Study of Multiplexing for Group-based Quality of Service Delivery

Xin Wang and Ioannis Stavrakakis
Electrical & Computer Engineering,
Northeastern University,
360 Huntington Avenue,
Boston, MA 02115, U.S.A.
email: ioannis@cdsp.neu.edu

*

Abstract

In this work, the problem of scheduling packets with a group as opposed to an individual deadline is considered. Packets of the same group are supposed to belong in the same application data unit for which a deadline is set, and they are assumed to arrive over fixed time intervals (frames). The Frame-based Shortest Time to Extinction (F-STE) scheduling policy is considered at a multiplexer fed by multiple streams with group deadlines; expired packets are dropped (not served). The performance of the F-STE policy is evaluated in terms of the induced packet loss (deadline violation) probabilities. It is established that the F-STE policy outperforms the similarly simple First In First Out (FIFO) policy, as well as the potentially more complex FIFO policy which identifies and drops expired packets.

1 INTRODUCTION

High speed networks are today considered to be capable of supporting high-rate real-time applications such as video and multimedia. In order for the Quality of Service (QoS) of such applications to be delivered, networks must meet stringent performance requirements in terms of delay, delay jitter, throughput and packet loss. The timely delivery of packets associated with a real time application can be achieved in a number of ways. The direct approach would be to associate deadlines by which the packets must be transmitted and employ deadline-based

*Research supported in part by the Advanced Research Project Agency (ARPA) under Grant F49620-93-1-0564 monitored by the Air Force Office of Scientific Research (AFOSR).

scheduling mechanisms within the network. Examples of such scheduling policies are the Earliest Due Date (EDD) (Liu *et al.* 1973) (Lim *et al.* 1990) (Saito 1990), the Delay-EDD (Ferrari *et al.* 1990) and the Shortest-Time-to-Extinction (STE) (Panwar *et al.* 1988). Alternatively, scheduling policies which control the induced delay jitter or guarantee a certain transmission (service) rate could be employed. Such policies include the Virtual Clock (VC) (Zhang 1990) (Lam *et al.* 1995), Hierarchical Round Robin (HRR) (Kalmanek *et al.* 1990), Stop-and-Go (Golestani 1990), Rate Controlled Static Priority Queueing (RCSP) (Zhang *et al.* 1993), Fair Queueing (Demers *et al.* 1989), Weighted Fair Queueing (WFQ) (Takagi *et al.* 1991), also known as General Processor Sharing (GPS), and the packet based version of GPS, Packet-by-Packet Generalized Processor Sharing (PGPS) (Parekh *et al.* 1993) (Parekh *et al.* 1994). Since this paper proposes and investigates a new *deadline* driven scheduling policy, the related EDD and STE policies mentioned above are discussed in more detail.

The classical EDD policy minimizes the maximum lateness and maximum tardiness by transmitting packets in the order of their due dates. The lateness of a packet is defined to be the difference between the finishing time of its transmission and its due date. Its tardiness is given by $\max\{0, lateness\}$. The STE policy is very similar to the EDD policy in the sense that the packet with the earliest due date is served first. However, it differs from the EDD in that packets which miss their deadline (expired packets) are not transmitted. It has been established that the STE scheduling policy is optimal in the sense that it maximizes the fraction of packets transmitted before their deadline or, equivalently, minimizes the deadline violation probability.

In some applications, such as packetized voice or video, packets must be transmitted before their deadlines expire in order to be useful. On the other hand, these applications can tolerate the loss of a small fraction of packets. Clearly, the STE scheduling policy is optimal for such applications. However, implementation of the STE policy requires a time-consuming sorting process in order to identify the task with the earliest deadline. This makes the STE policy too complex for real-time realization in high speed networks. For this reason, the simple to implement FIFO policy is often employed. Although the performance of the FIFO scheduling policy is typically inferior to that of the STE policy with respect to the induced deadline violation probability, it is easy to establish that the STE policy and the FIFO policy that drops expired packets become identical if all packets sharing the same resource have identical time to extinction upon arrival. In such environments, the easy to implement FIFO which drops expired packets is also optimal in the sense that it minimizes the deadline violation probability.

The nature of most real-time applications suggests, however, that a deadline be associated with an application data unit (group of packets), such as a video-frame in video applications, rather than with individual packets. For example, in the VBR video encoder, the data for one horizontal strip of blocks is assembled into a single self-contained unit which is transmitted by means of multiple ATM cells (the unit is referred to as a video packet) (Verbiest *et al.* 1989). The address of the strip in the frame provides a time reference for decoder synchronization. For such applications, a deadline should be associated with each data unit rather than with each packet. Packets belonging in the same group now have variable times to extinction upon arrival, and the FIFO policy is non-optimal even if all the applications have identical QoS requirements.

In this work, the Shortest Time to Extinction (STE) scheduling policy is considered for scheduling packets with a group as opposed to individual deadline. Packets of the same group are defined here as packets arriving over a fixed time interval (frame). Packets belonging to the group with the earliest deadline (shortest time to extinction) are served first and the policy is called Frame-STE (F-STE) due to the central role of the underlying frame in determining the arrivals and the associated deadlines. It is easy to establish that the F-STE is equivalent to a dynamic Deadline-Ordered Head-of-Line (DO–HoL) priority scheduling policy. Clearly, F-STE is optimal in the sense that it minimizes the deadline violation probability. While a scheduler implementing the STE policy must search for the packet with the shortest time to extinction every time it serves a packet, no such search is necessary for the F-STE policy. This is an advantage intrinsic to group-based scheduling policies, in which the scheduling priority needs to be updated less often than in a packet-based scheduling policy, and has been pointed out by other authors (for example, (Lam *et al.* 1996)).

In an environment in which packets must be transmitted before their deadlines in order to be useful, a scheduling policy that drops expired packets is more effective than the one which serves expired packets, in the sense that a larger fraction of packets can be transmitted before their deadlines. In order for expired packets to be dropped in a FIFO scheduler, they typically need to be time-stamped and searched for, which adds significantly to the implementation complexity. In this paper, an implementation of the F-STE scheduler is outlined which does not require time-stamping and searching mechanisms. The performance of the F-STE policy is compared with that of the easy to implement (standard) FIFO policy which does not drop expired packets, as well as the more complex FIFO policy which drops expired packets.

