Delay Specific Investigations on QoS Scheduling Schemes for Real-Time Traffic in Packet Switched Networks

P.S.Prakash, S.Selvan

***Abstract*—**Packet switched data network like Internet, which has traditionally supported throughput sensitive applications such as e-mail and file transfer, is increasingly supporting delay-sensitive multimedia applications such as interactive video. These delay-sensitive applications would often rather sacrifice some throughput for better delay. Unfortunately, the current packet switched network does not offer choices, but instead provides monolithic best-effort service to all applications. This paper evaluates Class Based Queuing (CBQ), Coordinated Earliest Deadline First (CEDF), Weighted Switch Deficit Round Robin (WSDRR) and RED-Boston scheduling schemes that is sensitive to delay bound expectations for variety of real time applications and an enhancement of WSDRR is proposed.

***Keywords*—**QoS, Delay-sensitive, Queuing delay, Scheduling

I. INTRODUCTION

IN a packet switched data network, a packet generated by a source node is sent through the network, which consists of set of switches to some destination node. Each node in the network has incoming and outgoing links, and finite buffer space to store packets that could not yet be transmitted through an outgoing link. Regardless of how simple or sophisticated, each router must implement some queuing discipline that governs how packets are buffered while waiting to be transmitted. The scheduling algorithm allocates both bandwidth & buffer space.

The current packet switched data network like Internet, which has traditionally supported throughput sensitive applications such as e-mail and file transfer, is increasingly supporting interactive real-time

Due to simplicity of the First-In-First-Out (FIFO) queuing mechanism, drop-tail buffers are the most widely used queuing scheme to Internet routers today. When drop-tail buffers overflow, newly arriving packets are dropped regardless of the application type of the arriving packet. To accommodate bursty traffic, drop-tail routers on the Internet backbone are over provisioned with large FIFO buffers. When faced with persistent congestion, these drop-tail routers yield high delays for all flows passing through the bottlenecked router. The current network does not offer choices, but instead provides monolithic best-effort service to all applications. This paper evaluates various scheduling scheme that is sensitive to Quality of Service (QoS) expectations. Description of scheduling algorithms, Experimental analysis and Enhancements are presented in section II and III respectively.

Manuscript received April 20,2007
P.S.Prakash is with Computer Science and Engineering (PG) Department, Sri Ramakrishna Engineering College, Coimbatore, TamilNadu, India. (Phone:+91 422 2312021, 99945 25625; e-mail: prakashpsrajan@rediffmail.com)

Dr. S. Selvan is with PSG College of Technology, Coimbatore, India.

II. SCHEDULING ALGORITHMS

The following are some of the scheduling mechanisms, developed with keeping real-time traffic in mind.

- Class Based Queuing (CBQ) [1]-[3]
- Coordinated Earliest Deadline First (CEDF) [4]-[6]
- Weighted Switch Deficit Round Robin [7]
- Random Early Detection-Boston (RED-Boston) [10],[11]

A. Class Based Queuing

Class based queuing is a scheduling mechanism that aims to provide link sharing between agencies that are using the same physical link and to provide a framework to differentiate traffic that has different priorities. CBQ schedulers are used as mechanism to provide hop-by-hop guarantees for Real-Time traffic. The main blocks for CBQ are shown in Fig.1

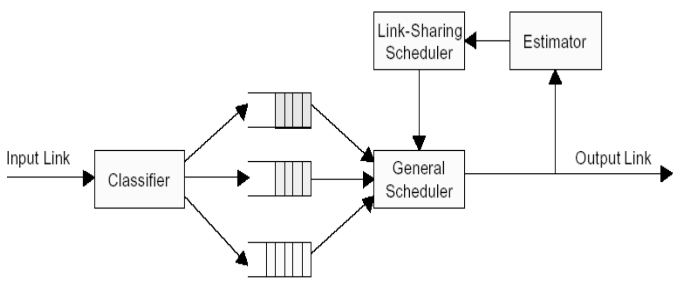


Fig. 1 Main blocks of CBQ

Classifier extracts flow information from packet, and to place packet into corresponding class. General scheduler is the scheduler mechanism that aims to share the bandwidth when all classes are backlogged. It guaranteed the right quantity of service to each leaf classes, distributing the bandwidth according to their allocations. Link sharing scheduler mechanism that aims to distribute the excess bandwidth according to link sharing structure. Estimator it measures the inter-packet time for each class, and estimates whether the class is under limit or over limit. In CBQ flows are differentiated based on flow id.

Hypothesis 1: Assuming that WFQ is used as a general scheduler and that leaf classes are never regulated to something more restrictive than their share, which means that *offtime* $= L_{max}/r_s p_i - L_{max}/r_s$.

Concepts of service and arrival curves of network calculus to formalize the behavior of CBQ are used. Let $R(t)$ be the quantity of bits that have entered the system up to time t, and $R^*(t)$ the quantity of bits that have left the system up to time t.

Definition 1 (Arrival curve, [16]): An arrival curve for flow R is a non strictly increasing function α such that:

$$s \le t, R(t) - R(s) \le \alpha(t-s) \tag{1}$$

That is equivalent to:

$$\forall t \ge 0, R(t) \le (a \otimes R)(t) \tag{2}$$

Where $\otimes$ is the min-plus convolution operator defined by:

$$\forall t \ge 0, (f \otimes g)(t) = \inf_{0 \le s \le t} \{f(t-s) + g(s)\} \tag{3}$$

Besides, a service curve β characterizes the behavior of network element regarding a flow R, independently of the traffic that can enter the system and of the flow itself.

