

On the Latency Bound of Pre-Order Deficit Round Robin*

Salil S. Kanhere and Harish Sethu

Department of ECE

Drexel University

3141 Chestnut Street, Philadelphia, PA 19104-2875.

E-mail: {salil, sethu}@ece.drexel.edu

Abstract

In the emerging high-speed packet-switched networks, packet scheduling algorithms used in the switches and routers will play a critical role in satisfying the Quality of Service (QoS) requirements of various applications. The latency bound of a scheduling discipline is an important QoS parameter, especially for real-time playback applications. Frame-based schedulers such as Deficit Round Robin (DRR), though extremely efficient with an $O(1)$ dequeuing complexity, lead to high latencies due to bursty transmissions of each flow's traffic. In a recent work by Tsao and Lin [14], the authors propose Pre-order Deficit Round Robin, a novel scheme that overcomes this limitation of DRR while still achieving a low work complexity. In Pre-order DRR, a priority queue module is appended to the original DRR scheduler which re-orders the packet transmission sequence in DRR to distribute the output more evenly among flows and thus reduce burstiness and improve the latency. In this paper, we employ a novel approach to analytically derive the latency bound of Pre-order DRR and show that our bound is a tight one. Our latency bound is significantly lower than the bound derived by Tsao and Lin, demonstrating that Pre-order DRR has even better performance characteristics than previously argued by its own authors.

1. Introduction

Future high-speed packet-switched networks are expected to support a variety of services beyond the best-effort service available in the Internet today. A number of new applications such as distance learning and multimedia tele-conferencing rely on the ability of the network to guarantee such services. For example, such applications would expect the network to ensure that each flow of traffic receives its fair share of the

bandwidth and is able to provide performance guarantees such as an upper bound on the end-to-end delay. This requires a Quality-of-Service (QoS) mechanism to efficiently apportion, allocate and manage limited resources among competing users. An important component of such a mechanism is the traffic scheduling algorithm typically used at the output links of switches and routers.

The function of a packet scheduler at an output link is to select the next packet for transmission from among the packets awaiting transmission through the output link. Some of the most important and desirable properties of a scheduling discipline are fairness, efficiency and low latency. Schedulers such as Weighted Fair Queueing [2, 9], Worst-Case Fair Weighted Fair Queueing (WF²Q) [1] and Self-Clocked Fair Queueing (SCFQ) [4] achieve good fairness through maintaining a global variable known as the virtual time or the system potential function. Such schedulers, known as sorted-priority schedulers, then use this variable to compute the timestamp for each packet indicating the relative priority of the packet for transmission over the output link. While they achieve good fairness and low latencies, they are not very efficient due to the complexity of computing the system virtual time and the complexity of maintaining a sorted list of packets based on their timestamps.

In frame-based schedulers such as Deficit Round Robin (DRR) [10], on the other hand, no global virtual time is maintained and the scheduler simply visits all the non-empty queues in a round robin order. This reduces the per-packet work complexity of DRR to $O(1)$ with respect to the number of flows, making it attractive for implementation in routers, and especially so, in hardware switches. However, such schedulers do have their limitations in fairness and latency. In the following, we briefly provide an overview of DRR and its limitations.

Let r be the transmission rate of the output link, the access to which is controlled by a DRR scheduler. Assume that there are a total of n flows multiplexed on this link. Let ρ_i be the reserved rate for flow i and let ρ_{min} be the minimum reserved rate among all the n flows. Since all these n flows share the same output link, $\sum_{i=1}^n \rho_i \leq r$. In order that the flows re-

*This work was supported in part by NSF CAREER Award CCR-9984161 and U.S. Air Force Contract F30602-00-2-0501.

ceive service proportional to their reserved rates, each flow i is assigned a weight, w_i , given by,

$$w_i = \frac{\rho_i}{\rho_{min}} \quad (1)$$

Note that for any flow i , $w_i \geq 1$.

