VC #97 as "1" and "2," and labeling the first four bursts of VC #34 as "a" to "d." All subsequent bursts for both VC #97 and VC #34 occur after the switch's warm-up period and therefore incur very small delays as evidenced by the extremely large peaks at the left-most edge of the delay functions.

By aligning the transmissions of the first burst of cells, cells in this burst are served in a round-robin sequence within each VC class. This results in the large range of delays for these cells (refer to labels "1" and "b" in Figure 13), similar to those observed in Figure 9. Furthermore, the large cell delays incurred by the first four bursts from VC #34 are attributed to their competition with both class A and class B cells. At the start of the simulation, class B cells are buffered in the switch until such time elapses that they are closer to their theoretical maximum delay bound than any other cell.

However, in Figure 9, the tails of the cell delay distributions are such that the number of cells which experience a given delay D is either one or zero, for $D > 2000\mu s$. This is not the case in Figure 13 as the second burst of cells of both VCs caused a peak to occur at the tail of the distribution function at delay values higher than those experienced by the first burst of cells. The question that arises, then, is whether it would be possible to come up with a worst-case scenario to increase the maximum delay by directly manipulating the parameters of the three-state source model.

It would seem that the delay for the second burst of cells would decrease if we reduced the first pause period of the VCs (refer to Figure 2). However, after we manually set the first pause period for the two VCs to a lower value while keeping all other parameters as before, the delay of the second burst of cells actually

simulation. Therefore, by increasing the number of cells simulated, we can make the curves in the figures to move arbitrarily close to the x-axis. However, the main observation that $D(n)$ decreases rapidly as the delay guarantee drops below 100% is still valid for any given number of simulated cells.

