A Scheduling Method for Bounded Delay Services in High Speed Networks

Xinwei Hong(*) Hao Che(**) Zailu Huang(*)

(*) Department of Electronic and Information Engineering
HuaZhong University of Science and Technology
Wuhan, Hubei, (430074) P.R.C

(**)The Department Electrical Engineering
Pennsylvania State University
University Park, PA 16802-2701

Abstract- As one of the most demanding applications in high speed networks, real-time services, such as audio and video, have stringent requirements on quality of service (QoS) guarantee, especially the bounded end-to-end delay and delay variations. Accordingly, the real-time services with deterministically bounded delay have been widely studied. The key is to design efficient scheduling algorithms. In this paper, a scheduling method, referred to as static-rotating-priority-queues (SRPQ), is proposed. Exact schedulability conditions for the method, which is important for call admission control, is also presented. The most desirable features of this method are its low complexity and reduced number of first-in-first-out (FIFO) queues in a scheduler while providing high bandwidth utilization.

I. INTRODUCTION

To support real-time services, which are expected to be one of the most demanding services in future high speed networks, steps must be taken to guarantee the Quality of Service (QoS) [1,2], especially, the end-to-end delay and delay variation requirements. Generally, the end-to-end packet delay includes processing delay (packetizing, unpacketizing delay, etc.), propagation delay, as well as queueing delay. Since the processing delay and propagation delay, which result from physical or technological constraints, are generally fixed [3], the design of a bounded delay service focuses on the study of queueing delay. Finding appropriate queue scheduling techniques has been considered as an important design aspect [4,5]

Several scheduling techniques, such as the first-come-first-service (FCFS), earliest-deadline-first (EDF) and static-priority (SP) [1,2], have been studied. Each method presents a particular tradeoff in satisfying the requirements of efficiency, flexibility, complexity, analyzability, as well as impartiality [3]. The FCFS method, which is the simplest one, is very limited because it guarantees only one delay bound for all services in scheduling. The EDF method always selects the packet, which has the shortest deadline. Hence it can achieve the highest bandwidth utilization. Another advantage of the EDF method is that it can satisfy all kinds of delay bound requirements. However, since a sorting operation must be carried out for every incoming packet, the EDF method suffers from high computational complexity. The SP method is easy to implement but offers low bandwidth utilization. Recently, a rotating-priority-queues (RPQ) method was proposed to overcome the shortcomings of the above methods [6]. A RPQ scheduler consists of a set of ordered FIFO queues with variable priority. It always selects packets in the queue with the highest priority first. The priority of each queue will be rotated once after a rotation interval. The RPQ method not only simplifies the implementation, but also achieves the maximum bandwidth utilization when decreasing a design parameter--the rotation interval. Unfortunately, a small rotation interval may result in a large number of FIFO queues, which greatly increases the complexity of the method. Achieving high bandwidth utilization at the cost of high complexity undermines the performance of the method, especially, when the delay bounds of the scheduled real-time services differ greatly. For example, a two-way voice or some urgent control messages of a real time system may have stringent delay bounds, whereas the delay bounds of a one way video can be relatively loose. When these services are scheduled in a RPQ at the same time, it will be very difficult to achieve satisfactory performance. To overcome the shortcomings of the RPQ method, we present, in this paper, a scheduling method, called static-rotating-priority-queues (SRPQ).

The SRPQ scheduler consists of several ordered RPQ queues with fixed priorities while the priority of the queues in every RPQ queue is updated according to the RPQ scheduling policy. The advantage of the SRPQ method over RPQ method lies in the fact that the number of FIFO queues used in the SRPQ reduces greatly while offering similar bandwidth utilization as the RPQ method. As a result, the SRPQ method can reduce the complexity of the scheduler and can be easily implemented.

The remainder of this paper is organized as follows. In Section II, we introduce the principle of the SRPQ method. In Section III, we give and also prove the necessary and sufficient schedulability conditions of the SRPQ method. In Section IV, we present an empirical evaluation of the SRPQ method with reference to the EDF, SP, and RPQ methods. Finally, a conclusion is given in Section V.