A flow is said to be *active* during a certain time interval, if it always has packets awaiting service in this interval. The DRR scheduler operates in terms of rounds wherein a *round* refers to one round robin iteration during which the DRR scheduler visits all the flows that are active at the instant that the round begins. The DRR scheduler allocates a *quantum*, Q_i to each active flow i , which is defined as the ideal service that flow i should receive in each round. The quanta assigned to the flows are in proportion to their reserved rates. Let Q_{min} represent the quantum assigned to the flow with the lowest reserved rate. Hence, Q_i is given by $w_i Q_{min}$. Let M denote the size of the largest packet that may potentially arrive during the execution of a scheduling algorithm. For the DRR scheduler to have a work complexity of $O(1)$, it is necessary that Q_{min} is greater than or equal to M . A DRR frame is defined as the sum of the quanta allocated to all the active flows in a DRR round. Note that during a certain service opportunity, a flow may not be able to transmit a packet because doing so would cause the flow to exceed its allocated quantum. In such a case, the scheduler records the remained of the quantum in the *deficit count* associated with the flow. This deficit is added to the quantum in the subsequent round. Hence a flow that does not receive its fair share of the bandwidth during a certain round is given an opportunity to receive proportionately more service in the next round. Let m represent the size of the largest packet that actually arrives during the execution of a scheduling algorithm. Note that $m \leq M$. It has been proved in [10] that,

$$0 \leq DC_i(s) \leq m - 1 \quad (2)$$

However in all frame-based schedulers including DRR, each flow is served for a continuous period of time in proportion to its weight resulting in a highly bursty packet stream at the output of the scheduler. Also due to the round robin order of service, a flow that is lagging in service as compared to the other flows has to wait for its turn in the next round to compensate for the service lag. Further, there is no way for a lagging flow to receive precedence over all the other flows.

In [14], Tsao and Lin have proposed a new scheduling discipline called Pre-order DRR which aims at overcoming the aforementioned drawbacks. In Pre-order DRR a limited number of priority queues, p are added to the DRR scheduler. These queues reorder the transmission sequence of the packets in each DRR round and hence allow each flow to utilize its quantum in pieces over the course of the round. It is shown in [14] that Pre-order DRR belongs to the general class of Latency-Rate (\mathcal{LR}) servers [11] and that an upper bound on its latency

is $\frac{(2+(1/p))F-Q_i}{r}$ where F denotes the size of a DRR frame and r represents the transmission rate of the output link. In this paper, we use a different, unique and novel approach to analytically re-derive the latency bound of Pre-order DRR and show that our bound is a tight one. Our approach is based on interpreting the Pre-order DRR bandwidth allocations as an instance of the Nested Deficit Round Robin (Nested-DRR) discipline discussed in [5]. The latency bound of Pre-order DRR derived in this paper is significantly lower than the bound derived by Tsao and Lin, demonstrating that Pre-order DRR has even better performance characteristics than previously argued by its own authors.

The rest of the paper is organized as follows. Section 2 presents a brief overview of the Pre-order DRR scheduling discipline. Section 3 discusses the interpretation of Pre-order DRR bandwidth allocations as an instance of allocations in Nested-DRR. In Section 4, we present our analysis of the latency bound of Pre-order DRR. Section 5 presents a detailed comparison of the latency bound derived in this paper and that derived by Tsao and Lin in [14]. Finally, Section 5 concludes the paper with a tabulated overview of the latency bounds of all other schedulers in comparison with that of Pre-order DRR.

2. Pre-order Deficit Round Robin

In this section, we present a brief overview of Pre-order DRR scheduling discipline, a detailed description of which can be found in [14]. The goal of the Pre-order DRR scheduler is to eliminate the drawbacks of the DRR scheduler while trying to preserve its good properties such as its low work complexity. The assignment of the weights and the quanta are identical to DRR. In fact, the Pre-order DRR scheduler also works in rounds. However, unlike the DRR scheduler which serves the active flows in a round robin fashion, the Pre-order DRR scheduler reorders the transmission sequence of the packets within each DRR round. We shall first introduce some terms and definitions which will prove useful in the description of the Pre-order DRR scheduler.

Let us assume that a total of y packets are transmitted from flow i in the s -th round of service. The packets are labeled as $1, 2, \dots, y$ indicating their position in the stream of packets that are scheduled from flow i in round s . Note that y represents the last packet that is served in round s from flow i . As in DRR, the deficit count serves as a measure of past unfairness. Let $DC_i^m(s)$ represent the deficit count of flow i following the transmission of the m -th packet of the s -th round.

Definition 1 Define the Quantum Availability, denoted by $QA_i^m(s)$, of flow i after the transmission of the m -th packet from flow i in round s as follows:

$$QA_i^m(s) = \frac{DC_i^m(s)}{Q_i} \quad (3)$$

The *Quantum Availability* of a flow keeps track of the unused quantum of the flow in the current round.