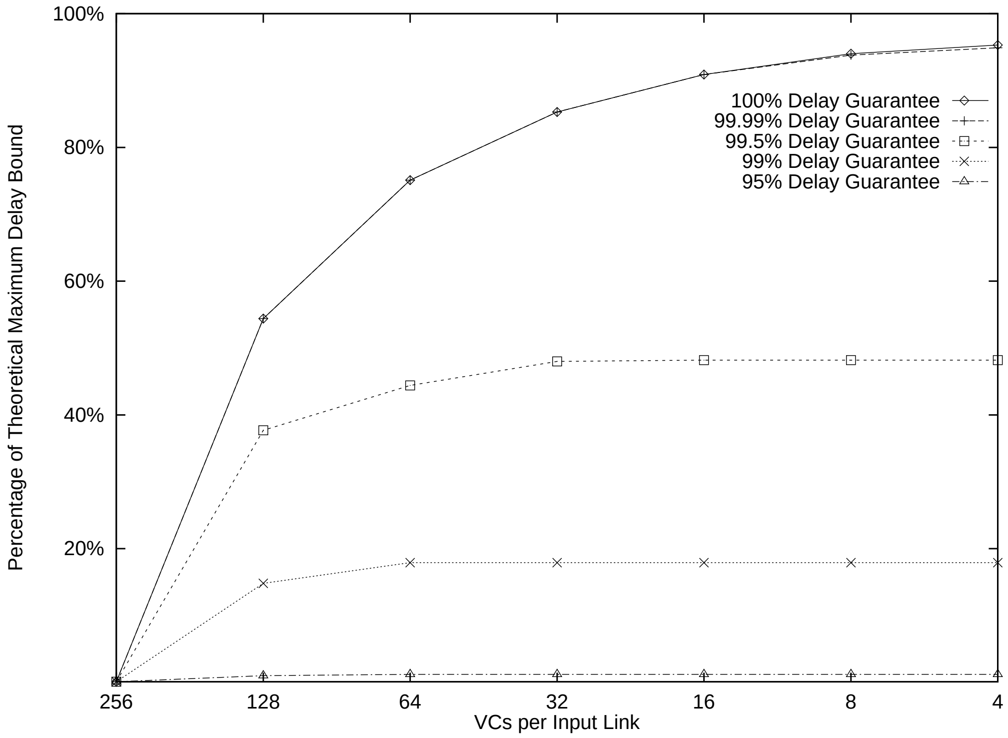

Figure 11: Delay percentages with 256 VCs and 97% output link reservation

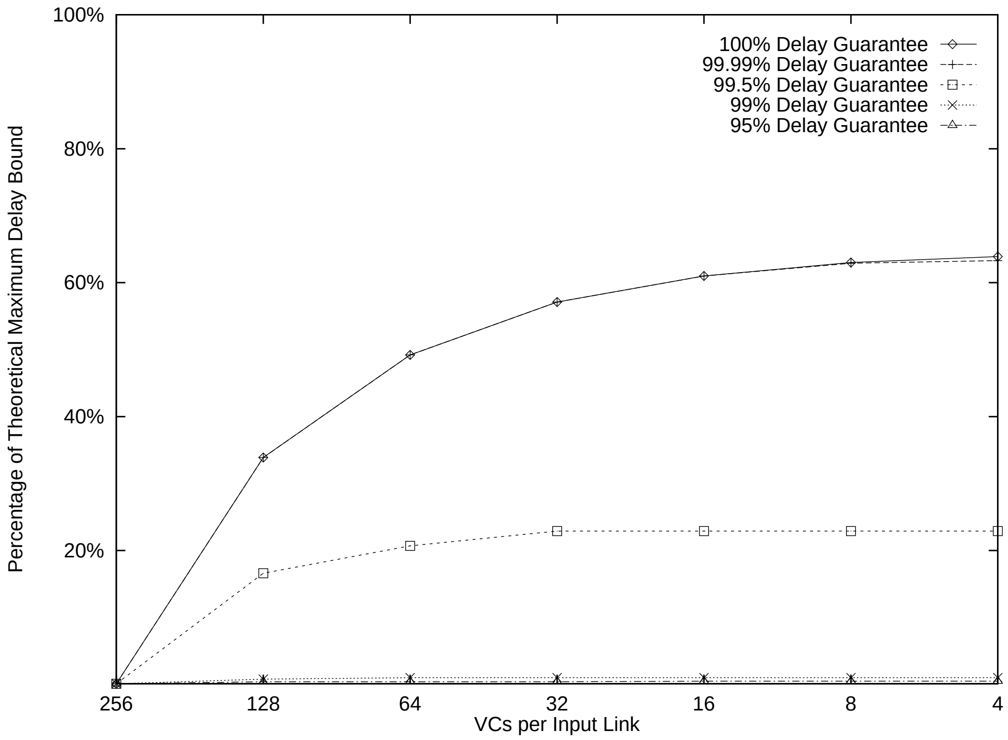

Figure 12: Delay percentages with 256 VCs and 65% output link reservation

decreased, as shown in Figure 14. This result can be explained by noting that decreasing the already small first pause period for the two VCs merely caused more cells to be non-conformant at the leaky buckets, leading to a smaller second burst size and, thus, a smaller queueing delay for those cells. Furthermore, we were not able to increase the delay encountered by the cells in the second burst by manipulating any other parameter of the three-state model.

4.4.2 Second Bounce Behavior

In Figure 15 we plot the cell delays for two VCs, VC #99 of class A and VC #34 of class B for a simulation of 256 VCs in two classes multiplexed onto 16 input links. As in the previous two figures, we indicate the delay experienced by the first few bursts of each VC by labeling the bursts of VC #99 as "1" to "7," and labeling the first four bursts of VC #34 as "a" to "d."

The most striking result in Figure 15 is that cells generated during the sixth burst of VC #99 encounter greater delays than those of either the fourth or fifth bursts. To explain how this is possible, let d_A^{max} and d_B^{max} be the theoretical maximum delay bound for class A and B VCs, respectively. Since $d_A^{max} < d_B^{max}$, we define the EDF threshold for all cells of class A, T_A, to be $T_A = 0$. Similarly, from (5), we get $T_B = d_B^{max} - d_A^{max} = 83,777\mu s$. Note, also, that when a given cell incurs a delay less than its EDF threshold then there were no cells inside the switch with a smaller theoretical maximum delay bound at the time of this cell's departure.

