The Priority Token Bank in a Network of Queues

(to appear in Proceedings of ICC '97)

Mark A. Lynn
Delphi-Packard Electric Systems
408 Dana St. MS 93L, Warren, OH 44486,
Phone: (330) 373-4038, Fax: (330) 373-2515
email: mark.lynn@ped.gmeds.com.

Jon M. Peha
Carnegie Mellon University
Department of ECE, Pittsburgh, PA 15213-3890
Phone: (412) 268-7126, Fax: (412) 268-2860
email: peha@ece.cmu.edu
http://www.ece.cmu.edu/afs/ece/usr/peha/peha.html

Abstract
This paper takes a known approach for scheduling and admission control in integrated services networks[1], the Priority Token Bank (PTB), whose mechanism and performance have been studied in a single queue, and extends the algorithm to address the challenges of implementing it in a network of queues. An integrated services network must handle traffic streams with disparate arrival processes and performance objectives, and efficiently supporting these streams becomes more difficult when performance guarantees must be met across a number of switches and when a stream's burstiness can dramatically increase from source to destination. Three variations on the basic mechanism are proposed to address this problem: PTB-Separate Classes, PTB - Minimum Interdeparture Time, and PTB-Weighted Fair Queuing. It is shown that the PTB scales well to a network of queues, and that, under the constraint that performance requirements for guaranteed traffic such as packet voice, HDTV, and packet video must be met, the mean delay experienced by other traffic is much better with the Priority Token Bank mechanisms than with other proposed algorithms.

1.0 Introduction
Efficiently supporting diverse traffic types is challenging when each stream has different arrival processes and different performance objectives ,or *Quality of Service* (QoS) requirements. The arrival process of traffic streams at the network access point is declared during the admission process, and this information can be used to more efficiently handle traffic. In a network of queues, the arrival process of cells at downstream switches may be very different from the arrival process at the edge of the network, and traffic control algorithms must account for a much wider range of arrival processes. Since a cell carries no information about the queuing delay it has experienced, the scheduling algorithm must make assumptions about the delay experienced by cells at previous switches, leading to potentially less efficient handling of traffic. If traffic streams are combined at downstream switches, the scheduling algorithm must also maintain performance guarantees when there are cells interspersed in the same queue which may have experienced very different delays.

This paper describes a traffic control algorithm, the Priority Token Bank, whose mechanism and performance in a single queue has been analyzed[1], and compares its performance in a network of queues to that of other proposed algorithms. Related work in the area of traffic control is discussed in section 2. The Priority Token Bank mechanism as used in a single queue is reviewed in section 3, and three variations on the basic algorithm for use in a network of queues are proposed in section 4. In section 5, the performance of the three algorithms is compared, in a network of queues, with the performance obtained by using other common algorithms, and, the paper is summarized in section 6.

2.0 Related Work
The simplest scheduling algorithms that can provide performance guarantees in a network of queues are *static priority* (SP) and *earliest deadline first* (EDF), but neither is capable of efficiently handling traffic with diverse performance objectives. *Polling* algorithms generally guarantee that cells from each class receive some service during a fixed interval, or frame, or that a fixed number of cells can be transmitted in each frame, without specifying where in the frame they are transmitted. If a stream has no cells to send, the unused portion of the frame may be utilized by another stream[2,3], the frame may be shortened, effectively sharing the unused bandwidth among all streams with queued cells[4] or the excess bandwidth may remain unused, creating a non-work-conserving mechanism[5]. While polling algorithms insure that bandwidth is fairly distributed, they do little to optimize system performance. A sophisticated algorithm, called *MARS*[6], has been proposed to run on top of Magnet's polling-based structure. In MARS, each class is permitted to transmit cells in a portion of a frame, and the fraction of a frame dedicated to each traffic class can change as needed to maximize total system performance. MARS has been shown to be an effective approach when all traffic meets its assumptions[1], but it is not clear how the mechanism can be expanded to handle traffic with more diverse performance objectives. *Occupancy*-based[7] algorithms calculate the oldest class i cell's transmission priority as a function of the number of cells from stream i currently queued. Occupancy works well when there is a tight correlation between the number of cells in the queue and the queuing delay experienced by the cell at the head of the queue, but with bursty arrival processes the correlation begins to break down. *Cost Based Scheduling* (CBS)[8] associates a cost function, $c_i(t)$, with each cell that defines a cost incurred as a function of its queuing delay, t. Cost functions are selected to represent application performance objectives. CBS then determines priority as a function of

queuing delay t to minimize total cost. CBS has been shown to perform well[8], and near optimal where an efficient optimal algorithm is known.[9] However, implementation is complex.