In Pre-order DRR, a priority queue module consisting of p queues and a *classifier* module are appended to the original DRR architecture. Let PQ_1, PQ_2, \dots, PQ_p represent the priority queues in the descending order of priority with PQ_1 as the highest priority queue and PQ_p denoting the lowest priority queue. Just as in DRR, the Pre-order DRR maintains a linked list of active flows called the *ActiveList*. However, the flows in the *ActiveList* are not served in a round robin manner as in DRR. This is a list of the active flows that have already received their fair share of service in the current round. These flows are, however, eligible for receiving service in the subsequent round. At the start of a round, the *Classifier* module classifies the packets that will be served in the current round from each flow present in the *ActiveList* according to its *Quantum Availability* into the p priority queues. In general, the priority queue, $z_i^m(s)$ into which the m -th packet served from flow i in the s -th round is added is calculated as follows,

$$z_i^m(s) = p - \lfloor QA_i^m(s) \times p \rfloor \quad (4)$$

Once all the packets that can be scheduled in the current round from flow i have been transferred from the flow buffers into the priority queues, if flow i is still active, it is added to the tail of the *ActiveList*.

When the scheduler is ready to transmit, it begins serving the packet at the head of the highest non-empty priority queue. Note that, if a packet is added to a priority queue that has a higher priority than the queue from which the scheduler is currently serving a packet, then following the current transmission, the scheduler will first serve the packet added into the higher priority queue. The round in progress ends when all the priority queues are empty. It has been proved in [14] that Pre-order DRR has a low worst-case work complexity of $O(\log p)$ resulting in an efficient hardware implementation.

3. The Nested-DRR Interpretation

The primary goal of the Pre-order DRR scheduler is to break the quantum allocated to a flow in a DRR round into several pieces so that it can be utilized in pieces over the course of the round. The Nested-DRR scheduler proposed in [5] tries to eliminate the drawbacks of the DRR scheduler by creating a set of multiple rounds inside each DRR round and executes a modified version of the DRR algorithm within each of these inner rounds. The Nested-DRR scheduler tries to serve Q_{min} worth of data from each flow during each inner round. During an outer round, a flow is considered to be eligible for service in as many inner rounds as are required by the scheduler to exhaust its quantum. This results in a significantly lower latency bound, while preserving the $O(1)$ work complexity and the fairness characteristics of DRR. We can hypothetically

interpret the operation of the Pre-order DRR scheduler as a *nested* version of DRR similar to Nested-DRR. This interpretation is useful in analyzing the latency bound of the Pre-order DRR scheduler. Each round in DRR can be referred to as an *outer round*. The time period during which the Pre-order DRR scheduler serves the flows present in the priority queue PQ_u during the s -th outer round is referred to as *inner round* (s, u) . Thus, each outer round can be split into as many inner rounds as the number of priority queues, p . Since the Pre-order DRR scheduler visits the priority queues in a descending order of priority starting at priority queue PQ_1 and ending with queue PQ_p , the first and last inner rounds in outer round s are $(s, 1)$ and (s, p) respectively.

The quantum assigned to each flow is divided equally among the p priority queues. Thus, the quantum allocated to flow i in each of its inner rounds is equal to $\frac{Q_i}{p}$. Let $Served_i(s, u)$ represent the total data scheduled from flow i in inner round (s, u) . Also let $DC_i(s, u)$ denote the deficit round of flow i at the end of the (s, u) -th inner round. Note that the deficit count of a flow at the end of the last inner round of an outer round is the same as its deficit count at the end of the corresponding round in DRR. Also, this deficit count is carried over to the first inner round of the subsequent outer round. Hence,

$$DC_i(s, p) = DC_i(s) = DC_i(s + 1, 0)$$

Note that $DC_i(s + 1, 0)$ is used to represent the deficit count of flow i at the start of the inner round $(s + 1, 1)$. As in DRR, the deficit count is calculated as follows,

$$DC_i(s, u) = \frac{Q_i}{p} + DC_i(s, u - 1) - S_i(s, u) \quad (5)$$

It can be easily proved that Equation (2) which represents the bounds on the deficit count, $DC_i(s)$, also holds true for $DC_i(s, u)$. Hence for any flow i and inner round (s, u) ,

