Deterministic Time-Varying Packet Fair Queueing for Integrated Services Networks

Anastasios Stamoulis and *Georgios B. Giannakis*
Dept. of Electrical and Computer Engineering
University of Minnesota, Minneapolis, MN 55455
stamouli,georgios@ece.umn.edu

Abstract— **Packet Fair Queueing (PFQ) algorithms have been extensively studied for provision of Quality of Service (QoS) guarantees in Integrated Services Networks. Because of the fixed weight assignment, the inherent in PFQ *delay-bandwidth coupling* imposes limitations on the range of QoS that can be supported. We develop PFQ with deterministic time-varying weight assignments, and we propose a low-overhead algorithm capable of supporting arbitrary piecewise linear service curves which achieve delay-bandwidth decoupling. Unlike existing service-curve based algorithms, our time-varying PFQ scheme mitigates the *punishment* phenomenon, and allows sessions to exploit the extra bandwidth in underloaded networks.**

Keywords—**packet fair queueing, generalized processor sharing, service curves, integrated services networks**

I. Introduction

In integrated services networks, the provision of Quality of Service (QoS) guarantees depends critically upon the scheduling algorithm employed at the network switches. The scheduling algorithm determines the transmission order of packets in outgoing links and thus, it has a direct impact on the packet delay and achievable throughput, which serve as primary figures of merit for the system performance. The Generalized Processor Sharing (GPS) [6] discipline and the numerous Packet Fair Queueing (PFQ) algorithms are widely considered as the primary scheduler candidates in the emerging broadband multiservice networks. This is because GPS has been shown to provide both minimum service rate guarantees and isolation from ill-behaved traffic sources. Not only have GPS-based algorithms been implemented in actual switches in wired networks, but also they have been studied in the context of wireless networks (see, e.g., [10] and references therein).

The fundamental notion in GPS-based algorithms is that the amount of service session i receives from the switch (in terms of transmitted packets) is proportional to a positive weight ϕ_i. As a result, GPS (and its numerous variants) is capable of delivering bandwidth guarantees; the latter translate to delay guarantees as long as there is an upper bound on the amount of incoming traffic (this bound could be either deterministic for leaky-bucket constrained sessions [6], or stochastic, as in e.g., [16]). However, it is expected that future networks will support multirate multimedia services with widely diverse delay and bandwidth specifications. For example, video and audio have delay requirements of the same order, but video has an order of magnitude greater bandwidth requirements than audio. Known as *delay-bandwidth coupling*, the mutual dependence of delay and bandwidth guarantees constitutes as one of the major shortcomings of PFQ.

To overcome these problems, [3] and [8] introduce the notion of service curves (SC); a SC $S_i(t)$ can be thought of as the minimum amount of service that the switch guarantees to session i in the interval $[0, t]$. SCs dispense with the delay-bandwidth coupling as the shape of $S_i(t)$ could be arbitrary. However, as noted in e.g., [13], SC algorithms suffer from the *punishment* effect: when session i receives in $[0, t_1]$ more service than $S_i(t_1)$ (for example, this could happen if the system is under-loaded in $[0, t_1]$), and the load increases at t_1, then there is an interval $(t_1, t_2]$ where session i does *not* receive any service at all. Eventually session i will receive service at least equal to $S_i(t_2)$ in $[0, t_2]$, but, nevertheless, it is *penalized* for the extra service it received in $[0, t_1]$. From a practical point of view, the punishment phenomenon is undesirable, because it does not allow sessions to take advantage of potentially available bandwidth in the system. We note that GPS does not suffer from the punishment problem [6]. Therefore, it is of interest to study whether PFQ with proper weight assignment is capable of providing the same QoS services as SC-based algorithms, while obviating the punishment phenomenon.