Based on the above discussion, the delay behavior in Figure 15 can be explained as follows. Between the time at which the last cell from VC #99's fifth burst exited the switch and the time at which the first cell of VC #99's sixth burst arrived at the switch, the delay values of the first burst of cells from all class B VCs surpassed their EDF thresholds (indeed, the delays experienced by the cells of the first burst of class B VCs in Figure 15 is seen to be greater than 83,777μs). Because of this, cells from the sixth burst of VC #99 must wait until all 128 of the class B first bursts have been served, giving them greater delays than the fifth burst of VC #99. We call this phenomenon the "second bounce" behavior.

The question that arises again, is whether there exists a worst-case scenario in which the second bounce behavior can cause an increase in the maximum delay encountered by any cell of a given VC. In Figure 16 we show the results of a controlled experiment where VC #99's fifth pause period was manually increased while all other parameters remained the same as in Figure 15. The arrival time of VC #99's sixth burst was carefully chosen such that the first cell of VC #99's sixth burst arrived at the switch just after the delays of the first burst of all class B VCs had passed their EDF threshold. In Figure 16, we observe that the cell delays of VC #99's sixth burst increase significantly and are almost at the tail of the cell delay function. However, we were not able to make the delay of sixth burst cells greater than that of the largest delay of first burst cells by direct manipulation of the parameters of the three-state model.

The objective of the two experiments in this and the previous subsection was to control the second burst and second bounce behavior to explore worst-case scenarios for the three-state source model. Despite our efforts, we were not able to construct situations in which the tail of the delay distribution is worse than that