II. SCHEDULING PRINCIPLE

According to the delay bounds, we classify all incoming services into N disjoint groups, $C_1, C_2, \cdots, C_N$, and let the services that have close delay bounds be in the same group. The smaller the delay bound, the lower the index value, and the higher the priority of the service. All services belonging to a specific group C_n will enter the n-th RPQ queue for further processing. We also require that the packets in lower priority RPQs be served if and only if there are no packets in the higher priority RPQs. Furthermore, we partition services

0-7803-6283-7/00/$10.00 © 2000 IEEE

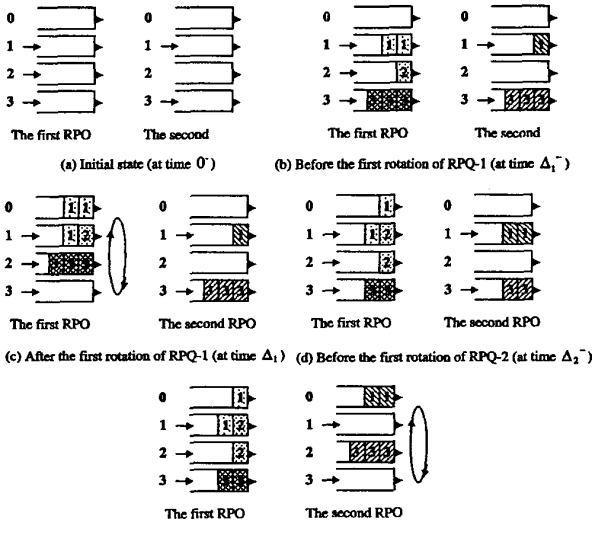

Fig.1 Example of SRPQ Scheduling

belonging to C_n into P_n categories, $C_n^1, C_n^2, \cdots, C_n^{P_n}$, where P_n varies with n. The services in the same category will have identical delay bounds. All delay bounds supported by the n-th RPQ scheduler are integer multiples of the n-th rotation interval Δ_n ($\Delta_n > 0$), which is a design parameter, i.e., $d^{n,p} = k^{n,p}\Delta_n$ for all services belonging to category C_n^p where $k^{n,p} > k^{n,q}$ if $p > q$ and $n > 0$. Thus, the n-th RPQ scheduler will maintain $k^{n,P_n} + 1$ FIFO queues and the length of each FIFO is unlimited. Each FIFO queue is tagged with an index of integer k ($k^{n,P_n} \geq k \geq 0$) to mark the priority of the FIFO queue. We refer to the FIFO queue that is tagged with k as the k-th queue in category C_n. The lower the index, the higher the priority. The scheduler will always serve packets from a FIFO queue, which has the highest priority in all nonempty FIFO queues. No packet from the lower priority FIFO queues could be served when there are packets in the higher priority queues.

At the end of every rotation interval Δ_n, the index, i.e., the priority, of all the FIFO queues in n-th RPQ will be rearranged. For every $k^{n,P_n} \geq k \geq 1$, the index k will be changed to k-1, and the index 0 will become k^{n,P_n}. The tagging of the queues are rotated and this is the reason why the rearrange interval is referred to as rotation interval. Note that, if there are still packets in 0-th-queue at the moment of rotation, the packets should be discarded, because the packets have encountered a deadline violation. Packets from the C_n^p category will enter the current $k^{n,p}$-th FIFO queue. Assume that the rotation is performed instantaneously and the rotations of different RPQs can be performed simultaneously.

An example is given in Fig. 1 to demonstrate how the scheduler works. There are two groups of bounded delay services, the first group has three categories with delay bounds $\Delta_1, 2\Delta_1$ and $3\Delta_1$, the other group contains two categories with delay bounds Δ_2 and $3\Delta_2$, where Δ_1 and Δ_2 are the rotation intervals of the first and the second RPQ, respectively. We also assume that Δ_2 is much larger than Δ_1, i.e., $\Delta_2 >> \Delta_1$. As shown in Fig.1, both the first and the second RPQ maintain four FIFO queues. For the first RPQ, packets will enter the queues tagged with index 1,2, and 3. For the second RPQ, packets will enter the queues tagged with index 1 and 3. Fig.1 (a) shows the initial state of the scheduler, i.e., at time 0^-. Assuming that packets start to arrive at time 0, Fig.1 (b) shows a scenario of the scheduler at the end of first rotation interval of the first RPQ, i.e. at time Δ_1^-. In Fig.1(c), we illustrate that, at time Δ_1, the tags of the FIFO queues in the first RPQ are rearranged, but the tags in the second RPQ are not changed. After several rotation operations of the first RPQ, the current time becomes Δ_2^-. Fig.1 (d) depicts a possible case at the end of the first rotation interval of the second RPQ, i.e., at time Δ_2^-. After the rotation, we can see that the tags in the second RPQ is rearranged while the tags in the first RPQ remain unchanged. Of course, tags in both RPQ may be rearranged simultaneously if Δ_2 is a rational multiples of Δ_1.

