Earliest Due First Scheduling for Application-Level QoS Delivery*

Zoe Antoniou   Ioannis Stavrakakis

Department of Electrical & Computer Engineering
Northeastern University
Boston, MA 02115, USA

Abstract

The Quality of Service (QoS) of real time applications is affected in a rather direct manner by the *timely delivery* of *application-level data units* which are typically mapped into several ATM cells. As a consequence, it would be reasonable to associate a *common* deadline with all the cells in one data unit (batch) and develop a scheduling policy that aims to maximise the number of data unit that are delivered by their respective deadlines.

A batch-based Earliest Due First scheduling policy is proposed and analysed in this paper under periodic batch arrival processes which are typical of continuous media streams. Depending on whether partial batch service is performed or not, two variations of this scheduler are considered. It is shown that the proposed policies can be easily implemented in terms of a First Batch In - First Batch Out (FBI-FBO) and a Partial FBI-FBO schedulers respectively. Numerical results are presented and the induced batch and cell loss rates are compared against those induced by the standard FIFO scheduler as well as a FIFO scheduler which drops expired cells.

Please send correspondence to:

Ioannis Stavrakakis, ECE Dept. - 409 Dana Bldg., Northeastern University, Boston, MA 02115. (E-mail: ioannis@cdsp.neu.edu)

*Research supported in part by the Advanced Research Project Agency under Grant F49620-93-1-0564 monitored by the Air Force Office of Scientific Research (AFOSR) and the National Science Foundation under Grant NCR 9628116.

1 Introduction

Asynchronous Transfer Mode (ATM) technology is expected to support applications with a wide range of service requirements and traffic characteristics. Real time applications, such as voice and video are particularly demanding since such applications require stringent guarantees on the Quality-of-Service (QoS) in terms of bounded delay, loss and jitter. The traffic characteristics of real time applications can be very diverse as they combine a variety of media data such as text, audio, graphics, images and full motion video. The transmission of continuous media traffic is an inherently difficult problem due to the time sensitive nature and the traffic variability of these applications. [1] [6] [8]

Consider, for instance, the transmission of video applications in which it is required that frames (images) are transmitted at a certain rate in order to guarantee no starvation or overflow at the receiving end. Each frame would be of variable size (in bits) according to its complexity and the coding scheme employed. The frame could be considered as an application data unit which is central to both the traffic generation process at the transmitter and the reconstruction process at the receiver. Such a role may be played by a smaller encoding unit than the frame such as a macroblock. [7]

Application data units, such as the frame, are organised into small fixed size and standardised packets, the cells, in order to be transported over an ATM network. Typically, an application data unit is mapped into several ATM cells. When network resources are reserved at peak demand, all cells of the application data unit will be delivered on time, leading to a timely and error-free reconstruction at the receiver. When resources are over-allocated to a number of applications - to increase network utilisation- all cells of an application data unit may not be delivered before this application data unit is required by the reconstruction process. In this case, application data units may be significantly distorted due to missing cells or be impossible to process at all. The required QoS for such applications may then be defined in terms of a metric associated with the correctly and timely delivered application data units. An application-level QoS metric such as the application data unit loss rate would describe more accurately the QoS delivered to the end user, than a network-level QoS metric such as the cell loss rate. [2]

In order to achieve QoS delivery to real time applications while efficiently utilising the network resources it is necessary that allocation of the resources (scheduling) be based on the applications' QoS requirement. For instance, the timely delivery of information associated with real time applications suggests that deadlines could be defined and deadline-driven scheduling policies be designed. If the nature of an application requires that a complete application data unit should be received before it can be processed by the receiver, then it would be reasonable to associate a **common** deadline with all the cells in the same data unit and develop scheduling policies that aim to maximise the number of data unit that are delivered by their respective due date. This approach is different from many traditional approaches in which every cell is assigned its own deadline and the scheduling policy typically attempts to maximise the number of individual cells transmitted by their due date.

In this paper a scheduling policy which would be appropriate for supporting applications such as those described above is introduced. This policy attempts to maximise the number of application data units -referred to as batches in the rest of the paper- served before their deadlines.

Unlike the typical cell-based Earliest Due First (EDF) scheduling policy which requires sorting on a cell-by-cell basis [4], the proposed policy requires such sorting on a batch-by-batch basis, reducing the processing burden. For continuous media applications in which batches (application data units) are delivered over well defined periods the processing burden of the proposed policy can be substantially reduced by ordering the multiplexed data streams according to their respective deadlines and serving them according to the First Batch In - First Batch Out (FBI-FBO) policy, as it will become clear in later sections. The complexity of this policy is similar to that of a (cell-based) FIFO policy applied to the deadline ordered streams. The performance of the FBI-FBO and the FIFO policies is evaluated in terms of the induced cell loss and batch loss probabilities. In view of the earlier discussion the batch loss probability may be a more relevant metric for applications in which higher layer protocol data units are fragmented into a number of ATM cells; these data units may be considered to be lost if any one of their cells are lost, unless some redundancy is introduced in the data unit to improve the probability of correct reception at the expense of increasing the traffic in the network.

2 System Description

This section describes the system and the traffic environment considered in this work. A network node with N incoming and one outgoing links is considered. All links are assumed to have the same capacity and, thus, a cell requires a fixed amount of time for transmission, referred to as a slot. All arrivals and departures from the node occur at slot boundaries. Let J denote an incoming data stream (application); J takes values in the range $0 \leq j \leq N-1$. A variable with superscript j represents a quantity associated with stream j.

The incoming traffic is assumed to be bursty periodic. It consists of variable size bursts of cells arriving at fixed time intervals (periods); all applications are assumed to have the same period, T.

This type of traffic is characteristic of many continuous media applications such as digital audio and video. It is further assumed that the bursts of stream j arrive over consecutive time slots, that is in the form of batches, and that their length, A^j, follows a general distribution with expected value $\overline{A^j}$. All batches of the same application j are assumed to have the same deadline,

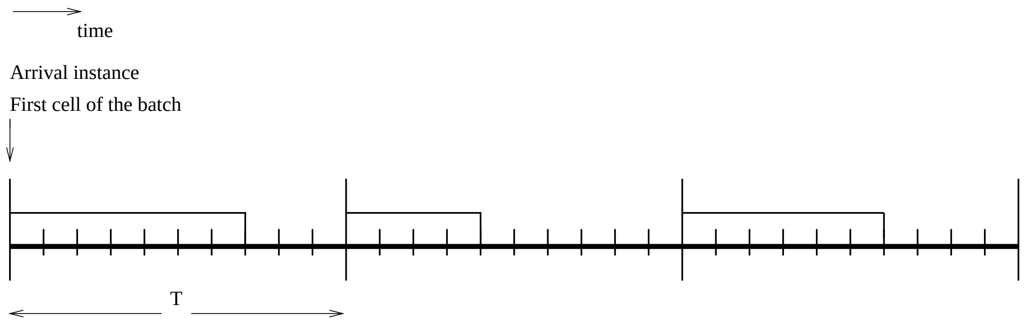

Figure 1: An example batch arrival pattern.

D^j, which is provided to the server upon connection set up. The *deadline* determines the maximum time (measured in cell slots) which the batch may remain in the node once the header of the batch has arrived. This includes both the delay and the transmission delay.