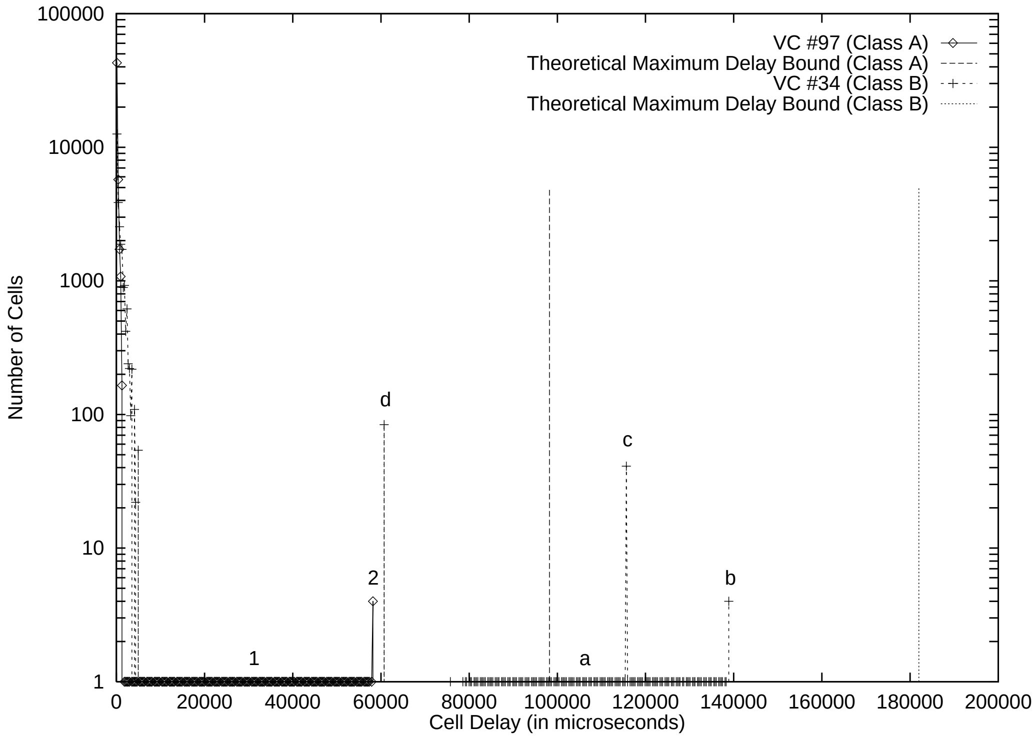

Figure 13: Cell delays with 256 VCs in 2 VC classes, 32 input links

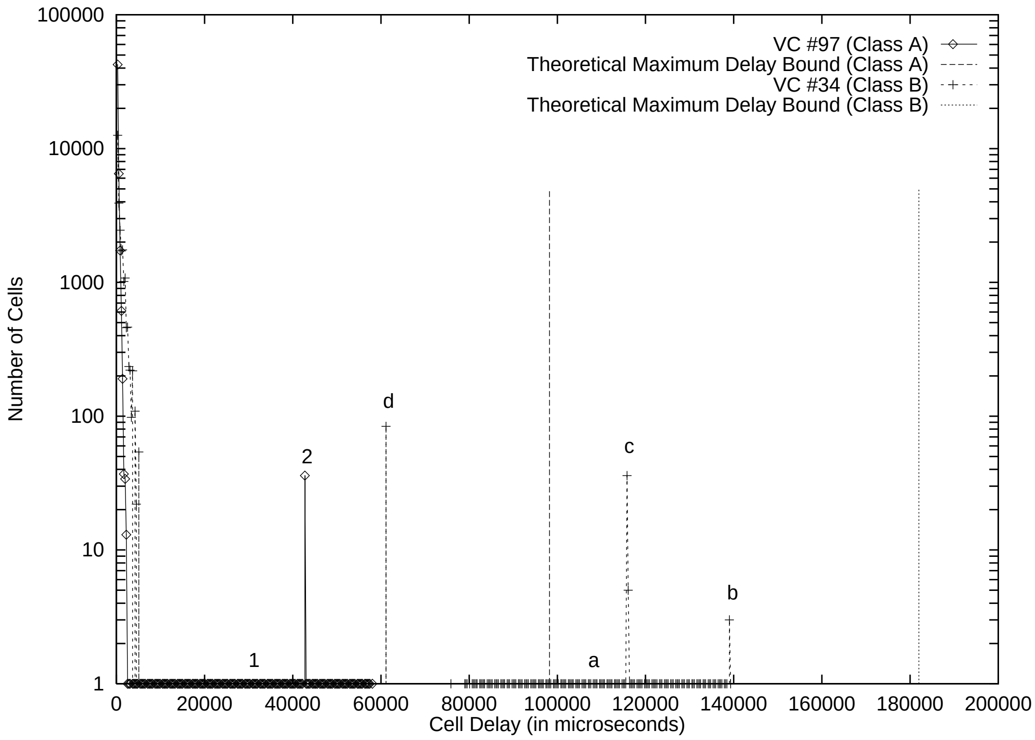

Figure 14: Manipulation of the second burst of cells

obtained under the presistent source or repetitive source models.

4.5 MPEG Traces

In addition to the three source models described above, we also run experiments in which actual MPEG video traces [4] were used to determine source cell generation times. In our experimentation with MPEG video traces, we simulated worst-case MPEG behavior. Specifically, for the results reported in Figure 17 we aligned all sources to start transmission of the first cell of *each* frame simultaneously, with subsequent cells in each frame transmitted consecutively. The figure plots the cell delays of VCs with the worst-case (VC #2 and VC #23) and best-case (VC #19) behavior. As we can see, the majority of cells experience very small delays, while the tails of the delay functions are much further from their theoretical maximum delay bound than any of the other worst-case source models.

5 Summary

The focus of this paper has been the study of statistical delays in an ATM switch using EDF scheduling. In particular, our interests are in the behavior of the tails of cell delay functions and their proximity to their theoretical maximum delay bounds. We observed many factors which contribute to cell delay: source behavior, the degree of multiplexing, and the synchronization of transmissions. We discovered that by staggering source transmission instants, cell delays can be greatly reduced. Finally, of the source models we studied, the repetitive burst model produces the worst-case cell delays.