3.0 The Priority Token Bank

Our approach to the traffic control problem is the *Priority Token Bank*[1] (PTB). In the single queue implementation, cells are divided into N classes, where all cells in a same class have similar performance objectives, though they may be from different streams. For each class i, there is a first in, first out (FIFO) queue, a token counter T_i, and a token counter update period τ_i. Also each class has a priority function $P(T_i)$ which gives the transmission priority of the cell at the head of the class i queue. T_i, is held at zero whenever queue i is empty. When cells are present, T_i is incremented every τ_i and decremented each time a class i cell is transmitted, so that the token counter value will approximate the queuing delay of the oldest class i cell. T_i can become negative, indicating that stream i has temporarily used more than its share of bandwidth. The next cell to be transmitted is the cell at the head of the queue with the highest transmission priority.

With proper parameter selection, the transmission of cells not in danger of violating their performance objectives can be delayed, providing better performance to traffic classes which provide greater value through early cell arrival. Let G be the maximum priority for low priority traffic. A class i cell from a guaranteed stream is permitted to have priority greater than G only if $T_i \geq L_i$, where L_i is the token counter limit. If $T_i = L_i$, we are in danger of violating the class i performance requirements, so its priority increases. Parameters for all streams must be chosen such that all class i cells with priority greater than G are transmitted within τ_i transmission times, so that T_i never exceeds L_i. Performance objectives for guaranteed traffic will be met by letting P_i equal some constant > G when $T_i > L_i$.

To select τ_i, L_i, and $P_i(T_i)$ to meet performance objectives in a single queue, we use a stream's arrival process parameters declared during negotiation for admission to the network. If an arriving stream is policed with a Leaky Bucket, the arrival process can be characterized by using parameters σ_L and ρ_L, where the amount of traffic arriving in a period of duration t is bounded by $\sigma_L + \rho_L t$ [10], where ρ_L is a measure of the average arrival rate and σ_L is a measure of burstiness. If cell arrivals are periodic, such as for CBR voice streams, we would choose $\tau_i = 1/\rho_L$ and set $L_i = \rho_L M - \sigma_L = \rho_L M - 1$. The same equations apply for VBR video; ρ_L would be set to the peak sustainable arrival rate and σ_L would be set large enough to accommodate the potential burstiness in the arriving stream.

It has been shown that, in a single queue[1], PTB provides QoS guarantees while providing performance that is significantly better than that provided by static priority and polling-based mechanisms, and nearly as good as CBS. Extending the mechanism to provide guarantees over a network of switches presents significant challenges. The most critical is that a stream's arrival processes change significantly after traversing several hops. Any algorithm that delays transmission of cells not in danger of violating their performance guarantees, increasing stream burstiness. Consider the case where a switch does not transmit class 1 cells until they are in danger of missing their deadline, building up a large queue of class 1 cells. When suddenly there are no other cells to transmit, a burst of cells are transmitted. To maintain performance guarantees, an algorithm must be able to handle this increased burstiness at downstream switches. All of the sophisticated algorithms, to which we will compare the PTB, do not handle the increased burstiness well, though MARS fares the worst. MARS divides traffic into three classes, requiring that the class 1 load be less than 1 during any period of duration d1, where d1 is the maximum tolerable delay for class 1 packets. A similar requirement exists for class 2 traffic as well. Consider the network in Figure 1. Let $d1^{(1)}$ be the maximum class 1 delay at each hop 1 switch, and $d1^{(2)}$ be the maximum class 1 delay at the hop 2 switch. If the input to the class 1 queue at each of the n hop 1 switches is constant bit rate traffic at a load of ρ, then the load from each of these streams at switch n+1 can be as large as $\min\left(1, \left\lceil \rho + \frac{d1^{(1)}}{d1^{(2)}} \rho \right\rceil\right)$. Thus, the class 1 load must be kept very low for MARS to work; if $d1^{(1)} = d1^{(2)}$ the class 1 load can be no higher than 0.5. If the input stream was bursty, or the source-destination path was longer than two hops, the problem would be worse. Algorithms such as CBS and occupancy do not have the same explicit upper bound on load, but they still experience problems with the increase in burstiness described.