$$0 \leq DC_i(s, u) \leq m - 1 \quad (6)$$

In DRR, since the quantum of each flow is greater than or equal to the size of the largest packet that may potentially arrive during its execution, the scheduler is guaranteed to serve at least one packet from each of the active flows in each round. However, in Pre-order DRR, it may be possible that the sum of $\frac{Q_i}{p}$ and $DC_i(s, u - 1)$ is less than the size of the packet at the head of flow i . In this case, flow i will not receive any service in inner round (s, u) . Thus, a flow need not necessarily receive service in each inner round. If the Pre-order DRR scheduler was serving flows in an exact round robin manner as in Nested-DRR then, in the worst-case, it may be possible that none of the active flows will be able to transmit a packet in an inner round resulting in a work complexity of $O(n)$ or greater, where n represents the total number of active flows. The *Classifier* module in the Pre-order DRR scheduler avoids this large

work complexity by classifying the packets into the p priority queues at the start of each outer round. This classification determines which inner rounds each flow will be served in and the scheduler does not need to query all the flows in a round robin order.

Note that the deficit count of a flow is updated at the end of each inner round using Equation (5) irrespective of whether it receives service in that inner round. From Equation (5), the service received by flow i in inner round (s, u) is,

$$Served_i(s, u) = \frac{Q_i}{p} + DC_i(s, u - 1) - DC_i(s, u) \quad (7)$$

Definition 2 Let $Sent_i(s, u)$ represent the total service received by flow i since the start of the s -th outer round until the time instant when the scheduler finishes serving the packets in the priority queue PQ_u .

$Sent_i(s, u)$ is computed as follows:

$$Sent_i(s, u) = \sum_{w=1}^{w=u} Served_i(s, w)$$

Substituting for $Served_i(s, w)$ from Equation (7) in the above, we have,

$$Sent_i(s, u) = \left(\frac{u}{p}\right)Q_i + DC_i(s - 1) - DC_i(s, u) \quad (8)$$

$Sent_i(s, u)$ will be positive only if the the sum of $\left(\frac{u}{p}\right)Q_i$ and $DC_i(s - 1)$ is greater than or equal to the size of the packet at the head of flow i . If this condition is not satisfied then it implies that flow i has not received any service in the first u inner rounds. However, each flow is guaranteed to receive service during at least one inner round within each outer round.

Definition 3 Define $Sent_i(s)$ as the total service received by flow i in outer round s .

Note that, $Sent_i(s)$ is equal to $Sent_i(s, p)$. Therefore, substituting $u = p$ in Equation (8), we get,

$$Sent_i(s) = Q_i + DC_i(s - 1) - DC_i(s) \quad (9)$$

4. Latency Analysis of Pre-order DRR

In deriving an upper bound on the latency of Pre-order DRR, we use the concept of Latency-Rate (\mathcal{LR}) servers first proposed in [11]. The following definitions lead to a formal definition of latency for guaranteed-rate schedulers. The reader is referred to [11] for a more detailed discussion.

Definition 4 An active period of a flow is defined as the maximal interval of time during which at least one packet of the flow is either awaiting service or is in service.

Definition 5 A busy period of a flow is defined as the maximal time interval during which the flow is active if it is served at exactly its reserved rate.

Since the busy period of a flow assumes that a flow is served exactly at its reserved rate, it depends only on the reserved rate and the traffic arrival pattern of the flow. However, an active period of a flow reflects the actual behavior of the scheduler where the instantaneous service offered to the flow varies according to the number of active flows.

Definition 6 Let $Sent_i(t_1, t_2)$ be defined as the total service received by flow i during the time interval (t_1, t_2) .

Note that this notation is identical to the one used in Definition 2. Hence the reader should interpret $Sent_i(\beta, \gamma)$ based on whether β and γ represent two time instant or (β, γ) denotes an inner round in the execution of the Pre-order DRR scheduler.

Let the time instant α_i be the start of a busy period for flow i . Let $t > \alpha_i$ be such that flow i is continuously busy during the time interval (α_i, t) . Let $S_i(\alpha_i, t)$ be the number of bits belonging to packets in flow i that arrive after time α_i and are scheduled during the time interval (α_i, t) . Note that, during this time interval the scheduler may still be serving packets from a previous busy period, and hence $S_i(\alpha_i, t)$ is not necessarily the same as $Sent_i(\alpha_i, t)$.

Definition 7 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,

$$S_i(\alpha_i, t) \geq \max\{0, \rho_i(t - \alpha_i - \Theta_i)\} \quad (10)$$

As defined in [11], a scheduler which satisfies Equation (10) for some non-negative constant value of Θ_i is said to belong to the class of Latency Rate (\mathcal{LR}) servers.