Finally, it should be noted that the non-work-conserving version of the F-STE policy – considered also in this paper – resembles the Stop-and-Go scheduling policy (Golestani 1990). However, the objectives, the evaluated performance measures of interest and the work-conserving version of the F-STE policy considered in this paper are different from those in (Golestani 1990).

2 DESCRIPTION OF THE SYSTEM AND THE SCHEDULING POLICY

A network node with N incoming and one outgoing links is considered in this paper. All links are assumed to have the same capacity and thus a packet (fixed size information unit) transmission requires the same amount of time, referred to as the slot. Slot level system synchronization is assumed implying that packet arrivals and departures from the node occur at slot boundaries.

In addition to the time slot, a larger time constant of length T (in slots), called frame, is associated with each of the input streams. T takes integer values and is assumed to have the same value for all streams to simplify the analysis and the discussion. The relative shift of the frame boundaries of the incoming streams can be arbitrary. Without loss of generality and for simplicity, the frame boundaries are assumed to be uniformly distributed in this paper. Packet arrivals associated with the same stream and frame occur according to a Bernoulli process with a fixed rate. This rate may be different for different streams.

The frame is also assumed to modulate the QoS of the associated packets. Specifically, all packets of a stream arriving over the same frame are assumed to have the same deadline, set to be equal to the end of the next frame. Since the objective of this work is to determine whether significant performance gain can be achieved under the proposed policy, the deadline is set so that the analysis complexity be minimized. Packets which are not transmitted (served) by their deadline are dropped.

It is easy to see that packets with the same deadline (end of next frame) will have different times to extinction upon arrival, since their arrival times are different (one packet per slot per frame). When the frame boundaries associated with different streams are not synchronized, which would be typically the case, it is easy to establish that earlier arriving packets from one stream may have a larger time to extinction than later arriving packets from a different stream. As a result, the easy to implement FIFO policy would not serve packets in order of decreasing times to extinction and, for this reason, would not minimize the deadline violation probabilities; as said earlier, these probabilities are minimized under the STE policy.

In this paper, packets are served according to the STE policy. Because of the frame-structured packet arrivals and QoS definition, the STE policy – called in this environment Frame-based STE (F-STE) - is easy to implement, and thus, can be employed in a high-speed networking environment. Details of a simple implementation are presented in the next subsection.

The following definitions and alternative policy description will be useful in both the analysis and the implementation of the F-STE policy. Figure 1 depicts the arrival axes of $N = 4$ streams. The frames of all streams are identical (of length equal to T slots) and their boundaries are uniformly distributed. Let $\{t_k\}$ denote the sequence of frame boundaries from any streams, that is, t_k denotes the time when the kth frame ends. The deadline of packets associated with the frame ending at t_k is time t_{k+N}. Let I_k denote the stream whose frame ends at t_k; $I_k \epsilon \{0, 1, 2, \ldots, N-1\}$.

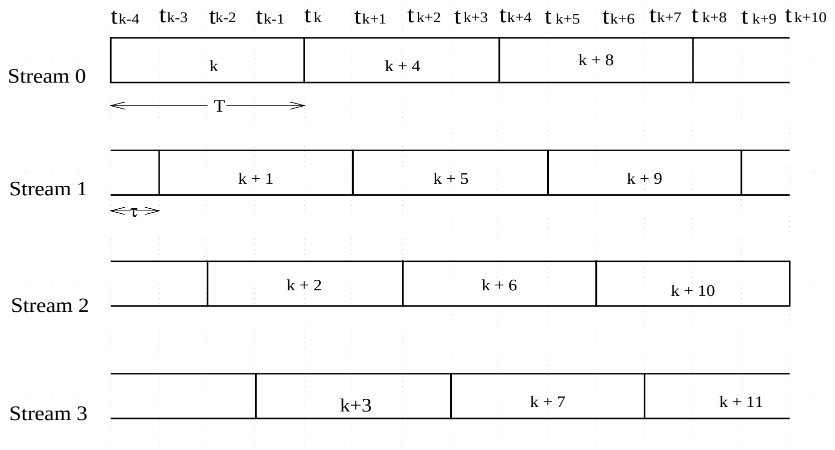

Figure 1 Frame structure of $N = 4$ traffic streams

According to the F-STE policy and in view of the above assumptions, the scheduler serves the streams according to the ordered HoL priority $(I_{k+1}, I_{k+2}, \ldots, I_{k+N})$ over the interval $(t_k, t_{k+1}]$. That is, high priority is given to packets from stream I_{k+1}, second highest priority is given to the packets from stream I_{k+2}, etc. It is easy to see that this policy serves packets in order of decreasing times to extinction. The ordered-HoL priority $(I_{k+1}, I_{k+2}, \ldots, I_{k+N})$ is cyclically shifted at the next frame boundary to $(I_{k+2}, I_{k+3}, \ldots, I_{k+N+1})$; and for this reason, the comprehensive service policy is called Dynamic-Ordered HoL priority policy (DO-HoL).

2.1 Implementation

The F-STE policy (or DO-HoL policy which drops expired packets) can be implemented without requiring a search for the packet with the shortest time to extinction, as described below and depicted in Figure 2.

Each stream is assigned a logically distinct data queue of capacity T which is the maximum number of packets delivered by the stream over one frame. At time t_k, the content of the data queue I_k is shifted to the service queue of capacity T. If the service queue overflows, the dropped packets will be precisely the ones which will miss their deadline. Packets accepted by the service queue will be served within T time units after their shifting since the service queue is served continuously.

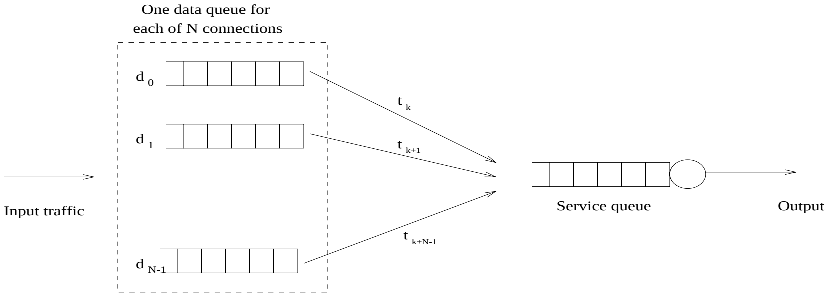

Figure 2 Implementation of the F-STE policy at a node with N incoming links (packets arriving over a frame accumulate in the corresponding data queue and are shifted to the service queue at the end of the frame).