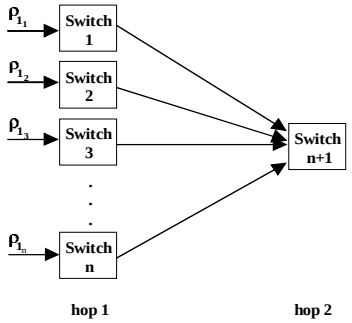

Figure 1: Multiplex Network Configuration

4.0 The Priority Token Bank in a Network of Queues

To allow performance guarantees over a network of switches, the basic mechanism must be modified to either handle or limit the increased stream burstiness seen in a network. We propose three variations: PTB - Separate Classes, PTB - Minimum Interdeparture Time, and PTB - Weighted Fair Queuing.

4.1 PTB - Separate Classes (PTB-SC)

If all the cells that have traveled along the same path through the network are combined, the basic PTB mechanism will handle the increased burstiness. If, in Figure 2, all cells leaving

switch 1 are maximally delayed, stream burstiness is not increased, so guarantees made by the single queue PTB mechanism still hold at switch 2. Stream burstiness may increase only if some cells are transmitted ahead of schedule from switch 1. In that case it is permissible for a cell to experience a delay longer than its delay allocation in switch 2 if the total delay experienced in the first two switches is less than $M_i^{(1)} + M_i^{(2)}$, where $M_i^{(k)}$ refers to the delay allocation at the kth hop along the stream i source-destination path. The excess delay experienced at hop 2 is no greater than the amount of time that cell was transmitted early from the previous switch. Let cell 0 be a cell that arrives to an empty class i queue at switch 1. Cell k: k>0 is defined to be the cell that arrives after cell k-1 while cell k-1 is still queued. The maximum delay, $D_k^{(1,N)}$, experienced by cell k from switch 1, to its destination beyond switch N is bounded by

$$D_k^{(1,N)} \le \left(\sum_{j=1}^{N} \left(L_i^{(j)} + 1 \right) * \tau_i \right) + \sigma_L + \sum_{j=2}^{N} P^{(j)} . \quad L_i^{(j)} :$$

$1 \le j \le N$, is the value of the class i token counter limit, L_i, at the jth switch, τ_i is the class i token counter update period at each switch, σ_L is the burstiness permitted by the leaky bucket policing mechanism at the network entry point, and $P^{(j)}$: $2 \le j \le N$ is one cell transmission time plus the propagation delay from switch j-1 to switch j.

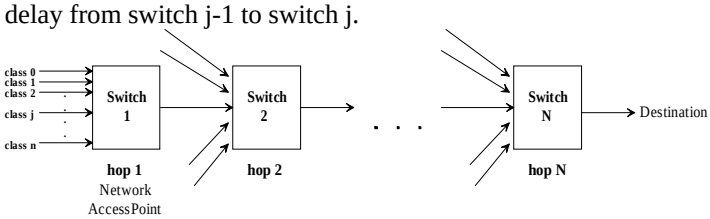

Figure 2: Straight - Through Network Configuration

The penalty PTB-SC pays for its ability to handle the increase in stream burstiness is an increased number of classes, increasing the number of FIFOs, the complexity, and the cost, although some ATM switch manufacturers are considering using a separate FIFO for each virtual channel, which is much more complex than one FIFO per virtual path. For traffic such as voice, where the individual stream bandwidth is small and the number of potential classes is very large, PTB-SC may not be practical. For traffic such as High Definition Television where individual streams consume a large amount of bandwidth, limiting the number of classes, PTB-SC is an excellent approach.