Figure 2 depicts the instances at which batch arrivals occur for $N = 4$ streams. Let $\{t_k\}_{k \geq 0}$ denote the sequence of batch arrival instances (from any stream); t_k indicates the arrival instance of the k^{th} batch. Variables with subscript k will represent the value of the associated quantity at time t_k. Let τ_k denote the interval between two consecutive batch arrivals (from any stream); that is :

$$\tau_k = t_{k+1} - t_k \qquad \text{with} \tag{1}$$

$\tau_k = \tau^j \quad \text{if} \quad J_k = j$, where τ^j denotes the interarrival time between two consecutive batches from streams j and $(j+1)mod(N)$, or equivalently, the distance between two consecutive batch arrival instances of streams j and $(j+1)mod(N)$. In the rest of this paper it is assumed that the streams are ordered in terms of increasing deadlines. That is, $D^j \leq D^{j+1}$ for $0 \leq j \leq N-2$.

As it was stated earlier, the goal of the proposed policy is the timely delivery of complete batches. This goal suggests that the scheduler should serve batches based on the Earliest Due First (EDF) policy. In view of the deadline ordering and the periodicity of the batch arrival process it is easy to establish that a scheduler that serves *batches* according to the *First In First Out* policy does serve them according to the EDF policy. This proposed policy is referred to as the First Batch In - First Batch Out (FBI-FBO) policy and is discussed in more detail below.

2.1 The FBI-FBO Scheduling Policy

Implementation of this policy requires that information regarding the length (in cells) of a batch be available to the scheduler upon batch arrival. This will be the case if the batch size is fixed and

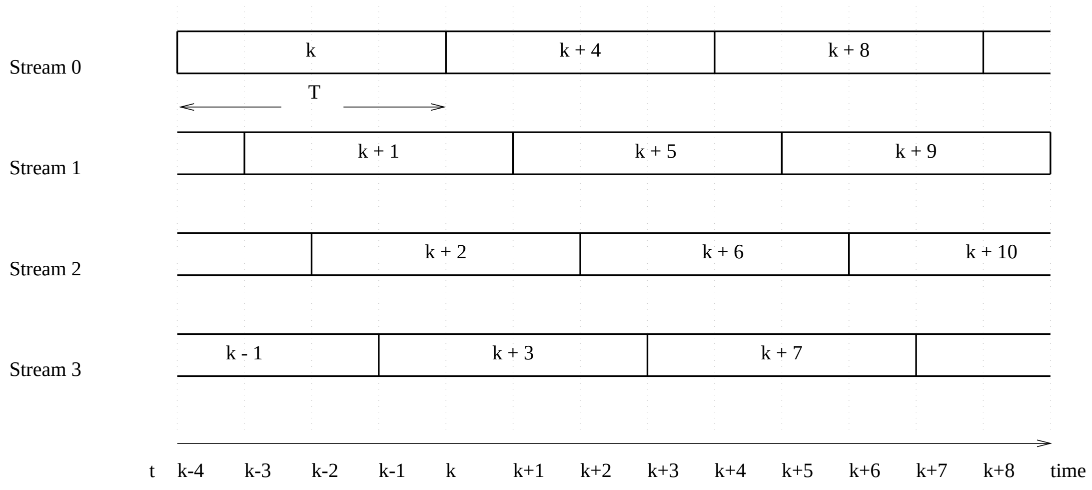

Figure 2: Periodic arrival pattern with shifted relative arrival instances for N = 4 traffic streams.

provided to the scheduler upon connection set up or if this information is carried by the first cell of the batch or by dedicated cells such as Resource Management Cells.

Upon Batch arrival the scheduler decides whether this batch can be served by its deadline or not. In the former case, the batch is served before its deadline expires as described below. In the latter case, the *entire* batch is discarded.

The scheduler is equipped with a *service queue* to which cells are shifted for storage before they are transmitted. All cells shifted to the service queue will be transmitted, following the first cell in - first cell out policy. Suppose that a batch from stream j arrives at time t_k. If the sum of the batch length A^j and the service queue occupancy Q does not exceed the batch deadline, D^j, a reservation is made in the service queue for a space of A^j slots. The service queue occupancy is then registered as being $(Q + A^j)$, even though only the first cell has arrived so far. The remaining $(A^j - 1)$ cells of the batch will be arriving over the next $(A^j - 1)$ time slots and they will be shifted to the reserved space in the service queue immediately upon arrival. In other words, data stream j 'sees' a logical service queue of capacity D^j and decides to initiate a batch shifting to the service queue only if the following condition is met upon arrival:

$$Q + A^j \leq D^j \tag{2}$$

This condition guarantees that the cells will be sent to the output link within D^j time slots and, thus, they will meet their deadline. If the associated logical queue overflows, then the dropped cells will be the ones to miss the batch deadline. In this case no cell of the batch is shifted to the service queue.

It should be noted that the server attends to a newly arrived batch once upon the arrival of the header, determines whether it will meet its deadline or not and decides whether to serve it or drop it. Once a scheduling decision is made, it is no longer necessary to keep a record of the batch arrival, its length or its relative deadline. The above scheduling policy is *work conserving* in the sense that the server will not remain idle as long as there are batches waiting to be served. If the service queue becomes empty, the server will become idle and will await for the next batch arrival.

2.2 The Partial FBI-FBO Scheduling Policy

In an environment in which complete batches need to be transmitted before their deadline in order to be useful, a scheduling policy that drops expired cells is more effective than the one which serves expired cells, in the sense that a larger fraction of batches can be transmitted before their deadlines. In order for the scheduler to determine whether an entire batch will meet its deadline or not, information about the length of the batch must be provided upon batch arrival. This is the assumption made in the FBI-FBO policy described above.

If the batch length information is not provided upon batch arrival the scheduler would attend to each cell of the batch individually by checking the service queue occupancy against the remaining time till the deadline expiration on a cell-by-cell basis. In this case, though, the scheduler may serve part of a batch and drop the remaining cells which will not meet the deadline. The entire batch would then be lost. The variation of the FBI-FBO policy -referred to as Partial FBI-FBO policy- is considered in this paper as well. This is similar to the FBI-FBO policy with the difference that the server serves the fraction of the batch that can meet its deadline. Clearly, a smaller number of complete batches and a higher number of cells would be served under the Partial FBI-FBO policy.

3 Performance Analysis

In this section a more precise notation is introduced and the derivation from first principles of the equations that mathematically describe the system is presented.

Let J_k denote the data stream from which a new batch arrival occurs at time t_k. The content of the new arrival is to be shifted to the service queue beginning at time t_k. Let Q_k denote the service queue occupancy at time t_k, *after* it has been updated by the number of cells in the new batch from stream J_k. In view of the above, the system process $\{Q_k, J_k\}_{k\geq 0}$ can be defined as a 2-D process with state space $\{(q_k, j_k) : 0 \leq q_k \leq D_{max}, 0 \leq j_k \leq N-1\}$. D_{max} denotes the maximum deadline of all N stream deadlines; that is $D_{max} = \max_j\{D^j\}$. Note that for J_k the only transitions allowed are from some j_k to $j_{k+1} = j_k mod(N)$. Based on the above definition of the system process, it is easy to establish that it is a Markov chain. This can be established by noting that J_k is a periodic random variable and that the next service queue occupancy, Q_{k+1}, can be probabilistically determined from the current one, Q_k, and the new batch arrival, $A^{j_{k+1}}$; A^{j_k} denotes the random variable representing the number of cells from stream j_k that arrived at time t_k. The remaining of this section contains the derivation of the probability that the Markov chain moves from any state (q_k, j_k) to state (q_{k+1}, j_{k+1}), given by $P(q_k, j_k, q_{k+1}, j_{k+1})$.