Unfortunately, there are cases where GPS is not capable of supporting arbitrary piecewise-linear SCs even with non rate-proportional weight assignment [9]. Noting that in GPS (and hence in PFQ) the weight assignment is fixed throughout the lifetime of the sessions, herein we study PFQ when the weight assignment is time-varying. In this paper, our goal is to extend PFQ and make it capable of supporting piecewise linear SCs with minimum overhead. Our contribution lies in showing how the time-varying weight assignment can be done deterministically for each session and independently of the other sessions, thus preserving the isolation properties of PFQ. As a result, our deterministic time-varying PFQ (DTV-PFQ) is capable of combining the strengths of GPS and the service flexibility of SC based algorithms. Moreover, illustrating that DTV-PFQ can support as many multiple leaky-bucket constrained sessions as EDF constitutes an important ramification of our work.

Work in this paper was supported by the NSF Wireless Initiative grant no. 99-79443.

0-7803-6451-1/00/$10.00 © 2000 IEEE

II. Model Description

In this section we briefly review results on GPS, SCs, EDF, and describe the deterministic model that we will use in the following sections to study DTV-PFQ. We consider a single network switch which multiplexes data packets sent by various sessions. The amount of traffic ("bits") that session i transmits in $(\tau, t]$ is denoted by $A_i(\tau, t)$, and is upper bounded by the traffic envelope $A_i(t) := \sup_{0 \le \tau}\{A(\tau, t+\tau)\}$. In the case of a leaky-bucket constrained session [2], two positive constants, σ_i, ρ_i, determine the affine $A_i(t) = \sigma_i + \rho_i t$; this traffic envelope is generalized to a multiple leaky-bucket [15]:

$$A_i(t) := \min_{1 \le k \le K_i}\{\sigma_{i,k} + \rho_{i,k}\}.$$

According to [6], a GPS server operates at a fixed rate r and is work-conserving. Each session i is characterized by a positive constant ϕ_i, and the amount of service $R_i(\tau, t)$ session i receives in the interval $(\tau, t]$ is proportional to ϕ_i (provided that the session is continuously backlogged). In the worst case, the minimum guaranteed rate g_i given to session i is $g_i = r\phi_i / \sum_{j=0}^{N-1} \phi_j$, where N is the maximum number of sessions that could be active in the system.

However, switches operate at the packet or cell level and GPS assumes a fluid model of traffic. Hence, in practice GPS is approximated by a Packet Fair Queueing (PFQ) algorithm [6]. Central to almost all PFQ algorithms is the notion of "virtual time" or "system potential" (see [12] for a unifying framework) which is used for the assignment of deadlines: when the n-th packet of session i arrives to the switch at time $t_i^{(n)}$, the packet is time-stamped with a deadline $F_i^{(n)}$, which is a function of the virtual time and ϕ_i. In [6], the virtual time $V(t)$ is a scalar quantity, which is incremented every time a packet arrives or departs from the scheduler:

$$V(t) = V(t-\tau) + \frac{\tau}{\sum_{j \in B(t)} \phi_j} \tag{1}$$

$$F_i^{(n)} = \max\{F_i^{(n-1)}, V(t_i^{(n)})\} + \frac{\phi_i}{r}\,. \tag{2}$$

The packets are transmitted by the switch in increasing order of their time-stamps. The virtual time measures the progress of the work in the system, and it is primarily responsible for the absence of the *punishment* phenomenon in PFQ algorithms [6].

Going a step beyond rate-based (such as GPS) or deadline-based (such as EDF) schedulers, a SC based scheduler attempts in $[0, t]$ to provide service to session i greater or equal to $S_i(t)$ [2, 8]. The SC can have an arbitrary shape (as long as it is a non-decreasing real function of t), and it can be used to provide delay or bandwidth guarantees [8]: a minimum bandwidth g_i is guaranteed if $\partial S_i(t)/\partial t \ge g_i$, and a delay bound d_i is guaranteed if $S_i(t) \ge A_i(t - d_i)$ (Fig. 1). Moreover, it is straightforward to check whether the switch is capable of satisfying all SCs by performing the following test [8]:

$$\sum_{i=0}^{N-1} S_i(t) \le t\,r, \quad \forall t \ge 0\ . \tag{3}$$

To implement SCs in packet switched networks, [8] proposes the SC based Earliest Deadline first policy (SCED), where every packet upon its arrival is assigned a deadline; the deadline is basically a function of $S_i(t)$ and the number of packets session i has transmitted up to time t. The packets are transmitted in increasing order of their deadlines. Apart from "resetting" [8], in SCED the assignment of deadlines to packets of a particular session does not take into consideration the behavior of the other sessions. Thus, the punishment feature of SCED, which is our main motivation for the development of the Deterministic Time-Varying PFQ (DTV-PFQ).