4.2 PTB-Minimum Interdeparture Time (PTB-MI)

PTB-MI combines into one class cells of the same traffic type which have traveled different paths and prevents stream burstiness from increasing by enforcing a minimum interdeparture time, m_i, between class i cells.. A class i cell with the highest priority is transmitted only if a minimum time, m_i, has elapsed since the last class i transmission, even if no other traffic is present in the switch. This technique, proposed in other scheduling algorithms,[5] creates a non-work-conserving mechanism, which provides inherently worse performance than one that always transmits a cell when it is possible to do so. Performance can be improved by reducing m_i, but that requires more bandwidth to be reserved. Let voice streams arrive at each of the n hop 1 switches in Figure 1 with a load of ρ and let one switch transmit maximally delayed cells, while the other n-1 switches transmit cells at the peak server rate. Switch n+1 must reserve bandwidth equal to $\rho + \rho \frac{\tau}{m_i}(n-1)$, instead of $n\rho$, if $\tau^{(j)} = \tau_i \ \ \forall j : 1 \le j \le n$. The minimum interdeparture time, m_i, can range from $1 \le m_i \le \tau_i$, with the least bursty output stream is generated by letting the minimum interdeparture time equal the token counter update constant, τ_i .

4.3 PTB-Weighted Fair Queuing (PTB-WFQ)

When streams are combined and cells from different streams are served first-come, first-served, burstier streams may get served too quickly, at the expense of other streams. If all streams are served regularly this can be prevented. Let all streams of the same traffic type with similar performance objectives be considered to be the same class. All cells which traveled the same path through the network are combined into a class, and each class is assigned a separate FIFO queue.. All FIFOs from this class would share the same token counter and priority function. When this class has the highest transmission priority, a cell is chosen from those at the head of each populated FIFO based on a Weighted Fair Queuing algorithm[11], which assigns polling weights based on the average arrival rate of each stream. The PTB-WFQ mechanism is perfectly efficient if the cells at the head of all FIFOs are equally urgent. When some have been maximally delayed and others have just arrived, as is more probable with very bursty streams, those less urgent cells must be transmitted along with their maximally delayed counterparts. PTB-WFQ is slightly less efficient than the PTB-SC mechanism, but should provide better performance than the non-work-conserving PTB-MI. The implementation complexity of PTB-WFQ is lower than that of PTB-SC, since the switch must update and compare priorities for fewer classes, but is greater than that for PTB-MI.

5.0 Results

In this section, we compare the performance of each PTB mechanism with simple algorithms such as static priority (SP), a work-conserving polling mechanism (POLL-W)[4] , and a non-work-conserving mechanism (POLL-N)[5], as well as with more sophisticated algorithms including an occupancy based algorithm (OCC)[7], Cost Based Scheduling (CBS)[8], and MARS[6]. In a single queue, the Priority Token Bank has been demonstrated to compare very favorably to each[1]. Here we compare their performance in a network of queues. A fluid flow approximation is assumed, and two types of guaranteed traffic are considered: packet voice and packet video. In some scenarios, exact analytical results can be obtained[12] while for others efficient simulation tools are used[13]. For all simulation

results, the 95% confidence interval is, at worst, within 5% of the values shown. Each scenario assumes a link bandwidth of 150Mbps, and two traffic classes: class 1, a guaranteed stream with an end-to-end maximum delay requirement, and class 2, a guaranteed or best effort stream with end-to-end mean delay objectives. Class 2 consists of data bursts of exponentially distributed length with mean β=500Kb that arrive according to a Poisson process. In all the scenarios discussed, independent class 2 streams arrive at each switch.

Each voice stream is modeled as many independent streams multiplexed so the combined stream has a constant bit rate. Each stream has a maximum end-to-end queuing delay and 0% loss rate requirement. For a HDTV stream, we use a model[14] which approximates the arrival process as the sum of 10 independent on/off sources, with exponentially distributed active and inactive times with means of 386ms and 765ms respectively. We scale the data rate of each source so that the mean rate of the combined sources is 1Mbps.