If the service queue becomes empty at some time t, $t_k < t < t_{k+1}$, then the server either remains idle until data queue I_{k+1} is shifted into the service queue at t_{k+1} (non-work-conserving F-STE policy), or the server attends to the data queues according to the DO-HoL priority policy (work-conserving F-STE policy).

3 PERFORMANCE ANALYSIS

In this section, the packet loss (deadline violation) probability is derived for N streams served under the F-STE policy. The arrival process and QoS requirements are as described earlier.

Let τ_k denote the interval (in slots) between the two consecutive frame boundaries ending at t_k and t_{k+1}; that is $\tau_k = t_{k+1} - t_k$. In view of the assumptions in section 2, τ_k is independent of k and will be denoted by τ. Clearly, $\tau = \frac{T}{N}$. Let I_k denote the data queue whose content is to be shifted to the service queue at t_k. Let Q_k denote the service queue occupancy at time t_k, after the packets from data queue I_k are shifted into the service queue. The 2-dimensional process $\{Q_k, I_k\}_{k\geq 0}$ with state space $\{(i,j) : 0 \leq i \leq T, 0 \leq j \leq N-1\}$ is defined to be the *system* process and is employed in the analysis of the scheduling policies. Let $P(i,j,i',j')$ denote the transition probability that process $\{Q_k, I_k\}_{k\geq 0}$ moves from state (i,j) at time t_k to state (i',j') at time t_{k+1}. Note that this probability is zero if $j' \neq (j+1)\mathrm{mod}(N)$, since only transitions from j to $j' = (j+1)\mathrm{mod}(N)$ are possible. Throughout the paper, j' will always be equal to $(j+1)\mathrm{mod}(N)$.

In the next subsection, the system process $\{Q_k, I_k\}_{k\geq 0}$ is determined under the non-work-conserving F-STE policy. In subsection 3.2, certain auxiliary processes are defined in order to obtain bounds or approximations on the evolution of the system process $\{Q_k, I_k\}_{k \geq 0}$ under the work-conserving F-STE policy. By employing the derivations of subsections 3.1 and 3.2, the packet loss (deadline violation) probability is derived for all cases in subsection 3.3. In the derivations that follow, A_m^k denotes the random variable representing the number of packets from stream k arrived over m slots. The probability mass function of A_m^k is given by the convolution of the Bernoulli random variables of the associated stream which are assumed to be of fixed value over a frame.

3.1 System equations for the non-work-conserving F-STE policy

Under this policy, packets can be served only after they have been shifted to the service queue, that is, only after the end of the frame over which they arrive. The server remains idle when the service queue is empty even if the data queues may be non-empty. This non-work-conserving version of the F-STE policy is considered primarily to facilitate the study of the F-STE policy presented afterwards. Nevertheless, this policy may be employed for applications in which the packets of the frame cannot be served before the packets of the entire frame have been received. It is easy to establish that the system process $\{Q_k, I_k\}_{k \geq 0}$ is a Markov chain, since I_k is a periodic Markov chain and Q_{k+1} is (probabilistically) determined from Q_k and A_m^k. The probability $P(i, j, i', j')$ that the Markov chain moves from state (i, j) to state (i', j') is given below. When $i \geq \tau$, the system queue cannot become empty between two successive shifts to the service queue, τ slots apart; when $i < \tau$, the service queue becomes empty between two successive shifts. Note that since only one packet can arrive over one slot, the maximum number of packets arriving over T time slots (one time frame) is T.

Case A : $\tau \leq i$

$$\text{For} \quad (i-\tau) \leq i' \leq T-1 : \quad P(i,j,i',j') = P\{A_T^{j'} = i' - i + \tau\} \tag{1}$$

$$\text{For} \qquad i' = T : \qquad P(i,j,i',j') = \sum_{k=T-i+\tau}^{T} P\{A_T^{j'} = k\} \tag{2}$$

Case B : $0 \leq i < \tau$

$$\text{For} \quad 0 \leq i' \leq T : \quad P(i,j,i',j') = P\{A_T^{j'} = i'\} \tag{3}$$

3.2 Auxiliary system equations for the work-conserving F-STE policy

According to the work-conserving F-STE policy, if the service queue becomes empty at some time t, $t_k < t < t_{k+1}$, the server attends to the data queues according to the ordered HoL priority $(I_{k+1}, \ldots, I_{k+N})$ (see section 2). An implication of this policy is that the system process $\{Q_k, I_k\}_{k\geq 0}$ is not a Markov chain. Some of the arrivals to be shifted into the data queue at t_{k+1}, given by $A_T^{I_{k+1}}$, may have already been served while in the corresponding data queue. Since the latter quantity is needed to determine the evolution of $\{Q_k, I_k\}_{k\geq 0}$, $\{Q_k, I_k\}_{k\geq 0}$ is not a Markov chain.

In order to obtain the system evolution for the work-conserving case, the Markovian process $\{Q_k, I_k, \bar{Q}_k\}_{k\geq 0}$ can be considered, where $\bar{Q}_k$ is an $(N-1)$-dimensional vector representing the occupancy of the data queues which are not shifted to the service queue at time t_k; the one shifted is always 0 at t_k. The resulting analysis would be involved and the complexity would be prohibitive for a large N. Nevertheless, this approach has been employed in the study of the work-conserving F-STE policy supporting $N = 2$ streams. The derivation of the system equations for this case is presented in the appendix. Results for the two stream case are presented at the end of the paper.

The work-conserving F-STE policy for large N is studied here by following the approach described below. This approach leads to the derivation of bounds and approximations on the performance of the work-conserving F-STE policy with a complexity similar to that for the non-work-conserving case, even for large N.