Definition 2 (Service curve, [16])): β is a service curve for a flow going through a system S if and only if β is wide sense increasing, $\beta(0) = 0$, and $R^* \ge R \otimes \beta$.

Theorem 1*: Let f and g be two continuous wide-sense increasing functions with* $\forall t < 0$ $f(t) = g(t) = 0$. *Then* $\forall t$, $\exists t_0$ *such that*

$$(f \otimes g)(t) = f(t_0) + g(t - t_0) \tag{4}$$

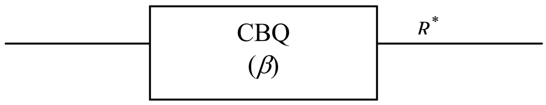
It is now model CBQ as the concatenation of two separate mechanisms: weighted fair queuing on one side, the priority queuing plus the regulation mechanism on the other side. Thus with Hypothesis 1, split the analysis of CBQ in two parts. In a first step, determine the service curve β_1 of system S_1 , and then the service curve β_2 of system S_2 .Using the combination theorem of network calculus[16], the resulting service curve β of system S is $\beta_1 \otimes \beta_2$, providing that S_1 and S_2 are used in cascade.

Lemma 1[16]**:** The service curve offered by system S_1 (i.e. a WFQ server) to a flow i is:

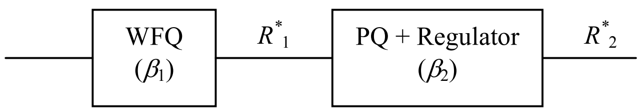
$$\beta_1(t) = r_s p_i \left[t - \frac{L_{max}}{r_s} \right]^+ \tag{5}$$

Lemma 2: *Under hypothesis 1, the service curve offered by system* S_2 *to a flow is:*

$$\beta_2(t) = \frac{1}{r_s} \cdot \frac{L_i \ p_i}{L_i + L_{\max} p_i} \cdot \left[t - \frac{L_{\max}}{r_s} \right]^+ \tag{6}$$



(a) System S



(b) System S_1 and S_2

Fig. 2 Decomposition of CBQ in two sub-systems S_1 and S_2

Compute $\beta = \beta_1 \otimes \beta_2$. Since $L_i r_s p_i / L_i + L_{\max} p_i \le r_s p_i$,

$$\beta(t) = \frac{L_i r_s p_i}{L_i + L_{\max} p_i} \cdot \left[t - \frac{2L_{\max}}{r_s} \right] \tag{7}$$

Theorem 2: under hypothesis 1, the service curve offered by a CBQ server using WFQ to a highest priority class is given by (7)

Delay bounds From the service curve given above in (7), we derive a delay bound for a highest-priority flow restricted by a leaky bucket and going through a CBQ server. Of course, we assume that classes of the highest-priority are never regulated to something more restrictive than their share, just as in hypothesis 1.

The delay bound D for a flow shaped by a leaky bucket (σ, $r_s p_i$) is obtained with network calculus:

$$D = \frac{\sigma}{r_s} \cdot \frac{L_i + L_{\max} p_i}{L_i p_i} + \frac{2nL_{\max}}{r_s} \tag{8}$$

Where L_i is the size of a packet of flow i. Using the concatenation principle, the bound can be derived for several nodes.

$$D_n = \frac{\sigma}{r_s} \text{ x } \frac{L_i + L_{\max} p_i}{L_i p_i} + \frac{2nL_{\max}}{r_s} \tag{9}$$

Where n is the number of CBQ nodes on the path of a highest- priority flow. If we assume that $L_i = L_{\max}$,a lower value of the bound, and it is less dependent on packet sizes:

$$D = \frac{\sigma}{r_s} \cdot \frac{1 + p_i}{pi} + \frac{2L_{\max}}{r_s} \tag{10}$$

B. Coordinated EDF (CEDF)

The CEDF (5) service discipline is developed with the goal of minimizing end-to-end delays. The approach is to use EDF together with randomization of packet injection time and coordination of servers. In CEDF K-hop delay does not have to K times 1-hop delay. The delay bound is represented as follows:

$$\frac{1}{\rho_i} + K_i \tag{11}$$

Where $\frac{1}{\rho_i}$ is delay at a node and K_i is number of nodes.

In this policy a deadline is assigned for every node through which a packet passes. By introducing randomness in the deadlines, deadlines can be sufficiently 'spread out' so that all the packets can meet all their deadlines. By introducing simple coordination among the deadlines, once a packet has passed through its first server, it can pass through all its subsequent servers quickly.

The basic idea of Coordinated-EDF [4],[5] is each packet *p*, assign deadlines $D_1, D_2 \ldots, D_K$ for every server, $m_1, m_2 \ldots, m_K$, through which *p* passes. The deadlines at a server *m* are defined using a parameter G_m, where G_m is essentially $(L_{max}/r^m)\log(.)$ In particular, D_1 is *rand* $+G_m$ time after *p*'s injection, where *rand* is a random number chosen from an appropriate range. Each subsequent deadline D_{k+1} is D_k+G_{mk}. CEDF gives priority to the packet with the earliest deadline if more than one packet is waiting for a server. Ties are broken arbitrarily. Let the t_{inj} is injection time at which session-*i* packet is injected.