3.1 FBI-FBO Policy

Under this policy, cells of a batch are served only if the *entire* batch can meet its deadline. Three distinct cases can be identified, based on the queue occupancy of the service buffer at the instant of a new batch arrival. In each of the following cases $0 \le j_k \le N-1$ and $j_{k+1} = j_k mod(N)$.

Case A: for $0 \le q_k \le \tau_k$. In this case all the cells in the buffer will have been served by the time the next batch arrives from stream j_{k+1} at time t_{k+1} and the new buffer occupancy will be equal to the number of new arrivals only:

$$P(q_k, j_k, q_{k+1}, j_{k+1}) = P\{A^{j_{k+1}} = q_{k+1}\} \quad \text{for} \quad 0 \le q_{k+1} \le T \tag{3}$$

Case B: for $\tau_k < q_k \le D^{j_{k+1}} + \tau_k - 1$. In this case by the time the next batch arrives the service buffer occupancy will have dropped to $(q_k - \tau_k)$. Hence, with the new arrival $A^{j_{k+1}}$ the occupancy will be updated to $(q_k - \tau_k + A^{j_{k+1}})$ if the entire batch can meet its deadline $D^{j_{k+1}}$:

$$P(q_k, j_k, q_{k+1}, j_{k+1}) = P\{A^{j_{k+1}} = q_{k+1} - (q_k - \tau_k)\} \quad \text{for} \quad (q_k - \tau_k) \le q_{k+1} \le D^{j_{k+1}} \tag{4}$$

Otherwise, the new batch will be dropped and the service buffer occupancy will become $(q_k - \tau_k)$:

$$P(q_k, j_k, q_{k+1}, j_{k+1}) = \sum_{n=D^{j_{k+1}}-(q_k-\tau_k)+1}^{T} P\{A^{j_{k+1}} = n\} \quad \text{for} \quad q_{k+1} = q_k - \tau_k \tag{5}$$

Case C: for $D^{j_{k+1}} + \tau_k \le q_k \le D_{max}$. In this last case the new batch will definitely miss its deadline and thus it is dropped with probability 1:

$$P(q_k, j_k, q_{k+1}, j_{k+1}) = 1 \quad \text{for} \quad q_{k+1} = q_k - \tau_k \tag{6}$$

3.2 Partial FBI-FBO Policy

This policy allows for a batch to be either partially or fully served. In the event where only a fraction of the batch can meet its deadline those cells are shifted to the service buffer and the remaining are dropped. Again three distinct cases can be identified. Cases A and C are the same as in the above section.

Case B: for $\tau_k \leq q_k \leq D_{j_{k+1}} + \tau_k - 1$. In this case depending on the length of the new batch there are two distinct possibilities of full or partial service:
Full service:

$$P(q_k, j_k, q_{k+1}, j_{k+1}) = P\{A^{j_{k+1}} = q_{k+1} - (q_k - \tau_k)\} \quad \text{for} \quad (q_k - \tau_k) \leq q_{k+1} \leq D^{j_{k+1}} - 1 \tag{7}$$

Partial service:

$$P(q_k, j_k, q_{k+1}, j_{k+1}) = \sum_{n=D^{j_{k+1}}-(q_k-\tau_k)}^{T} P\{A^{j_{k+1}} = n\} \quad \text{for} \quad q_{k+1} = D^{j_{k+1}} \tag{8}$$

4 Computation of Batch Loss and Cell Loss probabilities

Apart from the *cell* loss probability that can be used in order to assess the performance of the proposed policies, another meaningful metric would be the *batch* loss probability. A batch is considered to be lost unless all its cells are served. The batch or cell loss probabilities are easily computed by involving the Markov system process.

4.1 Batch Loss Probability

Since a batch is considered lost if one of its cells is lost, the analysis for the batch loss probability is the same for both policies.

Let $L^B(q_k, j_{k+1})$ denotes the number of batches that are dropped over the interval $[t_{k+1}, t_{k+2})$, where j_{k+1} is the data stream the scheduler will attend to over this interval (following stream j_k) and q_k is the queue occupancy at t_k. Clearly:

$$L^B(q_k, j_{k+1}) = \mathbf{1}_{\{A^{j_{k+1}} > D^{j_{k+1}} - (q_k - \tau_k)\}} \tag{9}$$

The conditional expected value of this indicator function given that the system is in state (q_k, j_{k+1}) is given by:

$$\overline{L^B}(q_k, j_{k+1}) = E\{\mathbf{1}_{\{A^{j_{k+1}} > D^{j_{k+1}} - (q_k - \tau_k)\}}\} \tag{10}$$

It represents the average number of batches lost as a function of the queue occupancy at the time instant t_k and the data stream arrival that occurs at t_{k+1} under both policies.

If $0 \leq q_k \leq \tau_k$ then clearly there will be no losses since the server will serve all the cells in the service buffer during the interarrival time τ_k between streams j_k and j_{k+1} and thus:

$$L^B(q_k, j_{k+1}) = 0 \tag{11}$$

$$\overline{L^B}(q_k, j_{k+1}) = 0 \tag{12}$$

If $\tau_k \leq q_k \leq D_{max}$ then losses may occur. The conditional expected value of those losses in any state (q_k, j_{k+1}) is given by:

$$\overline{L^B}(q_k, j_{k+1}) = \sum_{n=D^{j_{k+1}}-(q_k-\tau_k)+1}^{T} P\{A^{j_{k+1}} = n\} \tag{13}$$

Hence, the overall average value of the batch losses L^B over any interval τ_k will be:

$$\overline{L^B} = \sum_{j=0}^{N-1} \sum_{q_k=\tau_k+1}^{D_{max}} \sum_{n=D^{j_{k+1}}-(q_k-\tau_k)+1}^{T} P\{A^{j_{k+1}} = n\}\pi(q_k, j_k), \tag{14}$$

where $\pi(q_k, j_k)$ is the steady state probability distribution of the corresponding Markov Chain.

Finally, the batch loss probability is given by:

$$P_B = \frac{E\{\text{batches lost over any interval } \tau_k\}}{E\{\text{batches arrived over any interval } \tau_k\}} = \frac{\overline{L^B}}{1} = \overline{L^B} \tag{15}$$

The denominator is equal to 1 since by definition of the system, there is only one batch arrival over any interarrival interval τ_k.

4.2 Cell Loss Probability

The derivation of the cell loss probabilities is similar for both policies and follows the same procedure as in the above section. Under the FBI-FBO policy, the cell losses correspond to the cells of the batches which did not meet the deadline. Whereas under the Partial FBI-FBO policy the lost cells can be a fraction of the batch's total number of cells.

Again let $L^C(q_k, j_{k+1})$ denote the number of cells dropped over the interval $[t_{k+1}, t_{k+2})$, where j_{k+1} is the data stream the scheduler will attend to over this interval and q_k is the queue occupancy at t_k. Also, let $\overline{L^C}(q_k, j_{k+1})$ denote the conditional expected value given that the system is in state (q_k, j_{k+1}).