Since all packet losses (deadline violations) are "registered" as service queue overflows occurring at some t_k (when the corresponding data queue is shifted into the service queue), bounds on the packet loss under the work-conserving F-STE policy can be obtained by defining proper auxiliary systems U and L which are such that their service occupancies at times t_k – denoted by Q_k^u and Q_k^l respectively – satisfy

$$Q_k^l \leq Q_k \leq Q_k^u. \tag{4}$$

For properly selected auxiliary systems (policies), the above relationship would then lead to a similar one involving the corresponding packet loss probabilities given by

$$L_k^l \leq L_k \leq L_k^u. \tag{5}$$

Furthermore, by constructing the auxiliary system in a way that the system processes posses the Markovian property, the packet loss probabilities will be easily derived, as described in subsection 3.3.

Since the non-work-conserving F-STE scheduler remains idle when the service queue becomes empty, and is otherwise identical to the work-conserving F-STE scheduler, it is evident that the associated system can serve as the auxiliary system U. In 3.2.1, an auxiliary system L is proposed, and equations are derived for the system evolution. Another auxiliary system A is proposed in 3.2.2, in order to obtain a close approximation for the packet loss of the work-conserving F-STE policy. The following will be satisfied by this auxiliary system.

$$Q_k^l \leq Q_k^a \leq Q_k^u, \qquad L_k^l \leq L_k^a \leq L_k^u. \tag{6}$$

The tightness of the bounds and the accuracy of the approximate results are discussed in the last section where numerical results are presented.

Auxiliary system L

The auxiliary system L is defined to be similar to the system under the non-work-conserving F-STE policy with the following differences. If the service queue becomes empty before the next data queue shifting time t_{k+1}, the number of slots wasted under the non-work-conserving F-STE policy is considered as a service credit available at t_{k+1}; this amount is registered as an equivalent negative contribution to the service queue occupancy at t_{k+1}. There is an upper bound on the allowed credit accumulation equal to $T-\tau$. This bound is set by the requirement that earlier service credit cannot be utilized by future arrivals since it will be lost ("expired credit"). More specifically, service queue occupancy $x < \tau - T$ at t_k would imply the generation of at least $|x-(T-\tau)|$ credit units during the interval $(t_{k-N}, t_{k-(N-1)}]$; this credit must be wasted since all packets in the data queues at t_k have arrived after $t_{k-(N-1)}$.

The one-step transition probability for $\tau - T \leq i \leq T$ for system L is given by

$$\text{For} \quad i' = T: \quad P(i,j,i',j') = \sum_{k=T-i+\tau}^{T} P\{A_T^{j'} = k\} \tag{7}$$

$$\text{For} \quad (\tau - T) < i' < T-1: \quad P(i,j,i',j') = P\{A_T^{j'} = i' - i + \tau\} \tag{8}$$

$$\text{For} \quad i' = \tau - T: \quad P(i,j,i',j') = \sum_{k=0}^{2\tau - i - T} P\{A_T^{j'} = k\} \tag{9}$$

Auxiliary system A

In deriving the evolution of system L, it was assumed that all the "unexpired" slots (inducing "unexpired" credit) wasted under the non-work-conserving policy could be utilized by serving the data queues. This is true only if at least one data queue is non-empty during the interval from when the service queue becomes empty, till the next shift to the service queue. In order to obtain a better approximation for the packet loss probability of the work-conserving F-STE policy, an auxiliary system A is considered, in which a more precise credit accumulation rule is applied. As in the case of the auxiliary system L, the service queue capacity i must satisfy $\tau - T \leq i \leq T$.

Case A : $i' > 0$

In this case, no service credit is generated.

$$\text{For} \quad (i-\tau) \leq i' \leq T-1 : \quad P(i,j,i',j') = P\{A_T^{j'} = i' - i + \tau\} \qquad (10)$$

$$\text{For} \qquad i' = T : \qquad P(i,j,i',j') = \sum_{k=T-i+\tau}^{T} P\{A_T^{j'} = k\} \qquad (11)$$

Case B : $i' \leq 0$

$[B1] : i > 0$

Unlike in system L, out of the $l = |i - \tau|$ non-used (by the packets in the service queue) service opportunities over $(t_k, t_{k+1}]$, only a portion will be registered as credit, based on the number of (earlier) packet arrivals to the data queues which have not been shifted to the service queue at, or before, t_k. Thus, the actual amount of credits kept will be equal to $\min(|l|, \tilde{A_m})$ where $\tilde{A_m} = A^{j'+1}_{(N-2)\tau} + A^{j'+2}_{(N-3)\tau} + \ldots + A^{j'+N-2}_{\tau}$. Each of the terms in $\tilde{A_m}$ represents packet arrivals over some τ and for some stream which have not been shifted to the service queue at, or before, t_k. Arrivals over the interval $(t_k, t_{k+1}]$ over which this credit is generated are not considered since some of them may not arrive before the credit is generated; clearly, this is an approximation. In this case, the only possible transitions are to state i', where $-\tau < i' \leq 0$, and their probabilities are given by

$$\begin{aligned} P(i,j,i',j') \;=\; & P\{A_T^{j'} = i' - i + \tau\} P\{\tilde{A_m} \geq |i'|\} + \\ & \sum_{k=0}^{i'-i+\tau-1} P\{A_T^{j'} = k\} P\{\tilde{A_m} = |i'|\} \end{aligned} \qquad (12)$$

$[B2] : \tau - T \leq i \leq 0$

The service queue is empty at t_k, and $|i|$ service credits have already been accumulated. Let $l = i' - i$. If $l < 0$, $|l|$ is equal to the credit generated over $(t_k, t_{k+1}]$. Since packet arrivals to the data queues occuring before t_{k-1} have been considered toward the determination of the original credit $|i|$ (as indicated in the previous case, $[B1]$), only packet arrivals to the data queues occuring over $(t_{k-1}, t_k]$ will be used to determine the actual new credit to be kept. The latter arrivals are given by $\tilde{A}_n = A_\tau^{j'+1} + A_\tau^{j'+2} + \ldots + A_\tau^{j'+N-2}$. Finally, no credit is generated if $l \geq 0$. The transition probabilities are given by:

For $\tau - T < i' \leq 0$:

$$P(i,j,i',j') = \begin{cases} P\{A_T^{j'} = \tau + l\} & l > 0 \\ P\{A_T^{j'} = \tau + l\} P\{\tilde{A}_n \geq |l|\} + & \\ \sum_{k=0}^{\tau+l-1} P\{A_T^{j'} = k\} P\{\tilde{A}_n = |l|\} & l \leq 0 \end{cases} \tag{13}$$

For $i' = \tau - T$:

$$P(i,j,i',j') = \begin{cases} P\{A_T^{j'} = \tau + l\} & l > 0 \\ \sum_{k=0}^{\tau+l} P\{A_T^{j'} = k\} P\{\tilde{A}_n \geq |l|\} & l \leq 0 \end{cases} \tag{14}$$

3.3 Computation of packet loss probabilities

In view of the Markovian structure for the system processes defined in the previous subsections, the induced packet loss (deadline violation) probabilities are easy to determine. As indicated earlier, the packets whose deadline expires are precisely the packets which overflow from the service queue upon shifting to that queue.

Let L_{ij} denote the number of packets dropped over the interval $(t_k, t_{k+1}]$, where (i,j) is the system state at t_k. Notice that all losses will be associated with packets from stream $j' = (j+1)\mathrm{mod}(N)$. Clearly,

$$L_{ij} = (A_T^{j'} + (i - \tau) - T)^+. \tag{15}$$

Notice that at most $i - \tau$ packets may be lost since the maximum number of per stream and per frame arrivals is equal to T; no loss occurs if $i \leq \tau$. The average value of L_{ij}, $\bar{L_{ij}}$, is given by

$$\bar{L_{ij}} = \sum_{k=1}^{i-\tau} k P(A_T^{j'} = T - (i - \tau) + k). \tag{16}$$

The loss rate for stream $j' = (j+1)\mathrm{mod}(N)$ over $(t_k, t_{k+1}]$, when the system state at t_k is (i,j), is given by

$$R_{ij} = \frac{\bar{L_{ij}}}{\lambda^{j'} T}. \tag{17}$$

where $\lambda^{j'}$ denotes the arrival rate of stream j' per slot. Finally, the packet loss probability is given by

$$L = \sum_{i=\tau+1}^{T} \sum_{j=0}^{N-1} R_{ij}\pi(i,j), \tag{18}$$

where $\pi(i,j)$ is the steady state probability distribution of the corresponding Markov chain.

4 NUMERICAL RESULTS

In this section, two sets of numerical results are presented. Figures 3 to 6 are derived to evaluate the accuracy of the bounds and approximations associated with the analysis of the general F-STE policy, as well as to present exact results for the non-work-conserving F-STE policy, and the work conserving F-STE policy with $N = 2$. Figures 7 and 8 are derived for the comparative evaluation of the F-STE and the FIFO policies.

The exact packet loss probability vs system utilization λ (for $T = 8$) for $N = 2$ streams served under the work-conserving F-STE are shown in Figure 3; simulations are also shown for verification of the equation. Results for systems U (non-work-conserving F-STE policy) and L are also shown.

Results for the packet loss probability for $N = 4$ streams vs system utilization λ (for $T = 8$) and vs frame length (for $\lambda = 0.88$) are presented in Figures 4 and 5, respectively. Results are shown for systems L, U, A, and simulations. For the cases considered here, the approximate system (system A) seems to provide for an accurate approximation of the performance of the work-conserving F-STE policy. The lower bound (auxiliary system L) is observed to be tighter than the upper bound (auxiliary system U, that is the non-work-conserving policy). As λ increases, the tightness of the lower bound improves since more of the unexpired credit in system L will actually be used under the work-conserving F-STE policy. Similarly, the tightness of the upper bound (non-work-conserving system) improves as λ increases since the server will be found idle less often and, thus, the work-conserving and non-work-conserving policies will tend to become identical. As T increases, the tightness of the upper bound deteriorates since the work-conserving and non-work-conserving F-STE policies become increasingly different and the server is found more frequently idle under system U while there is work in the system. The slight improvement of the tightness of the lower bound as T increases may be attributed to an increasing probability that a data queue is non empty and thus increasing amount of credit is kept. The impact of the number of streams N on the induced packet loss probability is illustrated in Figure 6 vs frame length T (for

$\lambda = 0.88$); the results are for the work-conserving policy ($N = 2$, exact analysis) and for the approximate system A ($N = 4, 8, 16$).

Simulation results for the induced packet loss probability under a simple FIFO policy, the FIFO policy in which expired packets are dropped, and the work-conserving F-STE policy are shown in figures 7 and 8 vs system utilization λ (for $T = 16$ and $N = 4$) and vs frame length T (for $\lambda = 0.88$ and $N = 4$), respectively. The results from the analysis for the approximate system A are also shown. All results are derived for symmetric load.

As expected, the packet loss probability of all three scheduling policies decreases as the system utilization decreases. The packet loss probability also decreases as the frame length increases. The latter may be attributed to the associated increase -as the frame length increases- in the time to extinction as well as the increased smoothness of the cumulative traffic over a frame.

The FIFO policy which drops expired packets is seen to induce a smaller packet loss probability than the standard FIFO policy, as expected. The advantage is more pronounced when the scheduler is operating nearly at maximum capacity where the system utilization is high and the time to extinction is short (that is, the frame length is short).

As expected, the work-conserving F-STE policy outperforms both FIFO policies. It should be noted that the F-STE scheduler is comparable in terms of implementation complexity to the scheduler that implements the standard FIFO policy (which does not drop expired packets). Comparison of these two policies (Figures 7 and 8) shows that the F-STE policy always outperforms the FIFO policy by a large margin, irrespective of frame length and system utilization.

Comparison of the F-STE policy with the more complex FIFO policy which drops expired packets reveals that for high system utilization (Figure 7) the performance difference between F-STE and FIFO is small due to the scheduler throughput limitation. As λ decreases, the sub-optimality of the FIFO becomes a factor with increasing weight in inducing losses. The difference in performance between the above two policies is seen to increase as the frame length increases (Figure 8). This may again be attributed to increased sub-optimality of the FIFO policy: as the frame length increases, the range of extinction times of packets of the same frame increases leading to increased sub-optimality of the FIFO policy.