III. Punishment in SCED

As acknowledged in [8] and mentioned in [13], SCED exhibits the punishment property, which does not appear in GPS. Let us illustrate this with the following example (similar to an example in [6]): suppose that we are interested in providing minimum rate guarantees to a system with two sessions "1" and "2". Both of them are to receive 50% of the service rate. Using SCs, we could have $S_1(t) = S_2(t) = rt/2$, whereas under GPS we would have $\phi_1 = \phi_2 = 0.5$. We make the assumption that the scheduler operates in discrete time slots, and we let session "1" start transmitting at slot 0, whereas session "2" starts at slot 10. In the interval $[0, 9]$, session "1" receives 100% of the available bandwidth: normally, source "1" starts transmitting packets at a rate no greater than $r/2$, because this is the rate which is guaranteed to the source. However, based on feedback information from the receiver (e.g., "packets arrived sooner than expected"), session "1" could decide to increase its rate. Figs. 2 and 3 show the bandwidth which is allocated to sessions "1" and "2" in the interval $[0, 30]$ under SCED and WFQ. To study how bandwidth is allocated, we make both sessions continuously backlogged by having them transmit at rate $1.5r$. It is clearly illustrated that under WFQ, session "1" is allocated 100% of the bandwidth in $[0, 9]$ and 50% of the bandwidth in $[10, 30]$. On the other hand, under SCED, session "1" receives 100% of the bandwidth, but in $[10, 14]$ session "1" does not receive any service at all. Eventually, in $[0, 30]$, session "1" receives at least 50% of the bandwidth, as it was advertised. Nevertheless, session "1" is punished for being greedy in $[0, t_1]$.

IV. Deterministic Time-Varying PFQ

To overcome the performance limitations caused by the delay-bandwidth coupling of GPS, herein we propose a time-varying assignment of weights, which provides us with more degrees of freedom than the non rate-proportional weighting of [14]. In other words, the weight ϕ_i which is assigned to a session i is a

function of time $\phi_i(t)$. We focus in the case where the variations in $\phi_i(t)$ are carried out in a deterministic fashion and unlike [1, 5], we develop a framework with minimum overhead. First, we discuss why in theory a Time-Varying GPS is capable of implementing SCs of arbitrary shapes, but it is difficult to realize them in practice. Then, we focus on piecewise linear SCs[1]. We provide a practically realizable weight assignment algorithm which guarantees that prescribed delay bounds can be met by DTV-PFQ (as long as the delay bounds are EDF-feasible). Hence, we show that our newly introduced scheme is optimal in the schedulability-region sense if the traffic envelopes are piecewise linear functions.

In theory, a time-varying GPS system is capable of accommodating arbitrarily shaped SCs $S_i(t)$ provided that (3) is satisfied. By setting $\phi_i(t) = \partial S(t)/\partial t$, we obtain $\sum_{i=0}^{N-1} \phi_i(t) \leq 1$ at any time t, and as a result, $\int_0^t \phi_i(\tau)\, d\tau \geq S_i(t), \forall t \geq 0$, provided that $S_i(t)$ is differentiable (we will address the case when the derivative does not exist later on). Hence, the deterministic assignment $\phi_i(t) = \partial S(t)/\partial t$ makes GPS equivalent to a SC based scheduler. Intuitively thinking, the equivalence between a time-varying GPS system and a SC based system should not come as a surprise. However, what perhaps comes as a surprise is the difficulty of implementing an arbitrary weight assignment in a packet-by-packet practically realizable system.