Theorem 1: With high probability, the end-to-end delay guarantee for session i is

$$\frac{\sigma_i + 4L_i / \varepsilon}{\rho_i} + \alpha \sum_{K=1}^{K_i} \frac{L_{\max}}{r^{m_k}} \log_e \left(\frac{nMr^{m_k} \varepsilon}{L_{\min}} \right) \tag{12}$$

To prove Theorem 1,two statements are considered. First, with high probability the protocol is successful. (Lemmas 2 and 3).A protocol is *successful* if *every* packet meets all of its deadlines. The success of the protocol is equivalent to the successful placing of a *finite* number of tokens due to the periodicity of the token placement. Hence, a Chernoff-bound argument is in place to analyze the success probability. Second, τ is at most $t_{inj} + \sigma_i/\rho_i + 4L_i/(\varepsilon\rho_i)$ for each session-*i* packet, where t_{inj} is the injection time of that packet. (Lemma 4)

Consider a server *m* and a time interval *I*. Let *P* be the set of packets that have a deadline for server *m* in interval *I*. If the total size of the packets in *P* is *x*, then we say that *I services x* bits at server *m*.

Lemma2: *Consider any server m and any time interval* $I = [t - G_m, t]$, *where t is a potential deadline for some session at server m. With high probability, any such interval I services fewer than* $G_m r^m$ *bits at server m.*

Lemma 3: *If the assumption in Lemma* 2 *holds, then every packet meets all its deadlines*

Lemma 4: *For each session-i packet p injected at* t_{inj},

$$\tau \le t_{inj} + \frac{\sigma_i}{\rho_i} + \frac{4L_i}{\varepsilon\rho_i} \tag{13}$$

C. Weighted Switch Deficit Round Robin

Concept in DRR [8] [9] is that, during a given busy period for a given flow, the unused portion of the per-round bandwidth allocation rolls over to the next round. Consequently, a flow that is shortchanged in a particular round can be compensated in the next round. However, there has to be enough deficit accumulation prior to servicing a large packet by this mechanism, and latency problems arise if and when the busy periods for a given flow start with large packets, especially for some video streams having known variation (e.g. I-frame) that are large but last for short periods of time. Beyond protecting these delay sensitive flows through allocating high weighted bandwidth to their queue, we need to dynamically adjust the quantum of service. In WSDPR, this is addressed by allowing 'overdraft' i.e. borrowing against expected future deficits. With this modification, a flow can, in a particular round, exceed the available byte allowance up to a certain threshold (a fraction of the maximum packet size), thus yielding a negative deficit, which is to be restored in the subsequent rounds before another large packet can be serviced.

To calculate delay bound for WSDRR, delay bound analysis of Deficit Round Robin is taken and the analysis is extended with overdraft of WSDRR. The following discussions show basic definitions from DRR delay bound analysis [17],[19]and extended analysis of WSDRR.

Consider an output link of transmission rate *r*, access to which is controlled by the WSDRR scheduler. Let *n* be the total number of flows and let ρ_i be the reserved rate for flow *i*. Let ρ_{min} be the lowest of these reserved rates. In order that each flow receives service proportional to its guaranteed service rate, the WSDRR scheduler assigns a weight to each flow. The weight assigned to flow *i*, w_i is given by,

$$w_i = \frac{\rho_i}{\rho_{min}} \tag{14}$$

A flow is said to be *active* during a certain time interval, if it always has packets awaiting service during this interval. The WSDRR scheduler maintains a linked list of the active flows, called the *ActiveList*. At the start of an active period of a flow, the flow is added to the tail of the *ActiveList*. A *round* is defined as one round robin iteration during which the WSDRR scheduler serves all the flows that are present in the *ActiveList* at the outset of the round. Each active flow is assigned a *quantum* by the WSDRR scheduler [18]. The quantum allocated to a flow is defined as the service that the flow should receive during each round robin service opportunity. Let Q_i represent the quantum assigned to flow *i* and let Q_{min} be the quantum assigned to the flow with the lowest reserved rate. The quantum assigned to flow i Q_i is given by $w_i.Q_{min}$. Thus, the quanta assigned to the flows are in proportion of their reserved rates. The scheduler maintains a per-flow state, the *deficit count*, which records the difference between the amount of data actually sent thus far, and the amount that should have been sent. This deficit is added to the value of the quantum in the next round, as the amount of data the scheduler should try to schedule in the next round. Thus, a flow that received very little service in a certain round is given an opportunity to receive more service in the next round. During some service opportunity, some flows are allowed to transmit a packet even if its size exceeds its allocated quantum. For this 'Overdraft' parameter is used. It defines upper limit on amount exceed.

Definition 5: Define **T** as the set of all time instants at which the scheduler ends serving one flow and begins serving another. The set of all time instants at which a scheduler begins serving flow i is defined as $\mathbf{T}_i$. Note that the set **T** is the union of $\mathbf{T}_i$ for all active flows i.

Definition 6: The latency of a flow is defined as the minimum non-negative constant θ_i that satisfies the following for all possible busy periods of the flow,

Theorem 1: The WSDRR scheduler belongs to the class of LR servers, with an upper bound on the latency, θ_i for flow i, given by

$$\theta_i \le \frac{1}{r}\begin{bmatrix}(W - w_i)Q_{min} + (n-1)(m-1) + \\ (n_R - 1)TH - \left(\dfrac{W - w_i}{W}\right)TH\end{bmatrix} \quad (12)$$

Proof: Since the latency of an *LR* server can be estimated based on its behavior in the flow active period, let us prove the theorem by showing that,

$$\theta'_i \le \begin{bmatrix}(W - w_i)Q_{min} + (n-1)(m-1) \\ + (n_{R-1})TH - \left(\dfrac{W - w_i}{W}\right)TH\end{bmatrix} \quad (13)$$