Evidently, the performance difference between the two scheduling policies that drop expired packets decreases under conditions (small frame length, high system utilization) in which the packet loss is largely due to the scheduler limitation, rather than the sub-optimality of FIFO. The performance of the standard FIFO policy remains significantly worse even under these conditions; this may be attributed to the additional sub-optimality of this policy resulting from transmitting packets that have exceeded their deadlines which becomes significant under these conditions.

APPENDIX

A System evolution for 2 stream case (exact analysis)

The accurate calculation of the packet loss probability for the F-STE policy in the case of a node with two streams is described in this appendix. As mentioned in subsection 3.2, the Markov process $\{Q_k, I_k, \bar{Q}_k\}_{k\geq 0}$ (denoted as process M_1) embedded at $\{t_k\}_{k\geq 0}$ needs to be considered. Q_k denotes the service queue occupancy after packets have been shifted in from data queue I_k at t_k. $\bar{Q}_k$ is an $(N-1)$-dimensional vector representing the occupancy of the data queues at time t_k which have not been shifted to the service queue; that is, $\bar{Q}_k = (Q_k^{I_{k+1}}, Q_k^{I_{k+2}}, ..., Q_k^{I_{k+N-1}})$, where Q_k^j denotes the occupancy of data queue j at t_k. Notice that $Q_k^{I_k} = 0$ since this data queue is being emptied (shifted to the service queue) at t_k and thus there is no need to include it in the $(N-1)$-dimensional process. The state space of M_1 for $N = 2$ is given by $S_2 = \{(i, j, l) : 0 \leq i \leq T, 0 \leq j \leq N-1, 0 \leq l \leq \tau\}$. Let $P(i, j, l; i', j', l')$ denote the probability that Markov chain M_1 moves from state (i, j, l) at t_k to state (i', j', l') at t_{k+1}, (i, j, l), $(i', j', l') \epsilon S_2$. These probabilities are described below where different cases are considered based on the starting state of the service queue $(Q_k = i)$.

<u>Case A : $\tau \leq i$</u> (service queue will not become empty before t_{k+1})

For $(i-\tau) \leq i' \leq T-1$:

$$P(i, j, l; i', j', l') = P\{A_T^{j'} = i' - i + \tau\}P\{A_\tau^j = l'\} \qquad (19)$$

For $i' = T$:

$$P(i, j, l; i', j', l') = \sum_{k=T-i+\tau}^{T} P\{A_T^{j'} = k\}P\{A_\tau^j = l'\} \qquad (20)$$

<u>Case B : $0 \leq i < \tau$</u> (service queue will become empty before t_{k+1})

In this case, the service queue becomes empty at time $t_k + i$, and remains empty (for a period of $\tau - i$ slots) till the next shift to the service queue at t_{k+1}. Over the interval $(t_k + i, t_{k+1}]$, packets are served from the data queues according to the HoL priority $(I_{k+1}, I_{k+2}) = (j', j)$. Two cases need to be considered depending on whether $i' = 0$ or $i' > 0$ as explained below.

Case B.1 : $0 \leq i < \tau$, $i' > 0$

In this case, data queue $I_{k+1} = j'$ is always nonempty over $(t_k + i, t_{k+1}]$ since otherwise $i' = 0$. The latter is seen by noting that data queue j' will remain empty after it becomes empty for the first time within the interval $(t_k + i, t_{k+1}]$ – and, thus, $i' = 0$ – since stream j' generates at most one packet per slot and receives HoL priority over the interval $(t_k + i, t_{k+1}]$. As a result, no other data queue receives any service over this interval when $i' > 0$ and the transition probabilities of M_1 depend only on the cumulative arrivals over T for stream j' and τ for stream j, and they are identical to those under Case A above.

Case B.2 : $0 \leq i < \tau$, $i' = 0$

In this case, data queue $I_{k+1} = j'$ becomes empty at some slot in $(t_k + i, t_{k+1}]$ (and remains empty thereafter). During the slots in which data queue $I_{k+1} = j'$ is empty *and* no packet is generated by stream j' (which would be served under the HoL priority), the other queue will be served if nonempty. In order to determine the content of data queue j at t_{k+1}, to determine the state of $\bar{Q}_{k+1} = Q^j_{k+1}$ (since $I_{k+1} = j'$ and $j = (j' + 1) \bmod (2)$), the slot by slot evolution of data queue j needs to be followed over the interval $(t_k + i, t_{k+1}]$. Since the evolution of this queue depends on whether data queue j' is empty or not, the evolution of both queues over $(t_k + i, t_{k+1}]$ is considered by using the auxiliary Markov chain defined next.

Let $(\hat{Q}^{j'}_0, \hat{Q}^j_0)$ denote the occupancies of the corresponding data queues when the service queue becomes empty (at $t_k + i$). Let $\{\hat{Q}^{j'}_n, \hat{Q}^j_n\}_{n \geq 0}$ denote the data queue occupancy process embedded at the *slot* boundaries with initial state $(\hat{Q}^{j'}_0, \hat{Q}^j_0)$. This process (referred to as process M_2) evolves as a Markov chain over the interval $(t_k + i, t_{k+1}]$ and its state after $\tau - i$ transitions, $(\hat{Q}^{j'}_{\tau-i}, \hat{Q}^j_{\tau-i})$, is such that : $(Q_{k+1}, I_{k+1}, \bar{Q}_{k+1}) = (\hat{Q}^{j'}_{\tau-i}, j', \hat{Q}^j_{\tau-i})$. That is, the $(\tau - i)$-step transition of M_2 will determine the state of M_1 at t_{k+1}. The 1-step and m-step transition probabilities of M_2 are derived below. Let the state space of M_2 be defined by $\hat{S}_2 = \{(\hat{q}^{j'}_n, \hat{q}^j_n) : 0 \leq \hat{q}^{j'}_n \leq T, 0 \leq \hat{q}^j_n \leq \tau\}$. Let A^j denote the Bernoulli random variable describing the number of packets generated over a slot by stream j.