In a real system, the GPS scheduler assigns deadlines to incoming packets; these deadlines as explained in Section II, are given as a function of the virtual time of the system and the weight of the session. Let us consider a packet of session i that arrives at time t_1. This packet will be transmitted by the system at a later time $t_2 \geq t_1$. At time t_2, the weight of the session is $\phi_i(t_2) = \frac{\partial S_i(t)}{\partial t}|_{t=t_2}$. However, the time instant t_2 is not known upon the packet arrival at t_1. The time t_2 does not only depend on the backlog of the session i at time t_1, but also on the backlog and future packet arrivals of the other sessions. Therefore, unless the SC has a constant slope (which corresponds to fixed weight assignments) or the service curve depends only on the arrivals of a particular session, it is quite challenging to assign deadlines to packets upon their arrival. A possible solution is the computationally expensive algorithm of [1], which uses a vector $\mathbf{V} \in \mathcal{R}^3$ as virtual time in the system. However, in high-speed (gigabit) networks, the implementation overhead of the scheduler should be kept as small as possible.

Fortunately, in the case of piecewise linear SCs, the implementation of DTV-PFQ is possible using a scalar virtual time. Before we describe our approach, let us define formally the SCs which can be provided to individual sessions. [7, 8] have provided computationally efficient scheduling algorithms for piecewise linear SCs $S_i(t)$ defined as:

$$S_i(t) = \max\{0, \min_{k=1,\ldots,K_i} \{a_{i,k} + b_{i,k}t\}\} ,$$

where $a_{i,k}$, $b_{i,k}$ ($1 \leq i \leq N$, $1 \leq k \leq K_i$) are real constants satisfying:

$$b_1 > b_2 > \ldots > b_{K_i} > 0 \,, 1 \leq i \leq N \text{ and}$$

$$1 \leq \frac{a_2 - a_1}{b_1 - b_2} < \ldots < \frac{a_K - a_{K-1}}{b_{K_i} - b_{K_i - 1}} .$$

Herein, we allow the *piecewise linear* SC to be zero in the interval $[0, T_i)$ and have an initial "burst":

$$S_i(t) = \begin{cases} \max\{0, \min_{k=1,\ldots,K_i} \{a_{i,k} + b_{i,k}t\}\} & : \quad t \geq T_i \\ 0 & : \quad 0 \leq t < T_i \end{cases} , \tag{4}$$

We note that the introduction of the constant T_i (which is zero in [7,8]) allows our DTV-PFQ scheme to model an EDF scheduler for multiple-leaky buckets. Indeed, if session i has traffic envelope:

$$A_i(t) = \min_{1 \leq k \leq K_i} \{\sigma_{i,k} + \rho_{i,k}t\},$$

then the allocation[2] of the SC $S_i(t) \leftarrow A_i(t - d_i)$ guarantees the delay bound d_i as long as (3) holds (Fig. 1). Therefore, modulo approximation errors induced by any virtual-time based implementation [6], our DTV-PFQ scheme is capable of achieving approximately the schedulability region of an EDF scheduler (for multiple leaky-bucket constrained sessions).

As (1) and (2) indicate, the virtual-time based implementation of a PFQ system basically amounts to using session weights for the assignments of deadlines. In our DTV-PFQ, we encounter 3 issues:

I1) the weight of session i assumes K_i discrete values.
I2) the delay factor T_i should be taken into account in the assignment of the deadline of the first packet of a session
I3) the term $a_{i,1}$ (the "burst") in the service curve at time T_i corresponds to an infinite slope and should be handled in an appropriate way.

We can apply the following procedure to address I1): when a packet of session i arrives, we use the method of [7, 8] to determine what is the slope $b_{i,k}$ of $S_i(t)$ which corresponds to that packet. Then, we set ϕ_i equal to $b_{i,k}/r$.

Issue I2) is taken care of by adding to the deadline of the first packet of the session the term VirtualOffset_i, which is defined as:

$$\text{VirtualOffset}_i := \int_0^{T_i} \frac{r}{\sum_{j=1, j\neq i}^{N} \tilde{\phi}_j(t)} I_i(t)\, dt \ , \tag{5}$$

[1] In integrated services networks, piecewise linear SCs can be used to provide multirate services [7], and delay guarantees to sessions constrained by multiple leaky buckets; we note also that multiple leaky-buckets have been proposed to model real-life applications and have been shown to result in improved network utilization [4].