Note that the scheduler consists of N RPQs whose priorities are static. When N equals 1, a SRPQ scheduler contains only one RPQ and degenerates to a RPQ scheduler. On the other hand, when P_n becomes 1, for all n, the SRPQ method degenerates to the SP method.

III. Schedulability Conditions

To provide QoS guarantee for real-time services, we need to know the conditions for a scheduler, under which the delay bounds will not be violated. These conditions are referred to as schedulability conditions. The region of service characteristics that satisfies the schedulability conditions is called the schedulable region. Obviously, the size of the schedulable region indicates the level of the potential bandwidth utilization. The bandwidth utilization is in proportion to the size of the schedulable region when the service rate of the multiplexer is a constant.

The schedulability conditions are closely related to traffic characteristics. Hence we discuss traffic characteristics first. Since the deterministic model has been widely used in the literature related to real-time services [1,3,4,7], we also use deterministic model to characterize the worst-case traffic of a connection. Let $A_j[t, t+\tau]$ be the total traffic arrivals from service j to the scheduler in the time interval $[t, t+\tau]$, where the traffic from service j consists of packets with maximum transmission time $s_j^{\max}$ and minimum transmission time $s_j^{\min}$. If for all $t \geq 0$ and all $\tau \geq 0$, the following inequality holds:

$$A_j[t, t+\tau] \leq A_j^*(\tau),$$

where $A_j^*(t) = 0$ for all $t < 0$ and $A_j^*(t) \geq 0$ for all $t \geq 0$,

then A_j^* is called the traffic constraint function of connection j. The advantage of using deterministic models is that the deterministic models not only describe the worst-case traffic for a connection via a small set of parameters but also enable simple traffic policing and rate control mechanisms. A well-known deterministic model is (σ,ρ)-model [6]. The traffic constraint function for the (σ,ρ)-model is $A_j^*(t)=\rho_j t+\sigma_j$.

Assume a set of services can be described by $\{A_j^*,d_j\}_{j\in C}$, where A_j^* is the traffic constraint function for service j'. d_j is the delay bound of the service j. Based on the principle of the SRPQ method, the service set C can be divided into several subsets $C_n^p(1\le n\le N,1\le p\le P_n)$ with service delay bounds $d^{n,p}$, which equal $k^{n,p}\Delta_n$. Then the necessary and sufficient schedulability conditions for the SRPQ scheduler with rotation interval $\{\Delta_1,\Delta_2,\cdots,\Delta_N\}$ can be described as follows: for all n $(1\le n\le N)$ and $t\ge d^{n,1}$,

$$\mu t\ge \sum_{m=1}^{n-1}\sum_{q=1}^{P_m}\sum_{j\in C_m^q}A_j^*(t)+\sum_{j\in C_n^1}A_j^*(t-d^{n,1})+\cdot$$
$$+\sum_{q=2}^{P_n}\sum_{j\in C_n^q}A_j^*(t-d^{n,q}+\Delta_n)+\max_{j\in C_m^q,d_j>t+\Delta_m,m>n}s_j^{\max} \quad (1)$$

where μ is the service rate of the multiplexer, $\max_{j\in C_m^q,d_j>t+\Delta_m,m>n}s_j^{\max}\equiv 0$, for $(m<n)$ or $(m=n,\ t>d_j-\Delta m)$