For $0 < \hat{Q}^{j'}_n \leq T$, data queue j' will be served over the current slot and thus the evolution of the queues is given by

$$\begin{aligned} \hat{Q}^{j'}_{n+1} &= \hat{Q}^{j'}_n - 1 + A^{j'}, & (21) \\ \hat{Q}^j_{n+1} &= \hat{Q}^j_n + A^j. & (22) \end{aligned}$$

Thus, the 1-step transition probability of M_2 is given by

$$\begin{aligned} P_{M_2}(\hat{q}_n^{j'}, \hat{q}_n^j; \hat{q}_{n+1}^{j'}, \hat{q}_{n+1}^j) &= P\{A^{j'} = \hat{q}_{n+1}^{j'} - \hat{q}_n^{j'} + 1\} \\ & \quad P\{A^j = \hat{q}_{n+1}^j - \hat{q}_n^j\} \end{aligned} \tag{23}$$

for $0 < \hat{q}_n^{j'} \leq T, \quad 0 \leq \hat{q}_{n+1}^{j'} \leq T, \quad 0 \leq \hat{q}_n^j, \hat{q}_{n+1}^j \leq \tau$.

For $\hat{Q}_n^{j'} = 0$, data queue j will be served if nonempty *and* no packet is generated by stream j' over the current slot. Thus the evolution of the queues is given by

$$\hat{Q}_{n+1}^{j'} = 0, \tag{24}$$

$$\hat{Q}_{n+1}^j = (\hat{Q}_n^j - 1_{\{A^{j'}=0\}})^+ + A^j. \tag{25}$$

Thus, the 1-step transition probability of M_2 is given by

$$\begin{aligned} P_{M_2}(0, \hat{q}_n^j, 0, \hat{q}_{n+1}^j) &= P\{A^{j'} = 1\} P\{A^j = \hat{q}_{n+1}^j - \hat{q}_n^j\} + \\ & \quad P\{A^{j'} = 0\} P\{A^j = \hat{q}_{n+1}^j - \hat{q}_n^j + 1\} \end{aligned} \tag{26}$$

for $0 < \hat{q}_n^j \leq \tau, \quad 0 \leq \hat{q}_{n+1}^j \leq \tau$, and

$$\begin{aligned} P_{M_2}(0, 0; 0, \hat{q}_{n+1}^j) &= P\{A^{j'} = 1\} P\{A^j = \hat{q}_{n+1}^j\} + \\ & \quad P\{A^{j'} = 0\} 1_{\{\hat{q}_{n+1}^j = 0\}} \end{aligned} \tag{27}$$

for $0 \leq \hat{q}_{n+1}^j \leq 1$.

The m-step transition probability of M_2 can be derived recursively in terms of the $(m-1)$-step and the 1-step transition probabilities and it is given by

$$\begin{aligned} & P_{M_2}^m(\hat{q}_n^{j'}, \hat{q}_n^j; \hat{q}_{n+m}^{j'}, \hat{q}_{n+m}^j) = \\ & \sum_{\hat{q}_{n+m-1}^{j'}=0}^{T} \sum_{\hat{q}_{n+m-1}^{j}=0}^{\tau} P_{M_2}^{m-1}(\hat{q}_n^{j'}, \hat{q}_n^j; \hat{q}_{n+m-1}^{j'}, \hat{q}_{n+m-1}^j) \\ & \quad P_{M_2}(\hat{q}_{n+m-1}^{j'}, \hat{q}_{n+m-1}^j; \hat{q}_{n+m}^{j'}, \hat{q}_{n+m}^j) \end{aligned} \tag{28}$$

for $0 \leq \hat{q}_n^{j'}, \hat{q}_{n+m-1}^{j'}, \hat{q}_{n+m}^{j'} \leq T, \ 0 \leq \hat{q}_n^j, \hat{q}_{n+m-1}^j, \hat{q}_{n+m}^j \leq \tau$.

Finally, the transition probabilities of process M_1 for case B.2 are given by

$$P(i,j,l;0,j',l') = \sum_{k1=0}^{i}\sum_{k2=0}^{i} P\{A_i^{j'}=k_1\}P\{A_i^{j}=k_2\} \\ P_{M_2}^{\tau-i}(l+k_1,k_2;0,l'), \quad (29)$$

where A_i^j denotes the number of packet arrivals from stream j over i slots with $A_0^j \triangleq 0$.

B Computation of packet loss probability

The loss probability can be calculated by following a similar approach as in subsection 3.3. Let (i,j,l) be the system state at time t_k. The number of packet losses from stream $j' = (j+1)\mathrm{mod}(2)$ over the interval $(t_k, t_{k+1}]$ is given by

$$L_{ijl} = (A_T^{j'} + (i-\tau) - T)^+. \quad (30)$$

Notice that only packets from stream j' may expire over this interval. The average value of L_{ijl} is given by

$$\bar{L}_{ijl} = \sum_{k=1}^{i-\tau} kP(A_T^{j'} = T-(i-\tau)+k). \quad (31)$$

The loss rate for stream j' over $(t_k, t_{k+1}]$, when the system state at t_k is (i,j,l), can now be written as

$$R_{ijl} = \frac{\bar{L}_{ijl}}{\lambda^{j'}T}. \quad (32)$$

The total loss probability is therefore

$$L = \sum_{i=\tau+1}^{T}\sum_{j=0}^{N-1}\sum_{l=0}^{\tau} R_{ijl}\pi(i,j,l), \quad (33)$$

where $\pi(i,j,l)$ is the steady state probability distribution of the Markov chain M_1.

REFERENCES

L. Zhang (1990) “Virtual Clock: A New Traffic Control Algorithm for Packet. Switching Networks”, *Proceedings of SIGCOMM’90*, pp. 19-29, 1990.

Simon S. Lam and Geoffrey G. Xie (1995) “Burst Scheduling : Architecture and Algorithm for switching packet video”, *Technical Report TR-94-20, Department of Computer Sciences, UT-Austin*, Revised, January 6 1995.

Simon S. Lam and Geoffrey G. Xie (1996) “Group Priority Scheduling”, *Proceedings of IEEE INFOCOM’96, San Francisco*, pp. 1346-1356, June 1996.