[2] The allocation of the SC amounts to: $T_i \leftarrow d_i$, $b_{i,k} \leftarrow \rho_{i,k}$, $a_{i,k} \leftarrow \sigma_{i,k} - \rho_{i,k} d_i$

where $I_i(t)$ is an indicator function defined by:

$$I_i(t) = \begin{cases} 1 : \forall j \; S_j(t) \text{ is differentiable at t} \\ 0 : \qquad \text{otherwise} \end{cases} , \quad (6)$$

and $\tilde{\phi}_j(t)$ is the slope of $S_j(t)$:

$$\tilde{\phi}_j(t) = \begin{cases} \frac{\partial S_j(t)}{\partial t} \frac{1}{r} : S_j(t) \text{ is differentiable} \\ 0 \quad : \qquad \text{otherwise} \end{cases} . \quad (7)$$

Note that we adopt the convention that if no $S_j(t)$ is differentiable at a specific t, then the overall value of the integrated quantity is 0 (at that specific t).

Issue I3) is handled by setting $\phi_i = \infty$ for the packets of the session which correspond to the sudden "burst". As (1) and (2) suggest, if $\phi_i = \infty$, the virtual time and the finishing time of the session are not updated.

Our DTV-PFQ system can be implemented using the algorithm[3] in Fig. 4. Our algorithm extends the algorithms of [6, 8] by addressing I1), I2), and I3), while maintaining the same complexity as the algorithm of [8].

To illustrate the operation of our algorithm, we assume a system which supports two sessions ("1" and "2"). We simulate the system for 20 slots, and we make both sessions transmit at rate $2r$ (to keep them both continuously backlogged, which allows us to study the bandwidth allocation). Session "1" starts transmitting at slot 0, whereas session "2" starts transmitting at slot 4. Fig. 5 and 6 illustrate the bandwidth allocation under SCED and under DTV-PFQ. We assume that $b_1 = 25\%$, $b_2 = 75\%$, $T_1 = 10$, $T_2 = 5$. As Fig. 5 depicts, session "1" does not receive any service at all in [4,11], being penalized for the extra bandwidth it received in [0, 9]. On the other hand, DTV-PFQ does not penalize session "1" (Fig. 6), because session "1" still receives service in [4, 11]. Under both schedulers, we observe that in the long run sessions "1" and "2" receive respectively 25% and 75% of the bandwidth; but it is in the transient that DTV-PFQ performs better than SCED. It can be seen, however, that there is a small punishment for session "1": this comes as a result of the inherent trade-off between fairness and SC provision (see also, e.g., [13, pg. 253]) and the approximation errors induced by the packet-based implementation [11].

V. Conclusions

In this paper we have presented a deterministic time-varying weight assignment procedure for PFQ-based switching systems. By supporting piecewise linear SCs, our scheme dispenses with the *delay-bandwidth coupling* and targets integrated services networks. Unlike existing SC based algorithms, our time-varying PFQ scheme does not exhibit the *punishment* phenomenon and allows sessions to exploit the extra bandwidth in under-loaded networks. Future research avenues include the study of stochastic time-varying weight assignment procedures which take into account the probabilistic description of incoming traffic (for wired networks), and the varying channel capacity (in wireless networks).

[3] Further details, proofs, and simulation examples can be found in [11].