The proof of the above conclusion is attuned to the proof of the schedulibility conditions of RPQ and SP in [6]. We first calculate the number of packets in the SRPQ scheduler, which should be served no later than a tagged packet at any time. Let a tagged packet be a packet from service $j\in C_n^p$ that arrives at the SRPQ scheduler at time t and is completely transmitted at time $t+\delta$. We define the deadline of a packet as the arrival time of the packet plus the delay bound of the packet. Let $W^{\le x}(y)$ denote the workload in the SRPQ scheduler at time y due to the packets with deadlines no later than x. Assume $t-\hat{\tau}(\hat{\tau}\ge 0)$ is the last time before t when the scheduler does not contain packets with a deadline earlier than or equal to the deadline of the tagged packet. It means that if a packet is served at $t-\hat{\tau}$, the packet must belong to a service set with priority higher than the set C_n^p. So, $\hat{\tau}$ is given by

$$\hat{\tau}=\min\{z\,|\,W^{\le t+dn,p}(t-z)=0,\ z\ge 0\}. \quad (2)$$

Let $W^{n,p,t}(t+\tau)$ $(0\le\tau\le\delta)$ represent the workload in the scheduler at time $t+\tau$ that must be served before the departure time of the tagged packet. Note that $W^{n,p,t}(t+\tau)$ includes the tagged packet. $W^{n,p,t}(t+\tau)$ is determined by,

- Packets from all services belonging to C_m^q $(m<n,\ 1\le q\le P_m)$ in the time interval $[t-\hat{\tau},t+\tau]$.
- Packets from service $j\in C_n^p$ in the time interval $[t-\hat{\tau},t]$.
- Packets from all services which belong to $C_n^q(q<p)$ in the time interval $[t-\hat{\tau},\min\{t+\tau,(t-\varepsilon)+(k^{n,p}-k^{n,q})\Delta_n\}]$, where $t-\varepsilon$ $(\Delta_n\ge\varepsilon\ge 0)$ is the last priority rearranging time before t in the n-th RPQ.
- Packets from all services belonging to $j\in C_n^q(q>p)$ in the time interval $[t-\hat{\tau},\min\{t+\tau,(t-\varepsilon)+(k^{n,p}-k^{n,q}+1)\Delta_n\}]$, where $t-\varepsilon$ $(\Delta_n\ge\varepsilon\ge 0)$ is defined as above.
- $R(t-\hat{\tau})$, the remaining transmission time of a possible packet from lower priority services which is in transmission at time $t-\hat{\tau}$.
- Packets that are served in the time interval $[t-\hat{\tau},t+\tau]$.

The number of packets from service j in the time interval $[t_1,t_2]$ is given by $A_j[t_1,t_2]$. We then have,

$$W^{n,p,t}(t+\tau)=\sum_{m=1}^{n-1}\sum_{q=1}^{P_m}\sum_{j\in C_m^q}A_j[t-\hat{\tau},t+\tau]+\sum_{j\in C_n^p}A_j[t-\hat{\tau},t+\tau]+$$
$$+\sum_{q=1}^{p-1}\sum_{j\in C_n^q}A_j[t-\hat{\tau},\min\{t+\tau,(t-\varepsilon)+(k^{n,p}-k^{n,q})\Delta_n\}]+$$
$$+\sum_{q=p+1}^{P_n}\sum_{j\in C_n^q}A_j[t-\hat{\tau},\min\{t+\tau,(t-\varepsilon)+(k^{n,p}-k^{n,q}+1)\Delta_n\}]+$$
$$+R(t-\hat{\tau})-\mu(\hat{\tau}+\tau) \quad (3)$$

A. Proof of Sufficiency

Since the tagged packet is arbitrarily selected, it suffices to show that conditions (1) guarantees that the tagged packet will depart before or at its deadline. By selecting $\hat{\tau}$ as in (2) and the definition of $R(t-\hat{\tau})$, we have

$$R(t-\hat{\tau})\le\max_{j\in C_m^q,d_j>t+\Delta_m,m>n}s_j^{\max}. \quad (4)$$

Based on the property of traffic constraint function and the fact that the subset C_n^1 has the highest priority in service set C_n, we have,

$$W^{n,p,t}(t+k^{n,p}\Delta_n)\le\sum_{m=1}^{n-1}\sum_{q=1}^{P_m}\sum_{j\in C_m^q}A_j^*(\hat{\tau}+k^{n,p}\Delta_n)+$$
$$+\sum_{j\in C_n^1}A_j^*(\hat{\tau}+(k^{n,p}-k^{n,1})\Delta_n)+$$
$$+\sum_{q=2}^{P_n}\sum_{j\in C_n^q}A_j^*(\hat{\tau}+(k^{n,p}-k^{n,q}+1)\Delta_n)-$$
$$-\mu(\hat{\tau}+k^{n,p}\Delta_n)+\max_{j\in C_m^q,d_j>t+\Delta_m,m>n}s_j^{\max} \quad (5)$$

With (1), we have $W^{n,p,t}(t+k^{n,p}\Delta_n)\le 0$. Hence, the tagged packet is guaranteed to be served before or at its deadline, i.e., time $t+k^{n,p}\Delta_n$.