While we have only considered a single-node network, we believe that another interesting area of research would be that of studying statistical delays across multiple ATM switches implementing EDF scheduling. Another future area of research would be an in depth analysis of the buffer requirements necessary for statistical delay guarantees, using EDF or any link scheduling algorithm.

[4] The traces we used in our simulation were downloaded from
`ftp-info3.informatik.uni-wuerzburg.de/pub/MPEG/`

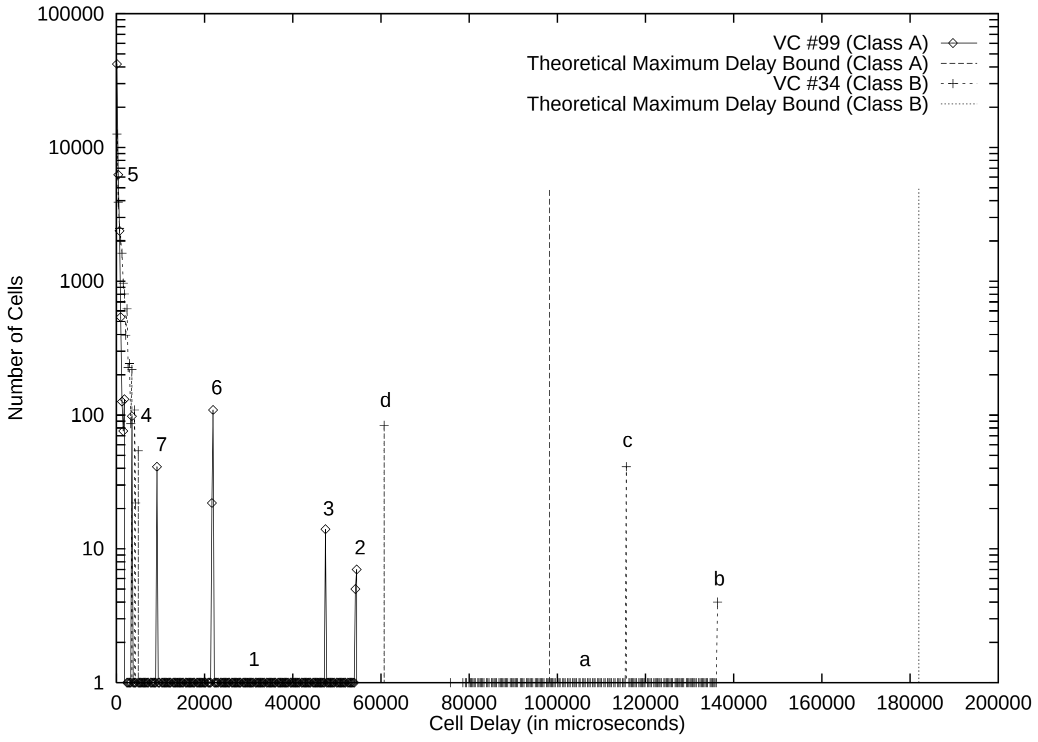

Figure 15: Cell delays with 256 VCs in 2 VC classes, 16 input links

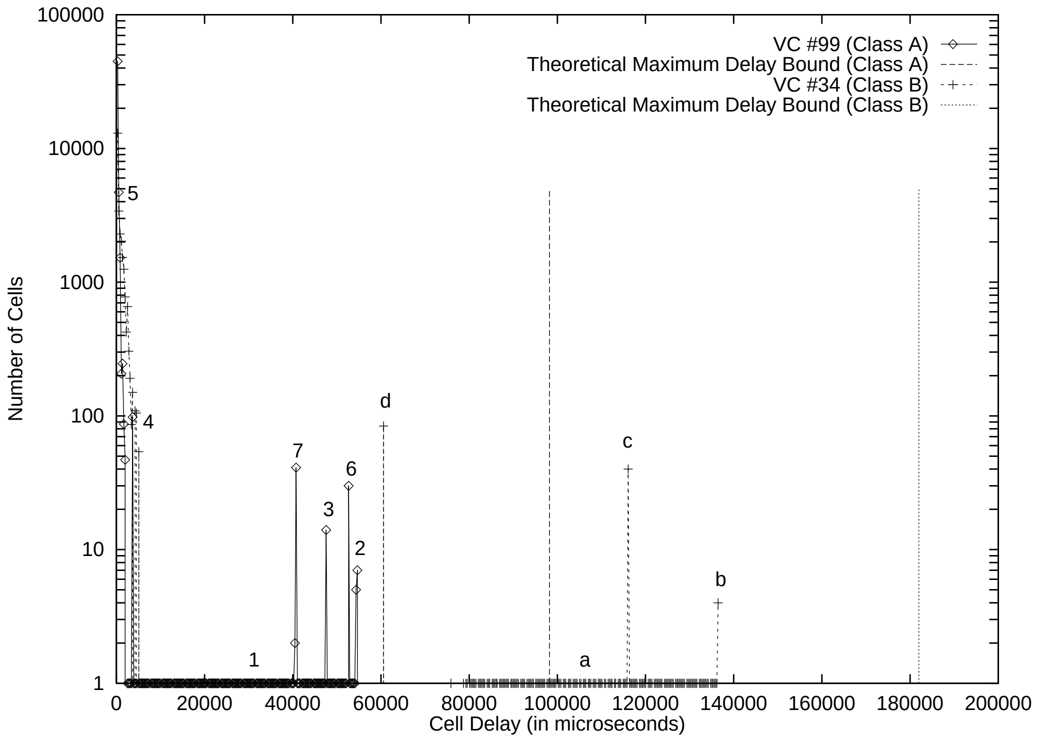

Figure 16: Manipulation of the second bounce in cell delays

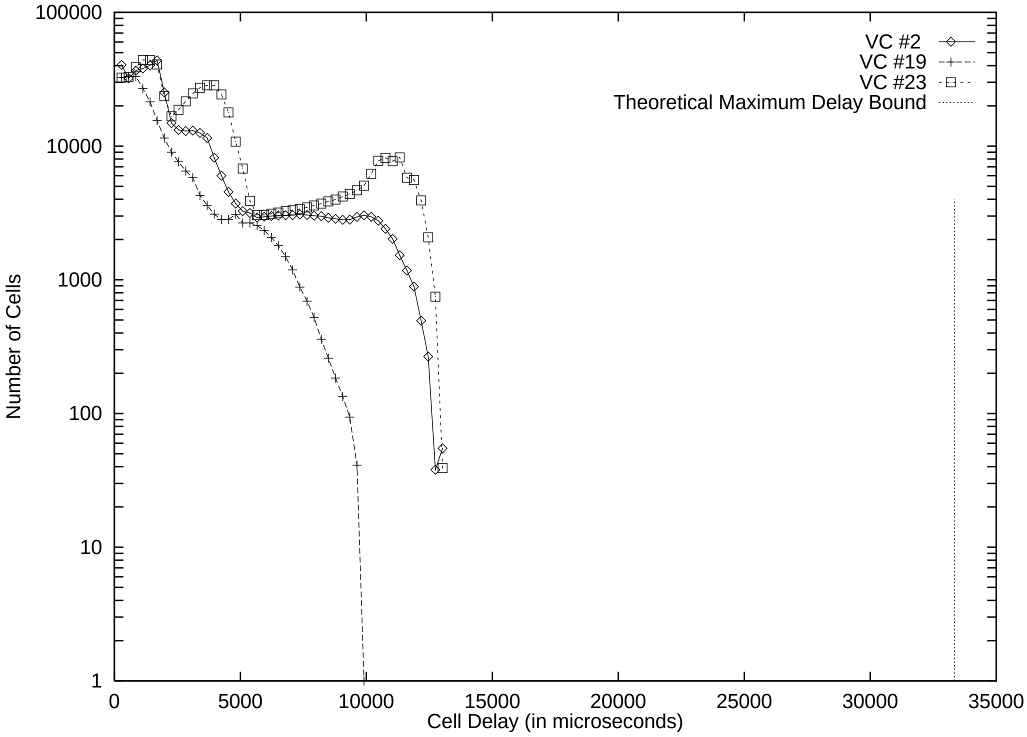

Figure 17: Cell delays with 25 MPEG video sources

References

[1] N. Agarwal and I. Viniotis, "Percentile Goal as a QOS Criterion: Part I Analysis," submitted to *IEEE/ACM Trans. Networking*, 1999.

[2] H. Jonathan Chao, Hsiling Cheng, Yau-Ren Jenq, and Daein Jeong. Design of a generalized priority queue manager for ATM switches. *IEEE Journal on Selected Areas in Communications*, 15(5):867–879, June 1997.

[3] K. Elsayed and H. Perros, "A Comparative Performance Analysis of Call Admission Control Schemes in ATM Networks." To appear in the *Journal of Computer Networks and ISDN Systems.*