Let flow i become active at time instant τ_i. In deriving an upper bound on the latency of WSDRR, consider a time interval (τ_i, t) during which flow i is continuously active. Then, obtain the lower bound on the total service received by flow i during this time interval. Refer to (12), for lower bound delay. In [19], in the context of deriving the latency bound of Elastic Round Robin, it is proved that if the upper bound of latency is met exactly during the active period (τ_i, t), then the following two conditions are satisfied:

1) $\tau_i \in \mathbf{T}$ and

2) $t \in \mathbf{T}_i$

It can be easily verified that these conditions are applicable in the analysis of the latency bound of all round robin schedulers including WSDRR. Let τ^k_i be the time instant marking the start of the k-th service opportunity of flow i. Note that τ^k_i belongs to the set $\mathbf{T}_i$. From the above, to determine a tight upper bound on the latency of the WSDRR scheduler, need to only consider time intervals (τ_i, τ^k_i) for all k. Fig. 3 illustrates the time interval under consideration for a given k. Note that the time instant τ_i may or may not coincide with the end of a round and the start of the subsequent round. Let k_0 be the round which is in progress at time instant τ_i or which ends exactly at time instant τ_i. Let the time instant t_h mark the end of round $(k_0 + h - 1)$ and the start of the subsequent round

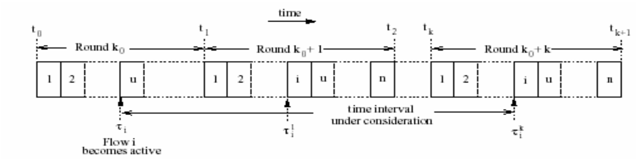


Fig. 3 An illustration of the time interval under consideration

Let R represent set of real-time flows for which overdraft is allowed and NR represent set of non real-time flows. Also let n_R represents number of real-time flows and n_{NR} represents the number of non real-time flows. Let $Sent^R_i(s)$ represent the total data transmitted from real-time flow i in round s of the WSDRR scheduler and $Sent^{NR}j(s)$ represent the total data transmitted from non real-time flow j in round s of the WSDRR scheduler. Also, let $DC^R_i(s)$ represent the deficit count of real-time flow i following its service in round s. Also, let $DC^{NR}_j(s)$ represent the deficit count of non real-time flow j following its service in round s. Let TH be the upper limit of 'overdraft' and owing to overdraft the deficit is become negative. For any flow i and j in any round s,

$$-TH \le DC_i{}^R(s) \le m-1 \quad (14)$$

$$0 \le DC_j{}^{NR}(s) \le m-1 \quad (15)$$

$$Sent_i{}^R(s) = w_i Q_{min} + DC_i{}^R(s-1) - DC_i{}^R(s) \quad (16)$$

$$Sent_i{}^{NR}(s) = w_i Q_{min} + DC_i{}^{NR}(s-1) - DC_i{}^{NR}(s) \quad (17)$$

As illustrated in Fig. 3, assume that the time instant when flow i becomes active coincides with the time instant when some flow u is about to start its service opportunity during the k_0-th round. Let G_a denote set of flows, which receive service during the time interval (τ_i, t_1), i.e., *after* flow i becomes active. Similarly, let G_b denote the set of flows, which are served by the WSDRR scheduler during the time interval (t_0, τ_i), i.e., *before* flow i becomes active. Note that flow i is not included in either of these two sets since flow i will receive its first service opportunity only in the $(k_0 + 1)^{th}$ round. If the time instant τ_i coincides with the time instant t_1, which marks the end of the k_0-th round and start of the $(k_0+1)^{th}$ round, then the set G_a will be empty and all the $(n - 1)$ flows will be included in the set G_b. Note that in this case, flow i will be the last to receive service in the $(k_0 + 1)^{th}$ round and all subsequent rounds during the time interval under consideration.

The first step towards analyzing the latency bound involves obtaining an upper bound on the size of the time interval (τ_i, τ^k_i). This time interval can be split into the following three sub-intervals:

1.(τ_i, t_1): This sub-interval includes the part of the k_0-th round during which all the flows belonging to the set G_a will be served by the WSDRR scheduler. G_a includes set of real-time flows and non real-time flows before real-time flow *i*. Summing (16) and (17) over all these flows.

$$t_1 - \tau_1 = \frac{1}{r}\left[\sum_{\substack{j\in Ga\\ j\in R}}\left\{w_j Q_{min} + DC_j^R(k_0-1) - DC_j^R(k_0)\right\}\right] + \frac{1}{r}\left[\sum_{\substack{j\in Ga\\ j\in NR}}\left\{w_j Q_{min} + DC_j^{NR}(k_0-1) - DC_j^{NR}(k_0)\right\}\right] \quad (18)$$

2. (t_1, t_k): This sub-interval includes k-1 rounds of the WSDRR scheduler starting at round $(k_0 + 1)$

$$t_k - t_1 = \frac{W}{r}(k-1)Q_{min} + \frac{1}{r}\left[\sum_{j\in R}\left\{DC_j^R(k_0) - DC_j^R(k_0+k-1)\right\}\right] + \frac{1}{r}\left[\sum_{j\in R}\left\{DC_j^{NR}(k_0) - DC_j^{NR}(k_0+k-1)\right\}\right] \quad (19)$$

(t_k, τ^k_i): This sub-interval includes the part of the $(k_0 + k)^{th}$ round during which all the flows belonging to the set G_b will be served by the WSDRR scheduler. G_b includes set of real-time flows and non real-time flows before the real-time flow *i*. Summing (16) and (17) over all these flows,

$$\tau_i^k - t_k = \frac{1}{r}\left[\sum_{\substack{j\in G_b\\ j\in R}}\left\{w_{jQ_{min}} + DC_j^R(k_0+k-1) - DC_j^R(k_0+k)\right\}\right] + \frac{1}{r}\left[\sum_{\substack{j\in G\\ j\in NR}}\left\{w_j Q_{min} + DC_j^{NR}(k_0+k-1) - DC_j^{NR}(k_0+k)\right\}\right] \quad (20)$$