As in the previous section, if $0 \leq q_k \leq \tau_k$ then clearly there will be no losses since the server will serve all the cells in the service buffer during the interarrival time τ_k between streams j_k and j_{k+1} and thus:

$$L^C(q_k, j_{k+1}) = 0 \tag{16}$$

$$\overline{L^C}(q_k, j_{k+1}) = 0 \tag{17}$$

for both policies.

If $\tau_k \leq q_k \leq D_{max}$ then losses may occur. For the FBI-FBO policy the conditional average number of cells lost over any interval τ_k is given by:

$$\overline{L_1^C}(q_k, j_{k+1}) = \sum_{n=D^{j_{k+1}}-(q_k-\tau_k)+1}^{T} nP\{A^{j_{k+1}} = n\} \tag{18}$$

The same quantity for the Partial FBI-FBO policy is given by:

$$\overline{L_2^C}(q_k, j_{k+1}) = \sum_{n=D^{j_{k+1}}-(q_k-\tau_k)+1}^{T} [n - (D_{j_{k+1}} - (q_k - \tau_k))] P\{A^{j_{k+1}} = n\} \tag{19}$$

The overall average values of $L_1^C(q_k, j_{k+1})$ and $L_2^C(q_k, j_{k+1})$ is given by:

$$\overline{L_1^C}(q_k, j_{k+1}) = \sum_{j=0}^{N-1} \sum_{q_k=\tau_k+1}^{D_{max}} \sum_{n=D^{j_{k+1}}-(q_k-\tau_k)+1}^{T} nP\{A^{j_{k+1}} = n\}\pi(q_k, j_k) \tag{20}$$

$$\overline{L_2^C}(q_k, j_{k+1}) = \sum_{j=0}^{N-1} \sum_{q_k=\tau_k+1}^{D_{max}} \sum_{n=D^{j_{k+1}}-(q_k-\tau_k)+1}^{T} P\{A^{j_{k+1}} = n\}[n-(D^{j_{k+1}}-(q_k-\tau_k))]\pi(q_k, j_k), \tag{21}$$

where $\pi(q_k, j_k)$ is the steady state probability distribution of the corresponding Markov Chain.

Finally, the cell loss probability is given by:

$$P_C = \frac{E\{\text{cells lost over any interval } \tau_k\}}{E\{\text{cells arrived over any interval } \tau_k\}} \tag{22}$$

The numerator is given by Equations (20) and (21). The denominator is equal to the weighted sum of the mean batch lengths of all the data streams in the system over any interval τ_k:

$$\overline{A} = \sum_{j=0}^{N-1} \frac{1}{N} \overline{A^j} \tag{23}$$

5 Numerical Results and Discussion

In this section, the performance of the proposed policies is compared to that of the FIFO policy which serves the deadline ordered streams on a cell-by-cell basis. Comparison is based on numerical results for the batch loss and cell probabilities induced by these policies.

The induced losses under the proposed policies have been computed using equations (14), (15), (20), (21), (22) and (23) while those under the FIFO policy have been computed through simulations. For each simulation run 10^7 batch instances have been generated from each stream and the average of two such simulation runs has been obtained.

Two versions of the FIFO policy have been considered: FIFO policies with and without an expired cell discarding mechanism. The first one assumes no knowledge of the deadlines and simply serves the cells on a first-come first-served basis. This is referred to as the pure FIFO approach in this paper and it is simple and straightforward in its implementation. The proposed policy is compared to the pure FIFO in order to evaluate their relative performances and to determine if a significant gain can be achieved by employing a more involved scheduling policy. The second FIFO policy that is considered has an expired cell discarding mechanism. It identifies and drops those cells that can not meet their deadline. In order for the expired cells to be dropped they typically need to be searched for. The computational overhead in this case is quite significant for a large number of streams. Another drawback of the FIFO approach is that cells from different batches are multiplexed at the service queue. This causes the shape of the data traffic to change and one information unit spreads over a longer time interval in the output link as compared to its original length.

5.1 Homogeneous Traffic

Initially a homogeneous traffic environment is considered whereby the traffic from all data streams is statistically identical and has the same performance requirement (that is common delay tolerance). The batch lengths are geometrically distributed. The interarrival interval τ_k is constant and equal to $\tau = \frac{T}{N}$ (τ is always rounded to the nearest integer). The system utilisation can be varied in proportion to the average batch length and the period T.

Figure 3 shows the batch loss probability under the FBI-FBO policy (referred to here as Policy 1), the Partial FBI-FBO policy (referred to here as Policy 2) and the FIFO with the expired cell discarding mechanism as a function of the system utilisation for $N = 4$, $T = 40$ and identical deadlines $D^j = D$. The following deadlines have been considered: $D = 40$ (solid line) , $D = 60$ (dotted line) and $D = 80$ (dashed line). In other words, the deadline has been set to 1, $1\frac{1}{2}$ and 2 frames (periods). Note that as the deadlines increase the losses decrease as was expected.

Figure 4 shows how the batch loss probability changes as a function of the offered load for all four policies for $N = 4$ and $T = 40$. The results are for deadlines equal to 1 and $1\frac{1}{2}$ frames. The top solid line corresponds to the FIFO without the expired cell discarding mechanism, the dashed line corresponds to the FIFO with expired cell discarding, the dotted line corresponds to Policy 2 and the bottom solid line is Policy 1. Among the four policies, Policy 1 provides the lowest batch loss rate. For low system utilisation it performs better than the FIFO approaches by approximately an order of magnitude. As the system utilisation increases Policy 1, Policy 2 and the FIFO with the expired cell discarding mechanism tend to converge to a common batch loss rate whereas the losses of the pure FIFO increase at a faster rate.

Similar observations can be made when the number of data streams is increased to $N = 8$. Figure 5 shows the results for deadlines equal to 1 and $1\frac{1}{2}$ frames and $T = 64$. Again the policies at the two extreme ends are Policy 1 and the pure FIFO. Policy 1 provides the lowest batch loss rate among the four and outperforms the pure FIFO by at least one order of magnitude. Policy 2 and the FIFO with the expired cell discarding mechanism tend to converge for high system utilisations.

So far, only the batch loss rates have been considered. Figure 6 shows the cell loss rates for the cases considered in Figure 5. Notice that Policy 2 (dotted line) performs better than Policy

1 (lower solid line) as expected. However, the performance of Policy 1 is the same or better than that of the pure FIFO (upper solid line).

From the above results it can be clearly concluded that Policy 1 has the lowest batch loss rate while at the same time it keeps the cell loss rate at the same or lower level than that of the pure FIFO approach. The performances of Policy 2 and of the FIFO with the expired cell discarding mechanism are generally in the middle with higher batch loss rates and slightly lower or equal cell loss rates than Policy 1.

Increasing the deadline of the input traffic to 2 frames or more (not shown in this paper) resulted in about the same performance under all policies. In other words, if the deadlines are long enough the losses become independent of the particular scheduling policy and they depend on the traffic load alone.

Figures 7 and 8 present the results under all four policies under a batch size distribution delivering a batch size of length L cells with probability p and length 1 with probability $1-p$ ($1-L$ bursty periodic traffic model). The system parameters are: $N = 8$, $T = 64$, $L = 36$, $p = 0.2$ and deadlines equal to 1 and $1\frac{1}{2}$ frames.