[4] V. Firoiu, J. Kurose, and D. Towsley, "Efficient Admission Control for EDF Schedulers." In *Proceedings of IEEE INFOCOM '97,* pp. 310-317, 1997.

[5] L. Georgiadis, R. Guérin, and A. Parekh. "Optimal multiplexing on a single link: delay and buffer requirements,". *IEEE Transactions on Information Theory*, vol. 43 no. 15, pp. 1518–1535, Sep. 1997.

[6] L. Georgiadis, R. Guerin and V. Peris, "Efficient network QoS provisioning based on per node traffic shaping," *IEEE/ACM Trans. Networking*, vol. 4, pp. 482-501, 1996.

[7] P. Goyal and H.M. Vin. "Generalized guarantee rate scheduling algorithms: a framework," *IEEE/ACM Transactions on Networking*, vol. 5, no 4, pp. 561–571, Aug. 1997.

[8] Jörg Liebeherr, Dallas E. Wrege, and Domenico Ferrari. Exact admission control for networks with bounded delay service. *IEEE/ACM Transactions on Networking*, 4(6):885–901, December 1996.

[9] R. Nagarajan, J. Kurose, D. Towsley, and M.G. Hluchyj, "On per-session end-to-end delay distributions and the call admission problem for real-time applications with QOS requirements," *Proceeidngs of SIGCOMM '93,* pp. 2-12, Sep. 1993.