First, we examine the impact of combining guaranteed traffic streams that have previously traveled different paths, by comparing the performance results of the PTB mechanisms with SP, POLL-W, and POLL-N. As described, this scenario poses significant problems for the more sophisticated algorithms, so MARS, OCC and CBS are not compared here. (although we will include them later, in a scenario where this problem does not occur). Each scenario uses the network model in Figure 1. The allowable end-to-end queuing delay for voice and video traffic is 30ms, with equal delay allocation at hop 1 and hop 2. The class 1 load at each hop 1 switch is $\rho_1^{(j)} = \rho_1$, $1 \le j \le n$, and $\rho_1^{(n+1)} = n\rho_1$, where $\rho_1^{(j)}$ is the class 1 load at switch j and n is the number of first hop switches. Each switch receives an independent bursty traffic stream with load = 0.4. Figure 3 shows the queuing delay of class 2 traffic at switch 3, $Q_2^{(3)}$, as a function of the class 1 load, $\rho_1^{(3)}$, at switch 3, where class 1 is packet voice. At the first hop switches, performance is equivalent to the single queue scenarios discussed elsewhere[1] so the results are not repeated. The results at switch 3 show that with PTB-SC, significantly smaller class 2 delays are achieved for the same guaranteed load than with SP, POLL-W, POLL-N, and the PTB-MI alternatives, but PTB-WFQ provides results that are very nearly as good. Because the burstiness of the streams arriving at switch 3 is low, the occurrence of one FIFO emptying while the other remains populated is rare, so PTB-WFQ is relatively efficient. If we decrease m_i, PTB-MI is also able to delay class 1 packet transmissions somewhat, and its performance improves as we decrease MI. For minimum interdeparture times of 0.5*τ, PTB-MI approaches the PTB-WFQ results. It has been shown[15], using the same scenario, PTB performs equally well when class 1 traffic is packet video or HDTV traffic. In each case, PTB-SC provides the best performance, although PTB-WFQ and PTB-MI (with small values of m_i) provide performance nearly as good.

The results presented thus far show that the PTB mechanisms can provide significantly improved performance for class 2 traffic while handling the same load of class 1 traffic as other algorithms. However, the network designer is concerned with the amount of traffic a network can support and still meet each applications performance requirements. We can illustrate how much more traffic the PTB mechanisms can support by using a *schedulable region* performance measure. A schedulable region shows, for a given set of traffic types and performance requirements, the range of loads for which the scheduling algorithm is able to provide performance guarantees, i.e., a point (x,y) is within the schedulable region if performance objectives are met with a class 1 load of x and a class 2 load of y. Figure 4 shows the schedulable region when class 1 traffic is packet video with 0% loss rate objectives and end to end tolerable delays of 30ms, and when class 2 traffic is bursty traffic which has an end-to-end mean delay objective. PTB-SC supports the greatest load of class 2 traffic for a given class 1 load, but again PTB-WFQ provides performance nearly as good. As we decrease m_i, the PTB-MI schedulable region increases significantly, although the incremental improvement lessens the further we decrease m_i. Each of the PTB mechanisms significantly expands the schedulable regions obtained with POLL-W, POLL-N, and SP.