Figure 7 shows the batch loss probability as a function of the offered load. Once again Policy 1 achieves the lowest batch loss rate whereas the pure FIFO policy has the highest. Notice, however, that the losses spread over mostly three orders of magnitude as opposed to five or six orders under geometrically distributed batches.

The cell loss probability is shown in Figure 8. As it can be seen, Policy 1 induces equal or lower cell loss rate than the pure FIFO approach, as before. Again the losses spread over mostly three orders of magnitude as opposed to five or six orders under geometrically distributed batches.

5.2 Inhomogeneous Traffic

An inhomogeneous traffic environment has also been considered. The objective in this case is to evaluate the level of diversity in the induced performance. Out of the $N = 8$ data streams, two are assumed to deliver probabilistically larger batches and since the period is the same for all streams, have larger load. τ is maintained constant for all streams. The six (background) streams are assumed to deliver batches of size 1 or L with parameter p (as before) with $L = 10$ and $p = 0.2$. The two (more active streams) streams have parameters L and $p = 0.4$, where L assumes a value between 8 and 56 so that a specific system utilisation is achieved.

Figure 9 shows the batch loss rates for the two more active streams (data streams 1 and 4) and the six background streams under all four policies. It can be observed that Policy 1 induces the lowest losses for the two more active streams which are of the order of $10^{-2} - 10^{-3}$ and are observed only for a system load higher than 0.7. The other policies induce higher losses for a wider range of system utilisation. The two FIFO policies induce approximately the same losses which are higher than those under Policy 1 by at least an order of magnitude. For the six background streams Policy 1 induces a loss rate of the order of $10^{-3} - 10^{-4}$ for system utilisation higher than 0.7 only.

To further reduce the batch losses an attempt was made to adjust the streams' interarrival time τ in proportion to the *expected batch length* of each stream. Increasing the values of τ for streams with high mean batch length could result in a reduction of the average occupancy of the service buffer since 'busy' streams will now have more time to be served before the next batch arrival occurs. The value of τ has been set using the following equation:

$$\tau^j = \frac{\overline{A_T^j}}{\sum_{j=0}^{N-1} \overline{A_T^j}} T \tag{24}$$

and is rounded to the nearest integer as before.

Figure 10 shows the batch losses for Policy 1 with equal values of τ (solid line) and with weighted values of τ (dashed line). It can be noted that adjusting the value of τ hardly affects the loss rate for the two more active streams. In fact, there is a slight increase. On the other hand, the loss rate for the six background has been improved significantly. Losses are of the order of 10^{-5} only for system utilisations higher than 0.9.

The more active streams have longer batches as compared to other streams. Long batches will reserve more space in the service buffer once they arrive and are accepted by the system. If the interarrival interval τ is the same for all streams then shorter batches from less busy streams may have to be dropped more frequently. However, when the value of τ is adjusted in proportion to the average batch length of each stream, there will be more time for the service of longer batches thus improving on the overall throughput. This is clearly observed in the results that are presented above.

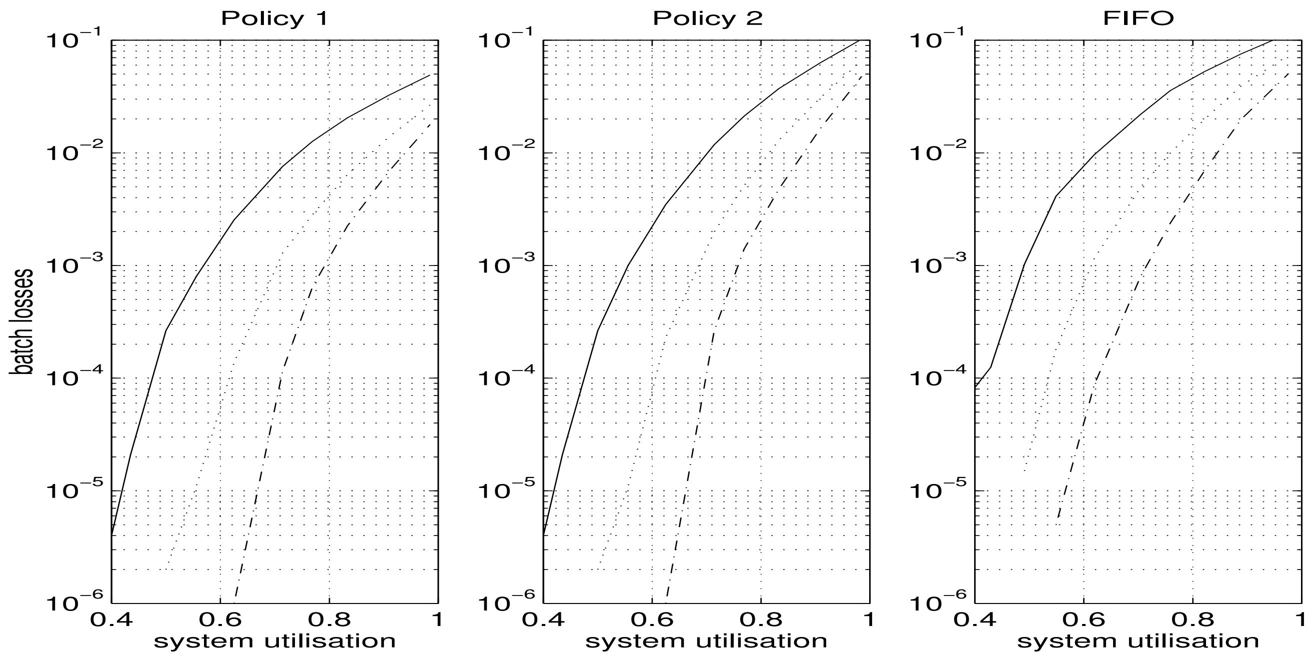

Figure 3: Batch loss curves for N=4 streams, period T=40 and deadlines D=40, 60, 80 (from left to right) for HOMOGENEOUS periodic traffic with GEOMETRIC distribution. The losses decrease as a function of increasing deadlines as expected.