Combining (18), (19), and (20) and since W is the sum of the weights of all the n flows. And also since flow i becomes active during round k_0, its deficit count at the end of the k_0-th round, $DC_i(k_0)$ is equal to zero. Using this fact, bounds on the deficit count from (14) and (15) substituted in (20),

$$\tau_i^k - \tau_i \le \frac{W}{r}(k-1)Q_{min} + \left(\frac{W-w_i}{r}\right)Q_{min} + \frac{(n_{R-1})(m-1)}{r}\cdot TH + \frac{(n_{NR})(m-1)}{r} - \frac{1}{r}(DC_i^R(k_0) - DC_i^R(k_0+k-1)) \quad (21)$$

Solving for (k-1),

$$(k-1) \ge \frac{r}{W\cdot Q_{min}}\left(\tau_i^k - \tau_i\right) - \left(\frac{W-w_i}{w}\right) - \frac{1}{W\cdot Q_{min}}(n_R-1)(m-1) - \frac{1}{W\cdot Q_{min}}(n_R-1)\cdot TH - \frac{1}{W\cdot Qmin}(n_{NR}-1)(m-1) + \frac{1}{W\cdot Q_{min}}DC_i^R(k_0+k-1) \quad (22)$$

Note that during the time interval under consideration, (τ_i,τ^k_i), flow i receives services in (k-1) rounds starting at round $(k_0 + 1)$.Hence, using (22) over these (k-1 rounds of service for now I, and since deficit count of a newly active flow is 0,then substituting (22) for (k-1) in (23), we get

$$Sent_i^R(\tau_i, \tau_i^k) = w_i(k-1)Q_{min} - DC_i^R(k_0+k-1) \quad (23)$$

$$Sent_i^R\left(\tau_i^k - \tau\right) \ge \frac{w_i r}{w}\left(\tau_i^k - \tau\right) - \left(\frac{w_i}{W}\right)(W-w_i)Q_{min} - \left(\frac{w_i}{W}\right)(n_R-1)(m-1) - \left(\frac{w_i}{W}\right)(n_R-1)TH - \left(\frac{w_i}{W}\right)(n_{NR})(m-1) + \left(\frac{w_i}{W}\right)DC_i^R(k_0+k-1) - DC_i^R(k_0+k-1) \quad (24)$$

Now, since the reserved rates are proportional to the weights assigned to the flows as given by (14), and since the sum of the reserved rates is no more than the link rate r, we have

$$\rho_i \le \frac{w_i}{W}\cdot r \quad (25)$$

$$Sent_i^R(\tau_i^k - \tau) \ge \rho_i(\tau_i^k)(W-w_i)Q_{min} - \left(\frac{\rho_i}{r}\right)(n_R-1)(m-1) - \left(\frac{\rho_i}{r}\right)(n_R-1)TH - \left(\frac{\rho_i}{r}\right)(n_{NR})(m-1) - \left(\frac{\rho_i}{r}\right)\left(\frac{W-w_i}{W}\right)DC_i^R(k_0+k-1) \quad (26)$$

Further simplifying and noting that the latency bound reaches the lower bound when $DC^R_i(k_0 + k - 1)$ equals TH,

$$\mathrm{Sent}_i^R(\tau_i^k-\tau)\geq\rho_i(\tau_i^k-\tau)-\left(\frac{\rho_i}{r}\right)(W-w_i)Q_{min}-\left(\frac{\rho_i}{r}\right)(n_R-1)(m-1)-\left(\frac{\rho_i}{r}\right)(n_R-1)TH-\left(\frac{\rho_i}{r}\right)(n_{NR})(m-1)+\left(\frac{\rho_i}{r}\right)\left(\frac{W-w_i}{W}\right) \quad (27)$$

$$\mathrm{Sent}_i^R(\tau_i^k-\tau)\geq\max\left\{0,\rho_i\left((\tau_i^k-\tau)-\frac{1}{r}\left[\begin{array}{l}(W-w_i)Q_{min}+(n-1)(m-1)+\\(n_{R-1})TH-\left(\frac{W-w_i}{W}\right)TH\end{array}\right]\right)\right\} \quad (28)$$

$$\mathrm{Sent}_i^R(\tau_i^k-\tau)\geq\max\left\{0,\rho_i\left((\tau_i^k-\tau)-\frac{1}{r}\left[\begin{array}{l}(W-w_i)Q_{min}+(n_R-1)(m-1)+\\(n_{R-1})TH+(n_{NR})(m-1)-\left(\frac{W-w_i}{W}\right)TH\end{array}\right]\right)\right\} \quad (29)$$

As discussed earlier, flow *i* will experience its worst latency during an interval (τ_i,τ^k_i) for some *k*. Therefore, from Equation (29), the statement of the theorem is proved.

D. RED Boston

In RED-Boston end-hosts indicate their sensitivity to queuing delay as source hints in the IP packet header that refer to as delay hints. RED-Boston is an Active Queue Management technique (AQM) that extends the Adaptive RED [12], [13]. The delay hint is not an absolute bound on queuing delay, but is rather a suggestion to Internet routers as to relative importance of delay versus throughput. At a congested router, queue manager uses the delay hints of all packets to calculate an average delay hint and a target average queue size chosen to best fit the aggregate traffic. RED-Boston inserts packets in the outbound queue based on their delay hint relative to the average delay hint. Packets with a delay hint lower than the average delay hint are inserted towards the front of the queue, while being dropped with a higher than average drop probability. Conversely, packets with a delay hint higher than the average delay hint are inserted towards the back of the queue, while being dropped with a lower than average drop probability. Applications such as multimedia streaming that seek to minimize end-to-end delay could choose to send a low delay hint.