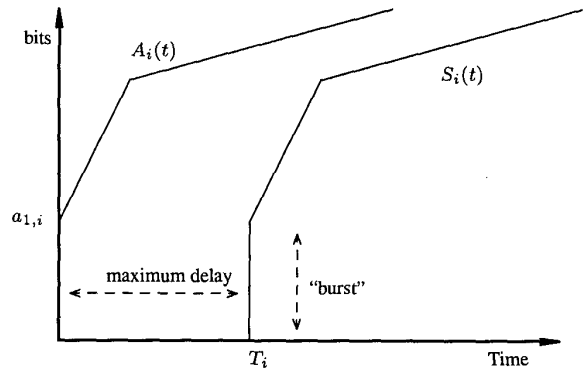

Fig. 1. Envelope and Service Curve

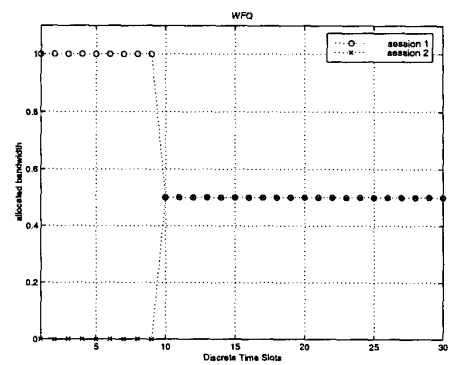

Fig. 2. Bandwidth Allocation under GPS

Acknowledgement

We would like to thank the anonymous reviewers for their constructive comments.

References

[1] C-S. Chang and K-C. Chen. Service curve proportional sharing algorithm for service-guaranteed multiaccess in integrated-service distributed networks. In *Proc. of GLOBECOM*, pages 1340–1344, In *Proc. of GLOBECOM*, Rio de Janeiro, Brazil, December 1999.

[2] R. L. Cruz. A calculus for network delay, part I: network elements in isolation. *IEEE Transactions on Information Theory*, 37(1):114–131, January 1991.

[3] R. L. Cruz. Quality of service guarantees in virtual circuit switched networks. *IEEE Journal on Selected Areas in Communications*, 13:1048–1056, 1995.

[4] J. Liebeherr, D. Wrege, and D. Ferrari. Exact admission control for networks with a bounded delay service. *IEEE Transactions on Networking*, 4:885–901, December 1996.

[5] H-T Ngin, C-K Tham, and W-S Sho. Generalized minimum queueing delay: An adaptive multi-rate service discipline for atm networks. In *Proc. of INFOCOM99*, pages 398–404, Piscataway, NJ, USA, 1999.

[6] A. K. Parekh and R. G. Gallager. A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-

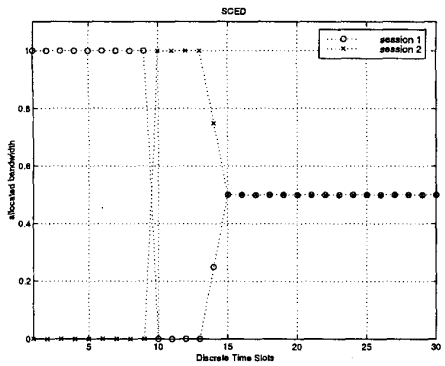

Fig. 3. Bandwidth Allocation under SCED

Initialize: $\delta_{i,k} \leftarrow 0, \delta^o_{i,k} \leftarrow -a_{i,k}/b_{i,k}$,
$1 \le i \le N, 1 \le k \le K_i$.

In each slot u during which packets have arrived:
$\phi_i \leftarrow 0, i = 1 \ldots N$
for $i = 1 \ldots N$ do
  if $A_i^{\text{in}}[u] \neq 0$ /* packets from i arrived in slot u */
    $\delta_{i,k} \leftarrow \max\{\delta_{i,k}, u - 1 + \delta^o_{i,k}\}$
    for $l = 1 \ldots A_i^{\text{in}}[u]$ do
      $\delta_{i,k} \leftarrow \delta_{i,k} + 1/b_{i,k}, 1 \le k \le K_i$
      $k = \arg \max_{1 \le k \le K_i} \delta_{i,k}$
      if $u \le \delta_{i,k}$
        then $\phi_i \leftarrow 0$
        else $\phi_i \leftarrow b_{i,k}/r$
      endif
      if $l = 1$ and session was not backlogged
        then $F_i = V + \text{VirtualOffset}_i$
        else if $\phi_i \neq 0$
          then $F_i = \max\{F_i, V\} + 1/\phi_i$
        endif
      endif
      assign the deadline F_i to the packet
    endfor
  endif
endfor
$\text{sum} = \sum_{i=1}^{N} \phi_i$
if sum $\neq 0$
  then $V \leftarrow V + \frac{r}{\text{sum}}$
endif

Fig. 4. Algorithm for deadline assignment

Node Case. *IEEE/ACM Transactions on Networking*, 1(3):344–357, June 1993.