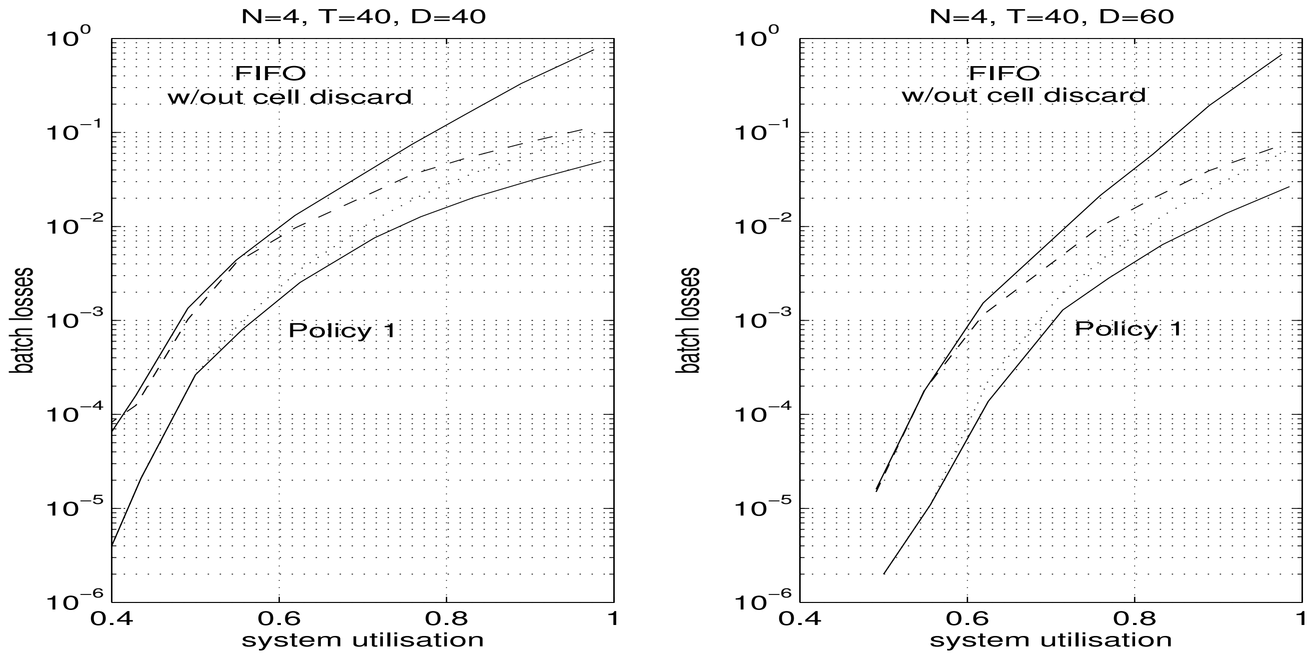

Figure 4: Batch loss curves for N=4 streams, period T=40 and deadlines D=40 and 60 (GEOMETRIC distribution). Top solid line is FIFO without expired cell discarding, dashed line is FIFO with expired cell discarding, dotted line is Policy 2 and bottom solid line is Policy 1.

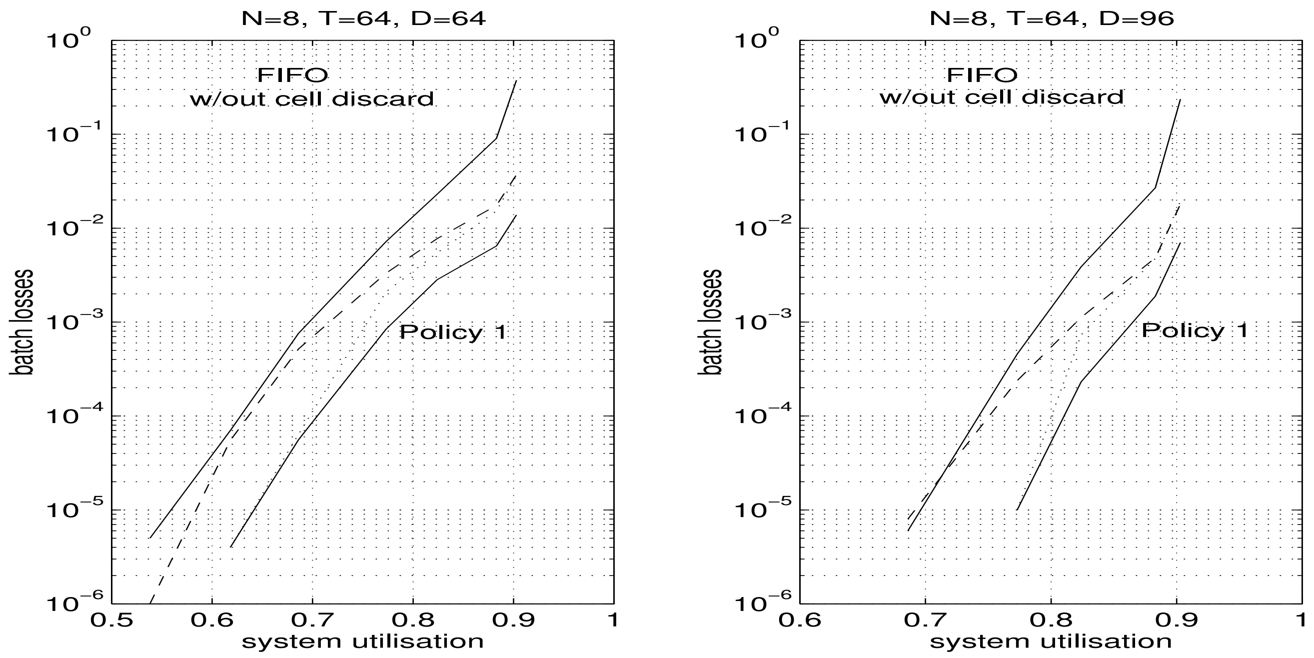

Figure 5: Batch loss curves for N=8 streams, period T=64 and deadlines D=64 and 96 (GEOMETRIC distribution). Top solid line is FIFO without expired cell discarding, dashed line is FIFO with expired cell discarding, dotted line is Policy 2 and bottom solid line is Policy 1.

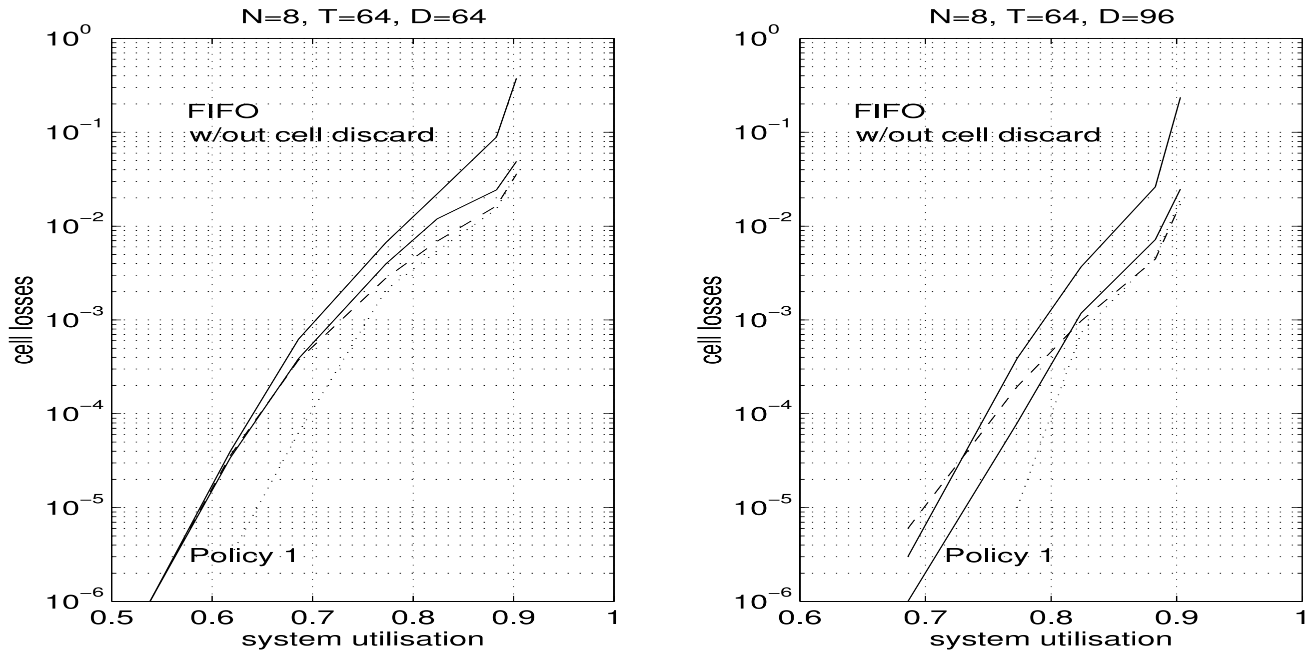

Figure 6: Cell loss curves for N=8 streams, period T=64 and deadlines D=64 and 96 (GEOMETRIC distribution). Top solid line is FIFO without expired cell discarding, dashed line is FIFO with expired cell discarding, dotted line is Policy 2 and bottom solid line is Policy 1.