B. Proof of Necessity

We prove necessity conditions by contradiction. Let's suppose that (1) does not hold. Then, we construct a feasible pattern of packet arrivals that results in a deadline violation. Assume that the SRPQ scheduler is empty before time 0^- and

a packet from service $j \in C_m (m > n)$ arrives at time 0^-. Also assume that starting at time 0 all services $j \in C_m (m \le n)$ transmit at their maximum rates as permitted by their traffic constraint functions A_j^*, with one exception, i.e., the last packet from service $j \in C_n^1$ before or at time $t\text{-}d^{n,1}$ is submitted at time $t\text{-}d^{n,1}$.

Noticing that the scheduler can be idle in the time interval $[0, t]$, and according to (3), we have,

$$W^{n,1,t-d^{n,1}}(t) \ge \sum_{m=1}^{n-1}\sum_{q=1}^{P_m}\sum_{j\in C_m^q} A_j^*(t) + \sum_{j\in C_n^1} A_j^*(t-d^{n,1}) + \sum_{q=2}^{P_n}\sum_{j\in C_n^q} A_j^*(t-d^{n,q}+\Delta_n) - \mu t + \max_{j\in C_m^q, d_j > t+\Delta_n, m>n} s_j^{\max} \quad (6)$$

Based on the assumption that (1) does not hold, we get $W^{n,1,t-d^{n,1}}(t) > 0$. Therefore, the tagged packet cannot be completely transmitted at time t, resulting in a deadline violation for the tagged packet.

Suppose that there is only one service in the service subset C_m^q (for all m, q) with delay bounds $d^{m,q}$ and the service can be characterized by (σ,ρ)-model with parameters $\sigma^{m,q}$ and $\rho^{m,q}$. Then, the necessary and sufficient schedulability conditions are the following,

$$\mu t \ge \sum_{m=1}^{n-1}\sum_{q=1}^{P_m}(\rho^{m,q}t+\sigma^{m,q}) + (\rho^{n,1}(t-d^{n,1})+\sigma^{n,1}) + \max_{j\in C_m^q,(m,q)>(n,1)} s_j^{\max}$$
$$\text{for } d^{n,1} \le t < d^{n,2} - \Delta_n,$$

$$\mu t \ge \sum_{m=1}^{n-1}\sum_{q=1}^{P_m}(\rho^{m,q}t+\sigma^{m,q}) + (\rho^{n,1}(t-d^{n,1})+\sigma^{n,1}) + \sum_{q=2}^{p}(\rho^{n,q}(t+\Delta_n-d^{n,q})+\sigma^{n,q}) + \max_{j\in C_m^q,(m,q)>(n,p)} s_j^{\max}$$
$$\text{for } d^{n,p} - \Delta_1 \le t < d^{n,p+1} - \Delta_n,\ 2 \le p < P_n$$

$$\mu t \ge \sum_{m=1}^{n-1}\sum_{q=1}^{P_m}(\rho^{m,q}t+\sigma^{m,q}) + (\rho^{n,1}(t-d^{n,1})+\sigma^{n,1}) + \sum_{q=2}^{P_n}(\rho^{n,q}(t+\Delta_n-d^{n,q})+\sigma^{n,q}) + \max_{j\in C_m, m>n} s_j^{\max}$$
$$\text{for } t \ge d^{n,Pn} - \Delta_n. \quad (7)$$