In practice, however it is easier to analyze scheduling algorithms based on the active period of a flow. Let flow i become active at time instant τ_i . Also let $t > \tau_i$ be some time instant such that the flow is continuously active during the time interval (τ_i, t) . Let Θ'_i be the smallest non-negative number such that the following equation is satisfied for all t .

$$Sent_i(\alpha_i, t) \geq \max\{0, \rho_i(t - \alpha_i - \Theta'_i)\} \quad (11)$$

Even though (τ_i, t) may not be a continuously busy period for flow i , it has been proved in [11], that the latency as defined by (10) is bounded by Θ'_i . This allows us to determine the latency bound of a scheduler by considering only the flow active periods.

Theorem 1 The Pre-order DRR scheduler belongs to the class of \mathcal{LR} servers, with an upper bound on the latency Θ_i for flow

i given by,

$$\Theta_i \leq \frac{1}{r} \left\{ \frac{(W - w_i)Q_{min}}{p} + (m - 1) \left(\frac{W}{w_i} + n - 2 \right) \right\} \quad (12)$$

where n is the total number of active flows, p represents the number of priority queues, W is the sum of the weights of all the flows and r denotes the transmission rate of the output link.

Proof. Since the latency of an \mathcal{LR} server can be estimated based on its behavior in the flow active periods, we will prove the theorem by showing that,

$$\Theta'_i \leq \frac{1}{r} \left\{ \frac{(W - w_i)Q_{min}}{p} + (m - 1) \left(\frac{W}{w_i} + n - 2 \right) \right\}$$

Let τ_i be the time instant when flow i becomes active. To prove the statement of the theorem we must consider an active period (τ_i, t) of flow i . We then obtain the lower bound on the total service received by flow i during the time interval under consideration. Lastly, we express the lower bound in the form of Equation (10) to derive the latency bound.

In [6] it has been proved that to obtain a tight upper bound on the latency of the Elastic Round Robin scheduler [8], we need to consider only those active periods (τ_i, t) which satisfy the following two requirements:

1. τ_i coincides with the start of a service opportunity of some flow.
2. Time instant t belongs to a subset of all possible time instants at which the scheduler begins serving flow i .

It can be easily verified that these two conditions are applicable for proving the upper bound on the latency of the Pre-order DRR scheduler. Let $\tau_i^{(e,f)}$ be the time instant marking the start of the service of flow i when flow i is at the head of priority queue PQ_f in round f . In other words, this time instant represents the start of the service opportunity of flow i in inner round (e, f) . Note that $\tau_i^{(e,f)}$ belongs to the set of time instants when the scheduler begins serving flow i . Therefore, in order to determine the latency bound of the Pre-order DRR we need to only consider time intervals $(\tau_i, \tau_i^{(e,f)})$ for all (e, f) in which flow i receives service.

The first step toward analyzing the latency bound involves choosing a suitable time interval $(\tau_i, \tau_i^{(e,f)})$ such that the size of this time interval is the maximum possible. Note that the time instant τ_i may or may not coincide with the start of a new outer round. Let k_0 be the outer round which is in progress at time instant τ_i or which starts exactly at time instant τ_i . In either case, flow i will receive an opportunity to transmit Q_i

worth of data in the k_0 -th round. Let the time instant t_h mark the start of the outer round $(k_0 + h)$. Consider the case when τ_i does not coincide with the time instant t_0 , the start of outer round k_0 , i.e., $\tau_i > t_0$. In this case, the time interval (t_0, τ_i) will be excluded from the time interval under consideration. On the other hand, when τ_i coincides with t_0 , the size of the time interval $(\tau_i, \tau_i^{(e,f)})$ is maximal. We, therefore, assume that the τ_i coincides with the start of the k_0 -th outer round. Fig. 1 illustrates the time interval under consideration assuming that (e, f) is equal to $(k_0 + k, v)$. Note that in Fig. 1, $OR(a)$ represents the a -th outer round and $IR(a, b)$ denotes the inner round (a, b) in the execution of the Pre-order DRR scheduler.