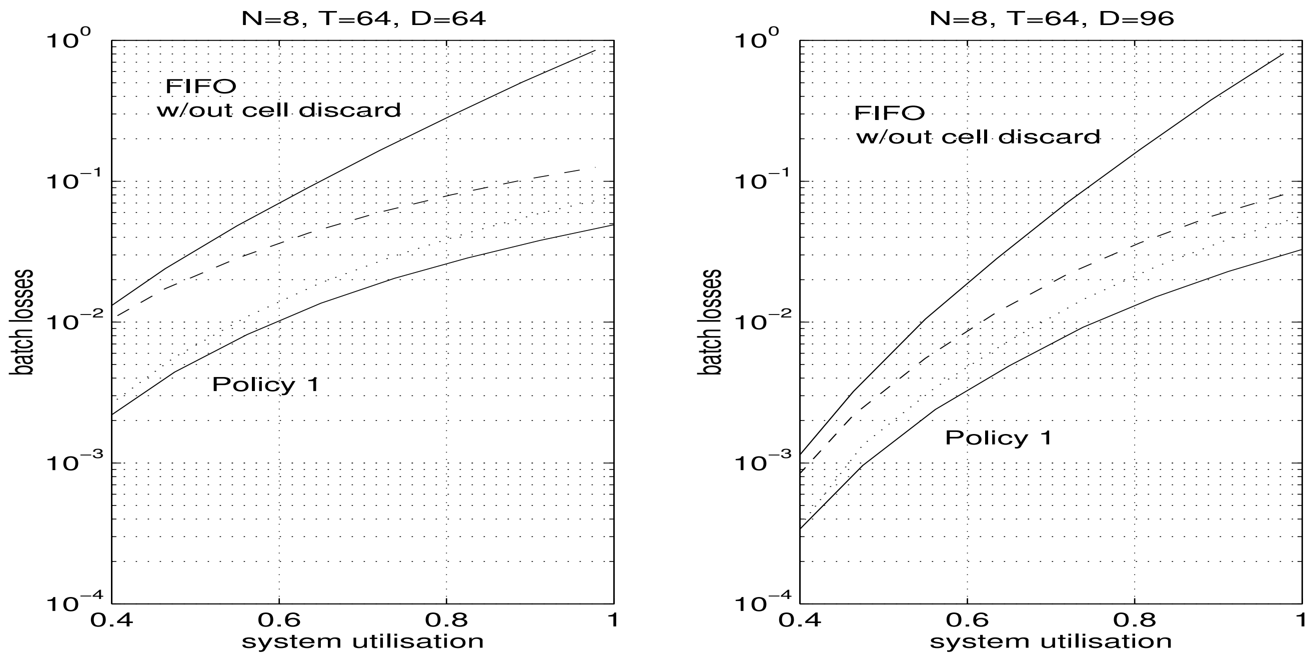

Figure 7: Batch loss curves for N=8 streams, period T=64 and deadlines D=64 and 96 for BURSTY 1-L periodic traffic. Top solid line is FIFO without expired cell discarding, dashed line is FIFO with expired cell discarding, dotted line is Policy 2 and bottom solid line is Policy 1.

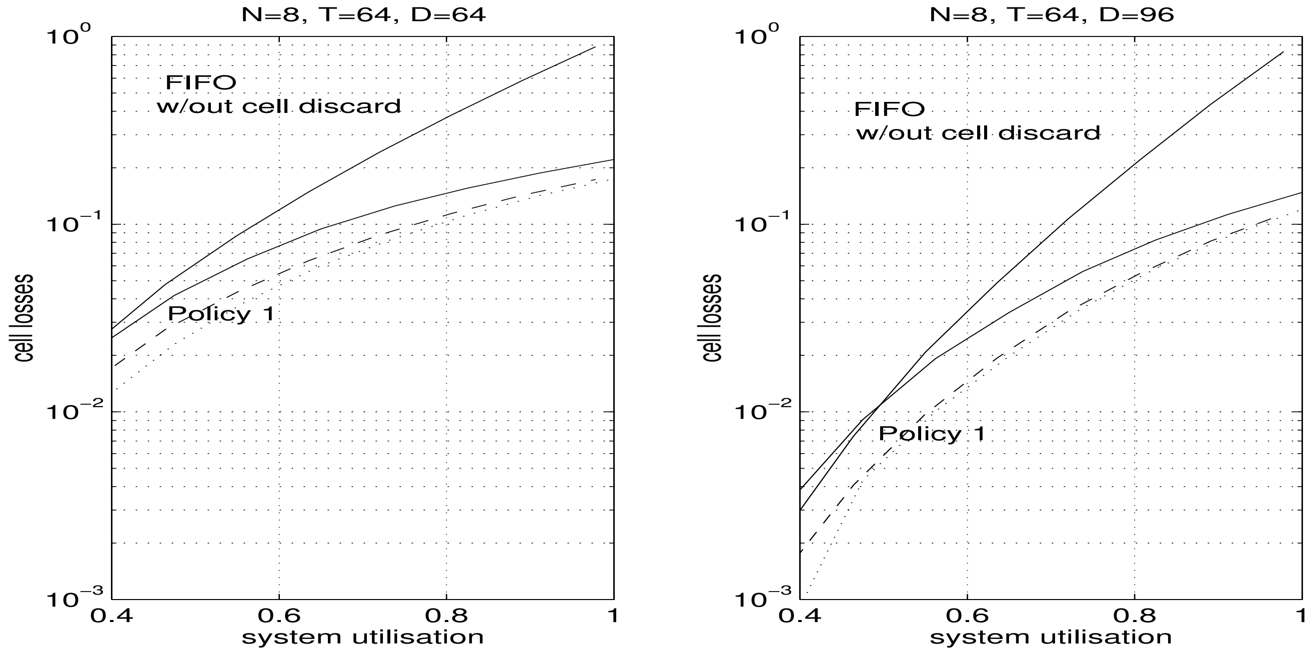

Figure 8: Cell loss curves for N=8 streams, period T=64 and deadlines D=64 and 96 for BURSTY 1-L periodic traffic. Top solid line is FIFO without expired cell discarding, dashed line is FIFO with expired cell discarding, dotted line is Policy 2 and bottom solid line is Policy 1.