III. EXPERIMENTAL ANALYSIS

In WSDRR [7], the queues of flows are served in round robin manner. When queue of real-time flow placed at the last, real-time flows needs to wait until one packet each of non real-time flow queue. To solve this problem we add the priority to the queues hence the queues are serviced based on priority. The priority is based on the delay hint of the flow. Delay hint gives the relative importance of the flow when compare to other flows and is named as Weighted Switched Deficit Priority Round Robin (WSDPRR).

This section discusses the experimental methods and NS-2 [14] simulation details associated with our simulation study of performance of scheduling schemes for real-time traffic in RED-Boston, CBQ, CEDF WSDRR, WSDPRR versus Drop Tail.

NS-2 simulator provides the ability to simulate drop-tail, RED routers and three simulations each with different network conditions are performed.

A. Regular topology

The generic topology used is shown in Fig. 4.

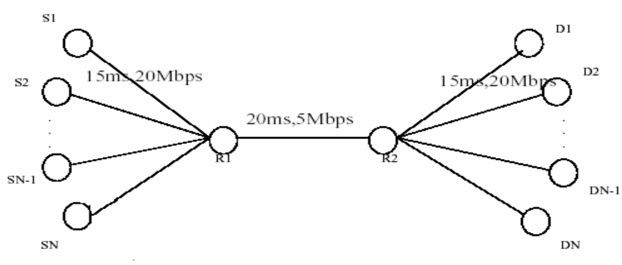


Fig. 4 Regular Topology

In the above topology, S1-SN are traffic sources and D1 – DN are destinations. The S-D pairs were varied to provide traffic sources that included different mixes of and TCP flows where UDP flows are meant to send real-time traffic. All links connecting sources to router R1 and all links connecting destinations to router R2 have 20Mbps bandwidth and 15ms delay. The bandwidth and delay of the bottleneck link going from R1 to R2 are set to 5 Mbps and 20 ms respectively. Queue Size is set 120 packets and minimum threshold is set to 20 packets and maximum threshold is set to 80 packets for RED-Boston. We focus on changing the percentages of delay-sensitive and throughput sensitive flows in the incoming traffic mix.

Using this scenario, five simulations were performed each with 20 flows. Table I provides details on traffic mixes for five simulations.

TABLE I
TRAFFIC MIX OF SIMULATION

Traffic Mix	Number of Real-Time flow	Number of Non Real-Time flow
1	01	19
2	05	15
3	10	10
4	15	05
5	19	01

Drop Tail scheduling gives more Queuing delay when compared to other algorithms since Drop Tail scheduling do not discriminate between real-time and non real-time traffic as shown in fig 5. All algorithms are compared against Drop Tail in this simulation.

When comparing WSDRR with other three mechanisms it gives slightly higher delay for Traffic Mix 5 (refer Table I) in which 19 real-time flows competing for one link. This is owing to the fact that in WSDRR scheme queues of flows are serviced in round robin fashion. But while servicing the real-time flows it allows P and I frame to pass the router even when their size of packet is large when compared to available byte allowance and hence P and I frames are not delayed. Even during congestion WSDRR do not drop P and I frames; instead it drops B frames so that video quality is not sacrificed. In modified WSDRR, since the flows are serviced based on delay hint, real-time flows received lower delay compare to original WSDRR.

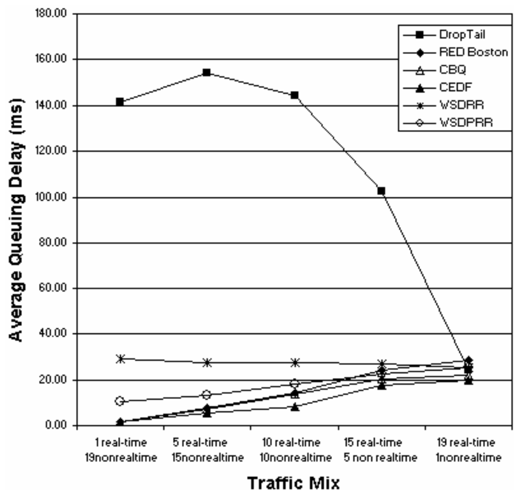


Fig. 5 Average Queuing Delay of real-time flows in Regular topology

In CBQ only two classes are maintained. Real time flows are placed in higher priority class and non real-time flows in lower priority flows so that real-time flows are serviced first. In the Traffic Mix 5 (refer Table I) as many as 19 real-time flows are placed in same class. Some real-time flows may have to wait for long time since other real-time flows are being serviced that arrived earlier and hence delay incurred is little higher. CEDF offers lower delay in Traffic mix 5, comparing to other three algorithms. It achieves this lower delay by randomization of packet injection time when the packet entered into the network and avoids the situation of congestion.

Since RED-Boston inserts per packet delay information based on their delay hint which in turn relative to average delay hint. In this experiment real-time flows having delay hint of 32 ms which is less than 100 ms for the FTP flows and hence real-time traffic flows will be serviced first. The queuing delay in RED-Boston is less than that of WSDRR. But during congestion RED-Boston drop the packets which are arrived at end of the queue without considering frame type and hence quality of real time flow may be reduced. As the percentage of real-time flow increases, average queuing delay of real-time flows increase slightly.