W. Verbiest, L. Pinneboo (1989) “A Variable Bit Rate Video Codec for Asynchronous Transfer Mode Networks”, *IEEE Journal on Selected Areas in Communications*, pp. 761-770, June 1989.

C. R. Kalmanek, H. Kanakia, S. Keshav (1990) “Rate Controlled Servers for Very High-Speed Networks”, *Proceedings of IEEE GLOBECOM‘90*, pp. 300.3.1 - 300.3.9 1990.

S. Jamaloddin Golestani (1990) “A Stop-and-Go queueing framework for congestion management”, *Proceedings of IEEE INFOCOM’90, San Francisco*, pp. 527-542, June 1990.

D. Ferrari, P. Verma (1990) “A Scheme for Real-Time Channel Establishment in Wide-Area networks”, *IEEE Journal on Selected Areas in Communications*, pp. 368-379, April 1990.

C. L. Liu, J. W. Layland (1973) “Scheduling Algorithms for Multiprogramming in a Hard Real Time Environment”, *Journal of the ACM*, pp. 46-61, Jan., 1973.

Y. Lim, J. Kobza (1990) “Analysis of a Delay-Dependent Priority Discipline in a Integrated MultiClass Traffic Fast Packet Switch”, *IEEE Transactions on Communications*, pp. 659-665, May 1990.

H. Saito (1990) “Optimal Queueing Discipline for Real-Time Traffic at ATM Switching Nodes”, *IEEE Transactions on Communications*, pp. 2131-2136, Dec. 1990.

S. S. Panwar, D. Towsley, J. K. Wolf (1988) “Optimal Scheduling Policies for a Class of Queues with Customer Deadlines to the Beginning of Service”, *Journal of the ACM*, pp. 832-844, 1988.

A. Demers, S. Keshav, S. Shenker (1989) “Analysis and Simulation of Fair Queuing Algorithm”, *Proceedings of SIGCOMM*, pp. 1-12, 1989.

A. K. Parekh, R. G. Gallager (1993) “A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: the Single Node Case”, *IEEE/ACM Transactions on Networking*, pp. 344-357, June 1993.

A. K. Parekh, R. G. Gallager (1994) “A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: the Multiple Node Case”, *IEEE/ACM Transactions on Networking*, pp. 137-150, April 1994.

Y. Takagi, S. Hino, T. Takahashi (1991) “Priority Assignment Control of ATM Line Buffers with Multiple QoS Classes”, *IEEE Journal on*

Selected Areas in Communications, pp. 1078-1092, Sep. 1991.

H. Zhang, D. Ferrari (1993) “Rate Controlled Static Priority Queueuing”, *Proceedings of IEEE INFOCOM’93, San Francisco*, pp. 227-236, March 1993.

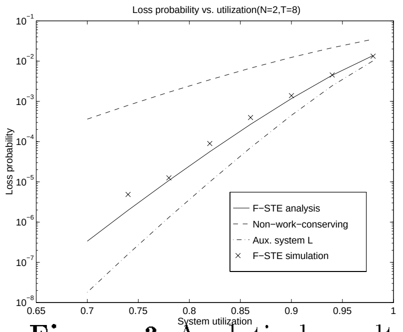

Figure 3 Analytical results for loss probability vs system utilization for $N = 2$ and $T = 8$.

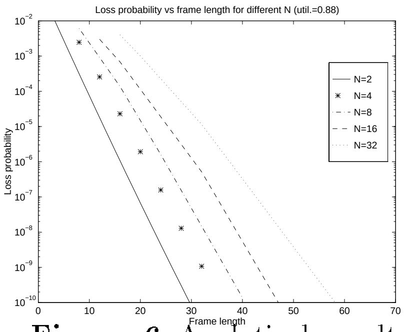

Figure 6 Analytical results for loss probability vs frame length, for different values of N given by system A for system utilization equal to 0.88.

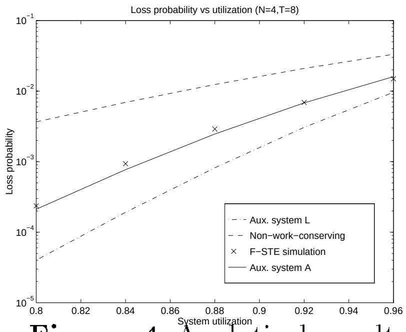

Figure 4 Analytical results for loss probability vs system utilization $N = 4$ and $T = 8$.

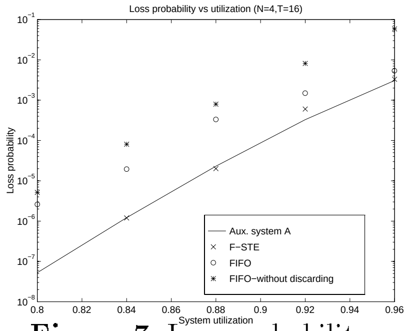

Figure 7 Loss probability vs system utilization for F-STE and FIFO for $N = 4$ and $T = 16$.

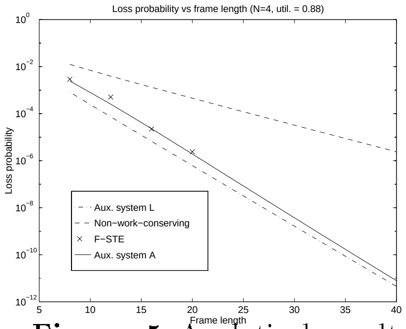

Figure 5 Analytical results for loss probability vs frame length $N = 4$ and system utilization equal to 0.88.

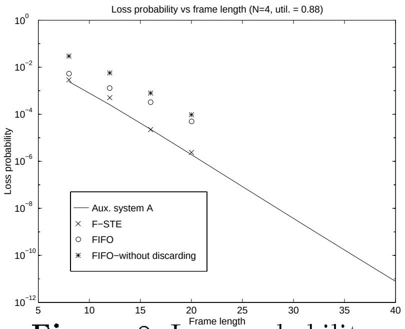

Figure 8 Loss probability versus frame length for F-STE and FIFO for $N = 4$ and system utilization equal to 0.88.