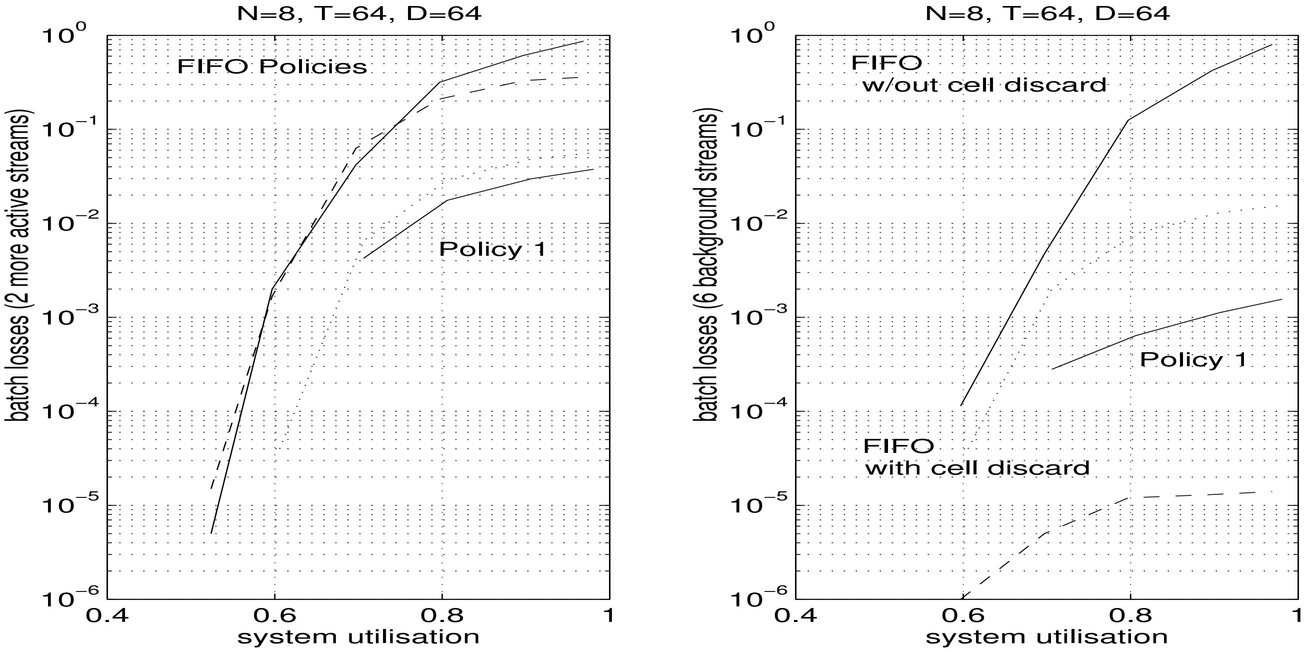

Figure 9: Batch loss curves for N=8 streams, period T=64 and deadline D=64 for INHOMOGENEOUS BURSTY 1-L periodic traffic. Top solid line is FIFO without expired cell discarding, dashed line is FIFO with expired cell discarding, dotted line is Policy 2 and bottom solid line is Policy 1.

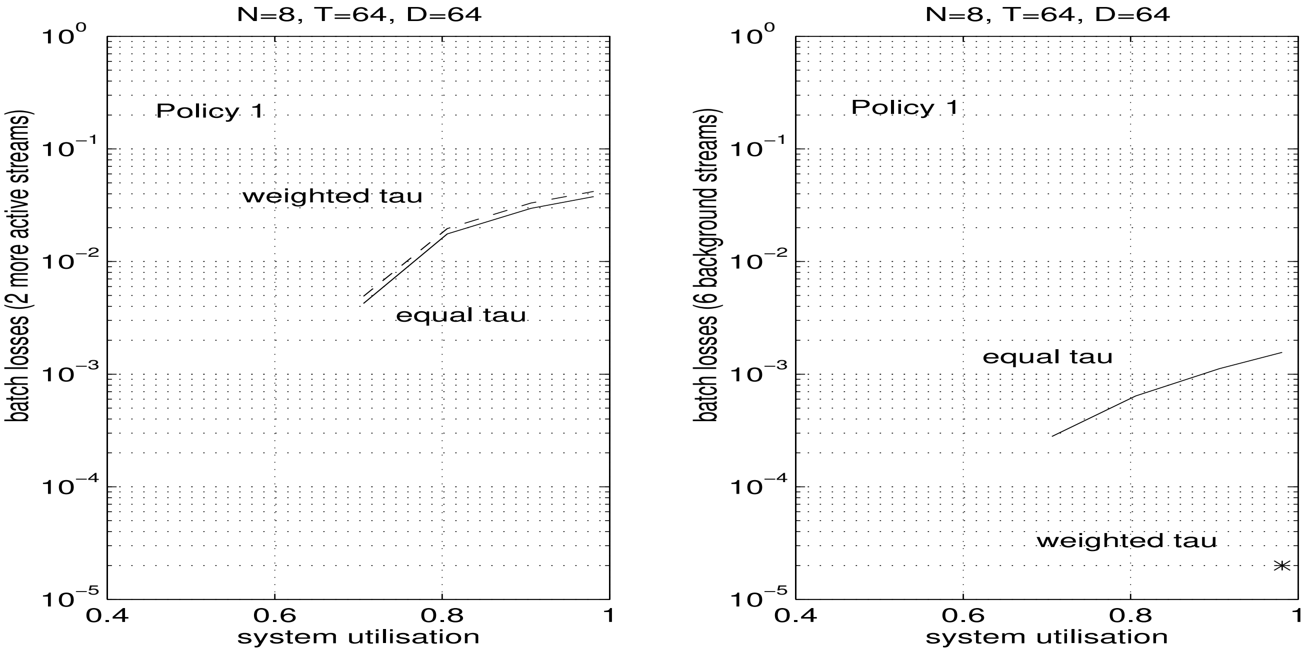

Figure 10: Batch loss curves for N=8 streams, period T=64 and deadline D=64 for INHOMOGENEOUS BURSTY 1-L periodic traffic. Solid line is Policy 1 with equal values of τ and dashed line is Policy 1 with weighted values of τ (tau).

References

[1] R.P. Tsang et al., *Dynamic Resource Control for Continuous Media Traffic over ATM Networks*. Computer Communications, v. 19, 1996, pp. 1092-1111.

[2] S. S. Lam and G.G Xie, *Group Priority Scheduling*. Technical Report TR-95-28, University of Texas at Austin, Austin, Texas, January 1996.

[3] S.S. Lam and G.G. Xie, *Real-Time Block Transfer Under a Link Sharing Hierarchy*. Technical Report, University of Texas at Austin, Austin, Texas, 1996.

[4] X. Wang and I. Stavrakakis, *Study of Scheduling for Group-based Quality of Service Delivery*. Third volume on Performance Modeling and Evaluation of ATM Networks, ed.: D. Kouvatsos, Chapman & Hall, 1997.

[5] I.R. Philp, K. Nahrstedt and J.W.S. Liu, *Scheduling and Buffer Management for Soft-Real-Tile VBR Traffic in Packet-Switched Network*. Technical Report, University of Illinois at Urbana-Champaign, Ill.

[6] K. Nahrstedt, *Challenges of Provided End-to-End QoS Guarantees in Networked Multimedia Systems*. Technical Report, University of Illinois at Urbana-Champaign, Ill.

[7] J.M. Mc Manus and K.W. Ross, *Video on Demand over ATM: Constant-Rate Transmission and Transport*. Technical Report, University of Pennsylvania, Philadelphia, PA, November 1995. http://www.seas.upenn.edu/ ross/.

[8] V. Srinivasan, A. Ghanwani and E. Gelenbe, *Block Loss reduction in ATM Networks*. Computer Communications, v. 19, 1996, pp. 1077-1091.