B. *Open Irregular Topology*

An irregular topology is constructed [20] which contains 30 nodes in which flows needs to pass through more than two router nodes when compared to previous topology. Three simulations with 10 flows each as in Table II were conducted and presented in fig. 6. In the Traffic Mix 1 with reference to Table II, out of 10 flows one real-time flow is competing for 9 non real-time flows in various nodes on the path. The non real-time flow packets arrived continuously with packet size of 1K bytes. Hence the delay is slightly higher for Traffic Mix1 compared to other Traffic Mixes.

TABLE II SIMULATIONS

Traffic Mix	Number of Real-Time Flows	Number of Non Real-time Flows
1	1	9
2	5	5
3	9	1

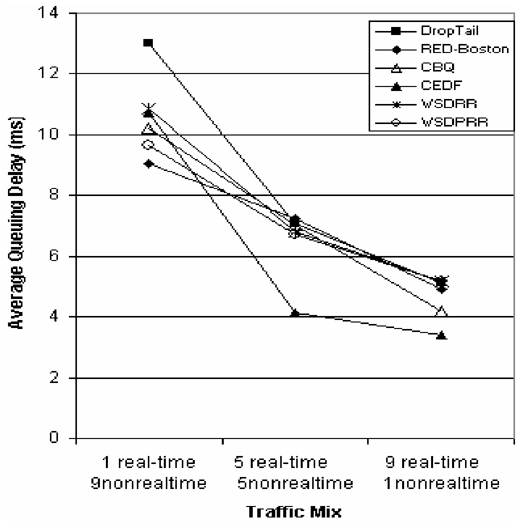


Fig. 6 Average Queuing Delays of real-time flows

In Traffic Mix 2 and 3, if an increase in number of real-time flows, the delay is reduced by small amount, for non real-time flows. Since real-time flow packets are bursty in nature, router node will not be congested all the time that makes delay reduced in these situations. When compared to other schemes CEDF yields low delay because of randomization of packet injection time and it avoids the congestion in nodes.

C. *Closed Irregular Topology*

A closed irregular topology which contains 49 nodes, constructed based on Waxman's method [20] and three simulations were carried out with 80 flows each as shown in Table III.

TABLE III SIMULATIONS

Traffic Mix	Number of Real-Time Flow	Number of Non Real-Time Flow
1	01	79
2	40	40
3	79	01

In this experiment CEDF revealed low average queuing delay compared to other schemes, because of randomization of packet injection time. But when moving from Traffic Mix 1 to 3, the delay is increased by small amount owing to the fact that injecting the packet at random time, some packets of real-time flows are delayed at first node on the path. WSDRR revealed high average queuing delay compared to RED-Boston, CBQ and CEDF as shown in Fig. 7 because of servicing the queue in Round-Robin fashion. But when servicing the queue of real-time flows it yields the property of 'overdraft' and hence it offers low delay compared to Drop Tail.

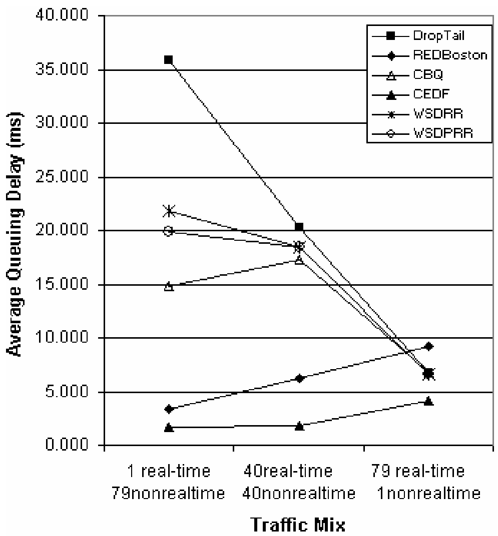


Fig. 7 Average Queuing Delay in closed irregular Topology

Modified WSDRR yields the lower queuing delay compared to original WSDRR since queue of flows are serviced based on delay hint of flow. CBQ reveals higher delay because in all real-time flow, packets are kept in same queue and packets belong to some real-time flows needs to wait for other real-time flows that arrived earlier. Average queuing delay, measured in milliseconds incurred by Drop Tail, RED Boston, CBQ, CEDF, WSDRR and WSDPRR for regular, open and closed topologies are presented in Table IV, V, and VI respectively. Real time flows are represented by 'R' and Non real time flows are represented by 'NR'.

TABLE IV
REGULAR TOPOLOGY

Scheme	1R & 19 NR	15 R & 05 NR	10 R & 10 NR	15 R & 05 NR
Drop-Tail	141.8	154.2	144.6	102.8
RED-Boston	1.79	7.74	14.49	24.42
CBQ	1.92	7.10	13.50	20.40
CEDF	1.71	5.34	8.27	17.42
WSDRR	29.03	33.52	28.25	26.69
WSDPRR	10.61	13.31	18.00	22.29

TABLE V
OPEN IRREGULAR TOPOLOGY

Scheme	19R & 1 NR	1 R & 9 NR	5 R & 5 NR	9 R & 1 NR
Drop-Tail	24.41	13.00	7.06	5.12
RED-Boston	28.33	9.04	7.26	4.93
CBQ	21.72	10.19	7.03	4.18
CEDF	19.59	10.71	4.12	3.42
WSDRR	25.50	10.86	6.80	5.18
WSDPRR	25.49	9.65	6.71	5.16