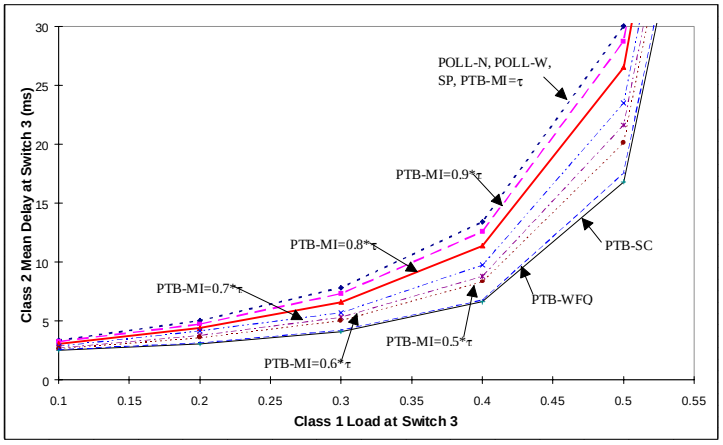

Figure 3: Multiplex Scenario-
Class 1: Packet Voice, Class 2: Bursty Traffic

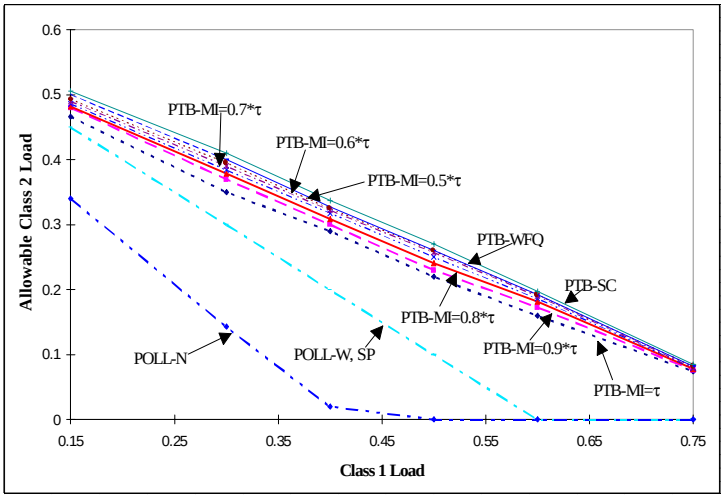

Figure 4: Schedulable Region,
Class 1: Packet Video, Class 2: Bursty Traffic

Finally, we examine a scenario where we track the class 2 mean delay obtained at each switch along a long packet voice source-destination path. At each switch, class 1 traffic arrives at load ρ_1=0.3, with a maximum tolerable class 1 delay of 5ms at each switch. The class 2 load at each switch is ρ_2=0.4. At the first switch, performance is identical to that shown in the single queue analysis[1]; CBS, MARS, PTB-SC, and PTB-WFQ all provide similar performance. However, PTB-SC, and PTB-WFQ to a lesser degree, can take advantage of the fact that a voice cell was transmitted early from previous hops to delay it beyond its allocation at the current hop and give better performance to class 2 traffic. As a result, with PTB-SC and PTB-WFQ class 2 mean delay decreases as more hops are traversed. Note that after 10 hops, they provide virtually identical performance. As we decrease m_i, PTB-MI gains this ability as well, and the class 2 mean delay begins to decrease as more hops are traversed. MARS, CBS, and OCC all increase stream burstiness with each hop, and as the burstiness increases, the class 2 mean delay increases. CBS causes the greatest increase in voice stream burstiness, so it experiences the greatest increase in class 2 mean delay. Thus, for source-destination paths which traverse a large number of hops, PTB's performance advantage is especially significant.

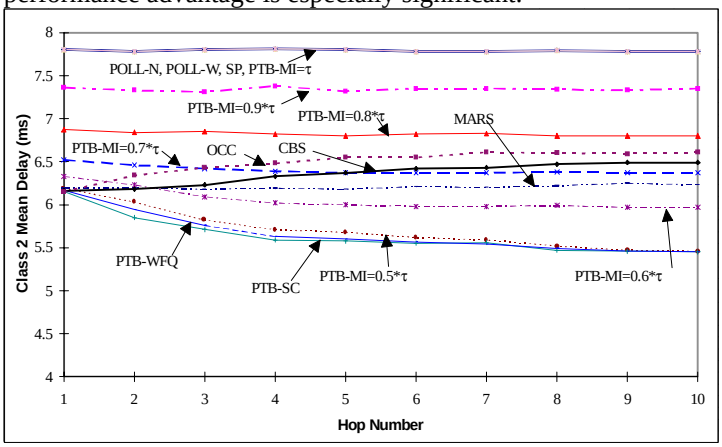

Figure 5: Path Length Scenario - 10 Hops,
Class 1: Packet Voice, Class 2: Bursty Traffic

6.0 Conclusion

This paper has proposed three mechanisms for extending the Priority Token Bank mechanism to provide performance guarantees in a network of queues: PTB - Separate Classes, PTB - Weighted Fair Queuing, and PTB - Minimum Interdeparture Time. The performance achieved with each mechanism was compared to alternative algorithms including static priority, several polling approaches, MARS, occupancy scheduling, and Cost Based Scheduling, using two types of guaranteed traffic packet voice and packet video. We have demonstrated that, for these traffic types, the Priority Token Bank provides better performance in a network of queues than was obtained with *any* of the other algorithms compared, and the PTB is the only algorithm for which performance actually improves in a network of queues. Similar results were found[15] for other scenarios and other traffic types.