Where $(m,q) > (n,p)$ means that the priority of service $j \in C_m^q$ is higher than that of service $j \in C_n^p$. If the SRPQ scheduler is stable, i.e., $\sum_{j\in C}\rho_j \le \mu$, the necessary and sufficient schedulability conditions can be further simplified as: for all integers $n > 0$,

$$d^{n,1} \ge \Big(\sum_{m=1}^{n-1}\sum_{q=1}^{P_m}\sigma^{m,q} + \sigma^{n,1} + \max_{j\in C_m^q,(m,q)>(n,1)} s_j^{\max}\Big) \Big/ \Big(\mu - \sum_{m=1}^{n-1}\sum_{q=1}^{P_m}\rho^{m,q}\Big),$$

$$d^{n,p} \ge \Big(\sum_{m=1}^{n-1}\sum_{q=1}^{P_m}(\sigma^{m,q}-\rho^{m,q}\Delta_n) + \sum_{q=1}^{p-1}(\sigma^{n,q}-\rho^{n,q}d^{n,q}) + \sigma^{n,p} + (\mu-\rho^{n,1})\Delta_n + \max_{j\in C_m^q,(m,q)>(n,p)} s_j^{\max}\Big) \Big/ \Big(\mu - \sum_{m=1}^{n-1}\sum_{q=1}^{P_m}\rho^{m,q} - \sum_{q=1}^{p-1}\rho^{n,q}\Big)$$
$$\text{for } 2 \le p \le P_n \quad (8)$$

TABLE I

TRAFFIC CHARACTERISTICS FOR THE SERVICES

Service index	Delay bound d_j (ms)	Burst size σ_j (cells)	Maximum rate ρ_j (Mbps)
1	0.1	10	0~155
2	1	200	0~155
3	5	500	0~155
4	10	1000	0~155

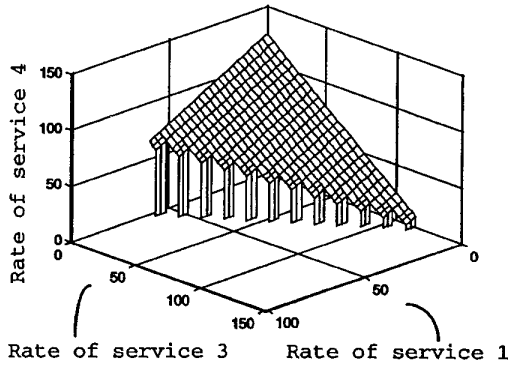

Fig.2 Schedulable region for EDF(unit: Mbps)

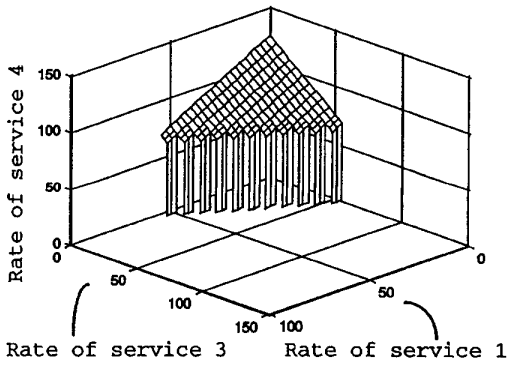

Fig.3 Schedulable region for SP(unit: Mbps)

IV. EMPIRICAL EVALUATION

Empirical calculations are performed to test the performance of the SRPQ scheduler. Assume that there are four kinds of services entering a multiplexer or a switch with an output rate of 155Mbps in ATM networks. (Note that the maximum and minimum transmission time of a packet is equal in ATM networks, i.e. $s_j^{\max} = s_j^{\min}$ =1cell=1*53*8bits) The services are characterized by (σ,ρ) model with listed in Table I. Using the schedulability conditions of the SRPQ method in this paper and the schedulability conditions of the EDF, SP and RPQ scheduling methods, we calculate and compare the schedulable regions of these methods. Let the rotation interval of the RPQ method be 0.1ms. For the SRPQ method,

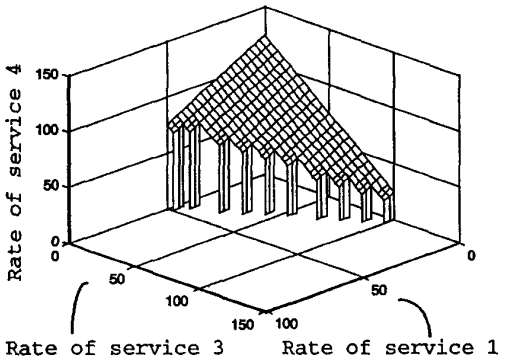

Fig.4 Schedulable region for RPQ(unit: Mbps)