[7] D. Saha, S. Mukherjee, and S. K. Tripathi. Multirate scheduling of vbr video traffic in atm networks. *IEEE Journal on Selected Areas in Communications*, 15(6):1132–1147, August 1997.

[8] H. Sariowan, R.L. Cruz, and G. G. Polyzos. SCED: A Generalized Scheduling Policy for Guaranteeing Quality-of-Service. *IEEE Transactions on Networking*, 7(5):669–684, October 1999.

[9] H. Sariowan, R.L. Cruz, and G.C. Polyzos. Scheduling for quality of service guarantees via service curves. In *Proc. of ICCCN95*, pages 512–520, In *Proc. of ICCCN*, Las Vegas, Nevada, September 1995.

[10] A. Stamoulis and G. B. Giannakis. Packet Fair Queueing Scheduling Based on Multirate Multipath-Transparent CDMA for Wireless Networks. In *Proc. of INFOCOM2000*, pages 1067–1076, Tel Aviv, Israel, March 2000.

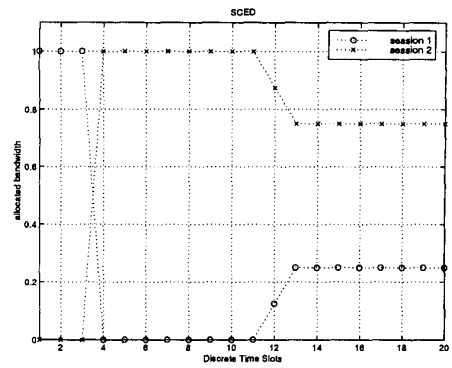

Fig. 5. Bandwidth Allocation under SCED

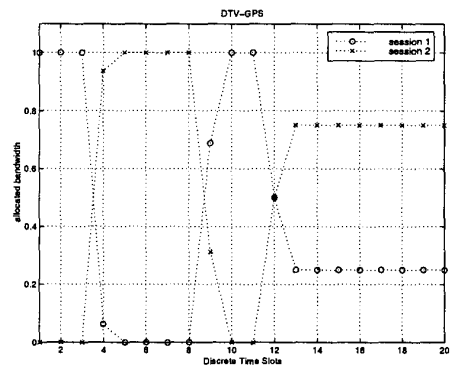

Fig. 6. Bandwidth Allocation under DTV-PFQ

[11] A. Stamoulis and G.B. Giannakis. Deterministic Time-Varying Packet Fair Queuing for Integrated Services Networks. *Journal of VLSI Signal Processing*, 2001 (to appear).

[12] D. Stiliadis and A. Varma. Latency-rate servers: a general model for analysis of traffic scheduling algorithms. *IEEE Transactions on Networking*, 6(5):611–624, October 1998.

[13] I. Stoica, H. Zhang, and T.S.E. Ng. A hiearchical fair service curve algorithm for link-sharing, real-time and priority services. In *Proc. ACM Sigcomm'97*, pages 249–262, 1997.

[14] R. Szabo, P. Barta, J. Biro, F. Nemeth, and C-G. Perntz. Non rate-proportional weighting of generalized processor sharing schedulers. In *Proc. of GLOBECOM*, pages 1334–1339, In *Proc. of GLOBECOM*, Rio de Janeiro, Brazil, December 1999.

[15] D.E. Wrege, E.W. Knightly, H. Zhang, and J. Liebeherr. Deterministic delay bounds for vbr video in packet-switching networks: Fundamental limits and practical tradeoffs. *IEEE Transactions on Networking*, 4:352–362, June 1996.

[16] Z.-L. Zhang, D. Towsley, and J. Kurose. Statistical Analysis of Generalized Processor Sharing Scheduling Discipline. *IEEE Journal on Selected Areas in Communications*, 13(6):1071–1080, August 1995.