For all traffic types discussed, PTB-SC provides the best performance of the PTB mechanisms, but for traffic such as packet voice or video, where the number of potential classes is large, PTB-WFQ and PTB-MI (for smaller minimum interdeparture times) offer reduced implementation complexity with minimal drop in performance. PTB-MI offers reduced complexity over PTB-WFQ, but can only achieve comparable performance if it is possible to reserve extra bandwidth. While the PTB is a sophisticated scheduling algorithm which does scale very well to a network of queues (in fact perhaps performing even better than in the single queue analysis), we demonstrated that other sophisticated algorithms (such as CBS, MARS, and occupancy) do not scale well, especially in those scenarios where streams are multiplexed. Less sophisticated algorithms, such as SP and Polling, which are often chosen because it is believed that sophisticated scheduling is not possible in a network of queues, are even less effective.

References

[1] J.M. Peha, "The Priority Token Bank: Integrated Scheduling and Admission Control for an Integrated Services Network," in Proc. IEEE ICC, Geneva, Switzerland, May 1993, pp.345-51. Expanded version submitted for journal publication.

[2]C.R. Kalmanek, H. Kanakia, and S. Keshav, "Rate Controlled Servers for Very High-Speed Networks,", in *Proc. IEEE Globecom*, Dec. 1990, pp. 12-20.

[3]K. Kummerle, "Multiplexer Performance for Integrated Line and Packet-Switched Traffic," in *Proc. ICCC*, 1974, pp.517-23.

[4]A.K. Parekh and R.G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services - the Single Node Case," *IEEE/ACM Trans. on Networking*, vol. 1, no. 3, pp. 344-57, June 1993.

[5]S. J. Golestani, "A Stop-and-Go Queuing Framework for Congestion Management," in *Proc. ACM Sigcomm*, Sept. 1990, pp. 8-18.

[6]J. Hyman, A.A. Lazar, G. Pacifici, "Joint Scheduling and Admission Conrtol for ATS-based Switching Nodes," in Proc. ACM Sigcomm, Sept. 1992, pp.223-34.

[7]D. Lee and B. Sengupta, “Queuing Analysis of a Threshold Based Priority Scheme for ATM Networks,” *IEEE/ACM Trans. on Networking*, vol.1, no. 6, pp. 709-17, Dec. 1993.
[8]J.M. Peha and F.A. Tobagi, “Cost-Based Scheduling and Dropping Algorithms to Support Integrated Services,”. *IEEE Trans. on Comm.*, Vol. 44, no. 2, Feb. 1996, pp. 192-202..
[9]J.M.Peha, “Heterogeneous-Criteria Scheduling: Minimizing Weighted Number of Tardy Jobs and Weighted Completion Time,” *Computers and Operations Research*, vol. 22, no. 10, Dec. 1995, pp. 1089-1100.
[10]R. L. Cruz, “A Calculus for Network Delay, Part 1: Network Elements in Isolation” *IEEE Trans. Info. Theory*, Vol. 37, no. 1, Jan. 1991, pp. 114-31.
[11]A. Demers, S. Keshav, and S. Shenker, “Analysis and Simulation of a Fair Queuing Algorithm,” *Proc. ACM Sigcomm-89*, Sept. 1989, pp. 1-12.
[12]J.M. Peha, “Analysis of Scheduling Algorithms for Integrated-Services Networks using a Semi-Fluid-Flow Model,” *Proc. IEEE Globecom*, Dec. 1992, pp. 330-4.
[13]J.M Peha, “Simulating ATM Integrated-Services Networks,” *Proc. 29th Annual IEEE/ACM/SCS Simulation Symp.*, Apr. 1996, pp. 162-71.
[14]B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, and J.D. Robbins, “Performance Models of Statistical Multiplexing in Packet Video Communications,” *IEEE Trans. Commun.*, Vol. 36, no. 7, pp. 834-44, July 1988.
[15]M. Lynn, M.S. Thesis, Carnegie Mellon University, Dept. of ECE, 1997.