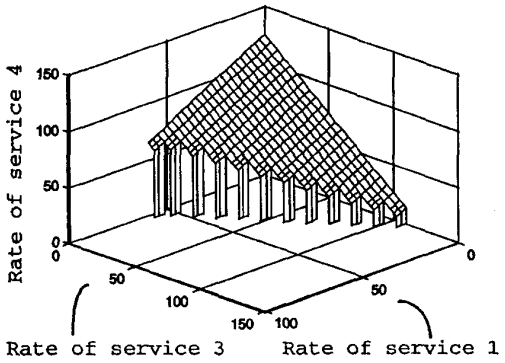

Fig.5 Schedulable region for SRPQ(unit: Mbps)

we classify all services into two categories, the first category represents the services with stringent delay bounds and includes service-1 and service-2, and the second category, which has loose delay requirements, includes service-3 and service-4. Packets from the first category enter RPQ-1 with rotation interval Δ_1=0.1ms. Packets from the second category enter RPQ-2 with rotation interval Δ_2=1ms. The results for the services with the given parameters when the rate of service-2 is fixed at 30Mbps are presented in Figs.2-5. The volumes below the surfaces of the graphs depict the schedulable regions of these methods for connection rate ρ_i, i.e., all values under the surface satisfy the schedulability conditions. Values above the surfaces violate the schedulability conditions and can not be assigned as the connection rates of the services at the multiplexer or switch. From the plots, we find that the schedulable region for the EDF method is the largest, and the one for the SP method is the smallest. The schedulable region for the RPQ method is almost identical to that for the EDF method. As for the SRPQ method, the schedulable region approximates that for the RPQ method but is much larger than that for the SP method. Note that the RPQ method must maintain 101 FIFO queues (10/0.1+1=101), whereas the SRPQ method needs to maintain only 22 FIFO queues (10/1+1/0.1+2=22). The results illustrate that the SRPQ method can achieve bandwidth utilization close to the RPQ method while greatly reducing the number of FIFO queues, simplifying the maintenance. Further numerical studies also achieve similar results.

V. Conclusion

In this paper, a new scheduling method, i.e., the SRPQ method, was developed. A SRPQ scheduler consists of several RPQ schedulers with predefined priorities. The exact schedulability conditions of the SRPQ scheduler were presented and proved, which is essential for call admission control in connection oriented networks. Empirical evaluation showed that the SRPQ method makes a better trade-off than other methods, such as the FCFS, SP, and RPQ methods, in satisfying important implementation requirements. The most appealing feature of SRPQ method is that it can reduce the number of FIFO queues in the multiplexer or switch, and thus reduce the complexity of the scheduler, while maintaining relatively high bandwidth utilization. Using the policing mechanism and schedulability conditions, the SRPQ method can provide guaranteed services with diverse delay bounds at low implementation cost.

References

[1] D. Ferrari, "Client Requirements for Real-Time Communication Services," IEEE Comm. Magazine, Vol. 28, No. 11, pp. 65-72, November 1990.

[2] C.M. Aras, J.F. Kurose, and D.S. Reeves, H. Schulzrinne, "Real-time Communication in Packet-Switched Networks," Proc. IEEE, Vol. 82, No.1, pp. 122-139, January. 1994.

[3] R.A. Cruz, "Calculus for Network Delay, part II: Network Analysis," IEEE Trans. on Comm., Vol. 37, No. 1, pp. 13-147, January 1991.

[4] D. Ferrari, and D.C. Verma, "A scheme for Real-Time Channel Establishment in Wide-Area Networks," IEEE JSAC, Vol. 8, No. 3, pp. 368-379, March 1990.

[5] A.K. Parekh, and P.G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: the Single-Node Case," IEEE/ACM Trans. on Networking, Vol. 1, No. 3, pp. 344-357, June 1993.

[6] J Liebeherr, D.E. Wrege, and D. Ferrri, "Exact Admission Control for Networks with a Bounded Delay Service," IEEE/ACM Trans. on Networking, Vol. 4, No. 6, pp. 885-901, December 1996.

[7] S.K. Kweon, and K.G. Shin, "Providing Deterministic Delay Guarantees in ATM Networks," IEEE/ACM Trans. on Networking, Vol. 6, No. 6, pp. 838-850, December 1998.