TABLE VI
CLOSED IRREGULAR TOPOLOGY

Scheme	01R & 79 NR	40 R & 40 NR	79 R & 01 NR
Drop-Tail	35.91	20.25	6.70
RED-Boston	3.42	6.17	9.21
CBQ	14.86	17.30	6.76
CEDF	1.67	1.87	4.11
WSDRR	21.83	18.45	6.67
WSDPRR	19.82	18.45	6.65

In this paper, a set of simulations performed to illustrate that scheduling algorithms such as RED-Boston, CBQ, CEDF, and WSDRR are contributing to minimize the queuing delays of real-time traffic at router. We conclude that by simulation, CEDF yields lower delay due to randomization of packet injection time. Also proposed a priority based scheduling scheme called WSDPRR, in which high priority is assigned to real-time queues and is found that delay is drastically reduced comparing to WSDRR. While simulating the WSDPRR maximum of 63% reduction in average queuing delay is achieved upon WSDRR.

ACKNOWLEDGMENT

The Authors wish to thank the reviewers whose valuable comments on this paper helped to improve the quality of the paper significantly.

REFERENCES

[1] Fulvio Risso, "Quality of Service on Packet Switched Networks", Politechnico Di Torino, Italy,Ph.D Thesis Jan. 2000.
[2] S. Floyd and V. Jacobson, "Link-sharing and Resource Management Models for Packet Networks," *IEEE/ACM Transactions on Networking*, vol. 3, no. 4, pp. 365–386, Aug. 1995.

[3] F. Risso and P. Gevros,"Operational and performance issues of a cbq router," *ACM Computer Communication Review*, Oct. 1999.

[4] M. Andrews, L. Zhang, "Minimizing end-to-end delay in high-speed networks with a simple coordinated schedule", in Proc. IEEE INFOCOM, vol. 1, New York, Mar. 1999, pp. 380-388

[5] Chengzhi LiEdward, W. Knightly, "Coordinated multihop scheduling: a framework for end-to- end services", IEEE/ACM Transactions on Networking, vol. 10, no. 6, pp. 776 - 789, Dec. 2002.

[6] Chengzhi LiEdward,W. Knightly, "Schdeulability Criterion and Performance Analysis of Coordinated Schedulers" IEEE/ACM Transactions on Networking, Vol.13, No.2, Apr.2005.

[7] Min Chen and Gang Wei, "Scheduling Algorithm for Real-time VBR Video Streams Using Weighted Switch Deficit Round Robin"**,** IEEE Computer Society, 2003

[8] Jung-Shian Li, "An Evaluation of Deficit Round Robin Fair Queuing Applied in Router Congestion Control," *Journal of Information Science and Engineering*, vol. 18/2, pp. 333-339, March 2002

[9] M. Shreedha, G. Varghese, "Efficient fair queuing using deficit round-robin," *IEEE/ACM Transactions on Networking*, vol.4, Issue 3, pp. 375 –385, June 1996

[10] Vishal Phirke, Mark Claypool, Robert Kinichi, " Traffic Sensitive Active Queue Management for Improved Multimedia Streaming", In Proceedings of International Workshop on Quality of Service in Multiservice IP Networks, Italy, February 2003, pp. 551-566

[11] Vishal Phirke, Mark Claypool, Robert Kinichi, " Traffic Sensitive Active Queue Management for Improved Multimedia Streaming", Technical Report WPI-CS-TR-02-10, Worcester Polytechnic Institute, April 2002

[12] Sally Floyd, Ramakrishna Gummadi, and Scott Shenker, "Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management". Submitted for Publication.

[13] Sally Floyd, V.Jacobso, "Random Early Detection Gateways for Congestion Avoidance", *IEEE/ACM Transactions on Networking*, August 1993

[14] University of California Berkeley, The Network Simulator - ns-2. Available http:// www.isi.edu/nsnam/ns

[15] Anne Millet, Zoubir Mammeri, "Delay bound Guarantees with WFQ-based CBQ discipline", The Twelth IEEE International workshop on Quality of Service (IWQoS 2004), June 2004.

[16] J.-Y. Le Boudec and P. Thiran,*Network Calculus, A Theory of Deterministic Queuing Systems for the Internet*, online version LNCS 2050. Springer Verlag, 2002.

[17] Salil S. Kanhere and Harish Sethu, "On the Latency Bound of Deficit Round Robin", in Proceedings of the International Conference on Computer Communications and Networks Miami, Florida, USA, October 14–16, 2002

[18] D. Stiliadis and A. Verma, "Latency-rate servers: A general model for analysis of traffic scheduling algorithms," *IEEE Transactions on Networking*, vol. 6, no. 3, pp. 611–624, October 1996.

[19] S. Kanhere and H. Sethu, "Low-latency guaranteed-rate scheduling using elastic round robin," *Computer Communications*, vol. 25, no. 14, pp. 1315–1322, September 2002

[20] Bernard M.Waxman "Routing of Multipoint Connections", IEEE Journal on selected areas in communications, vol 6, No 9, December 1988.

Dr. S. Selvan is Professor and Head,Department of Information Technology at PSG College of Technology, Coimbatore, India. He has 27 years of teaching experience. He has published more than 60 papers in international and national journals and conference proceedings. His areas of research include, computer networking digital signal processing.

P.S.Prakash received B.E (EEE) and M.E (Electrical Machines) degrees from P.S.G College of Technology, Coimbatore, India and also received M.E degree in Computer Science and Engineering from Bharathiar University, India. His research interests include QoS Scheduling, Routing, Network Security and Management. At present, serving as Asst.Prof at Sri Ramakrishna Engg. College, Coimbatore, India.