[10] A. Parekh and R. Gallager, "A Generalized Processor Sharing Approach to Flow Control - the Multiple Node Case," *IEEE/ACM Trans. Networking*, vol. 2, pp. 137-150, 1994.

[11] H. Schulzrinne, J. Kurose, and D. Towsley. "An evaluation of scheduling mechanisms for providing best-effort, real-time communication in wide-area networks," *Proceedings of INFOCOM '94*, pp 1352–1361, 1994.

[12] S. Wright and I. Viniotis, "Simulation Study of Maximal Delays in Various Link Scheduling Algorithms," *Proceedings of IEEE Globecom '98*, Australia.

[13] S. Wright and I. Viniotis, "Delay Bounds and Connection Admission Control for Burstiness Class based Queueing", Ph. D. thesis, North Carolina State University, 1999.

[14] Hui Zhang and E. Knightly. Providing end-to-end statistical performance guarantees with bounding interval dependent stochastic models. *Performance Evaluation Review*, 22(1):211–220, May 1994.

[15] H. Zhang and E. Knightly, "Providing end-to-end statistical performance guarantees with interval dependent stochastic models," *Proceedings of ACM Sigmetrics '94*, pp. 211-220, 1994.

[16] Z. Zhang, D. Towsley and J. Kurose, "Statistical Network Performance Guarantees with Generalized Processor Sharing Scheduling," *IEEE J. Select. Areas. Commun.*, vol. 13, pp. 1071-1080, 1995.