The time interval under consideration, $(\tau_i, \tau_i^{(k_0+k,v)})$, can be split into two sub-intervals:

1. (τ_i, t_k) : This sub-interval includes k outer rounds of execution of the Pre-order DRR scheduler starting at outer round k_0 . Consider the time interval (t_h, t_{h+1}) when outer round $(k_0 + h)$ is in progress. Summing Equation (9) over all n flows,

$$t_{h+1} - t_h = \frac{W}{r} Q_{min} + \frac{1}{r} \sum_{j=1}^n \{DC_j(k_0 + h - 1) - DC_j(k_0 + h)\} \quad (13)$$

Summing the above over k rounds beginning with round k_0 ,

$$t_k - \tau_i = \frac{W}{r} (kQ_{min}) + \frac{1}{r} \sum_{j=1}^n \{DC_j(k_0 - 1) - DC_j(k_0 + k - 1)\} \quad (14)$$

2. $(t_k, \tau_i^{(k_0+k,v)})$: This sub-interval includes the part of the $(k_0 + k)$ -th round prior to the start of the service of flow i when it is at the head of priority queue PQ_v . In the worst-case, flow i will be the last flow to receive service among all the flows which may be present in priority queue PQ_v . In this case, during the sub-interval under consideration, the service received by flow i equals $Sent_i(k_0 + k, v - 1)$ whereas the service received by each flow j among the other $(n - 1)$ flows equals $Sent_j(k_0 + k, v)$. Note that if v equals 1 then flow i does not receive service in this sub-interval. Summing $Sent_i(k_0 + k, v - 1)$ and $Sent_j(k_0 + k, v)$ for each flow j such that $1 \leq j \leq n, j \neq i$ and using Equation (8), we have,

$$\tau_i^{(k_0+k,v)} - t_k = \frac{1}{r} \sum_{\substack{j=1 \\ j \neq i}}^n \left(\frac{v}{p} \right) w_j Q_{min} + \frac{1}{r} \left(\frac{v-1}{p} \right) w_i Q_{min}$$

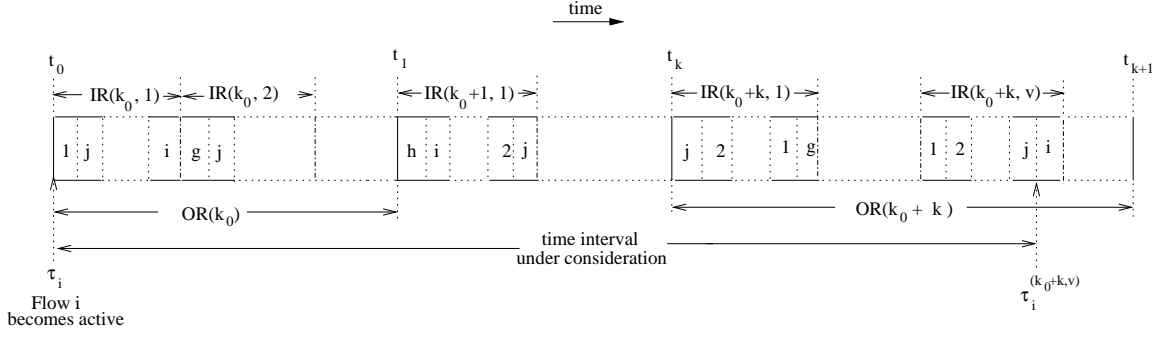


Figure 1. An illustration of the time interval under consideration

$$\begin{aligned}
& + \frac{1}{r} \sum_{\substack{j=1 \\ j \neq i}}^n (DC_j(k_0 + k - 1) - DC_j(k_0 + k, v)) \\
& + \frac{1}{r} (DC_i(k_0 + k - 1) - DC_i(k_0 + k, v - 1)) \quad (15)
\end{aligned}$$

Combining Equations (14) and (15), we have,

$$\begin{aligned}
\tau_i^{(k_0+k, v)} - \tau_i &= \frac{W}{r} (kQ_{min}) + \frac{1}{r} \sum_{\substack{j=1 \\ j \neq i}}^n \left(\frac{v}{p}\right) w_j Q_{min} \\
& + \frac{1}{r} \left(\frac{v-1}{p}\right) w_i Q_{min} \\
& + \frac{1}{r} \sum_{\substack{j=1 \\ j \neq i}}^n (DC_j(k_0 - 1) - DC_j(k_0 + k, v)) \\
& + \frac{1}{r} (DC_i(k_0 - 1) - DC_i(k_0 + k, v - 1)) \quad (16)
\end{aligned}$$

Now since flow i becomes active at the start of outer round k_0 , its deficit count at the start of the k_0 -th outer round, $DC_i(k_0 - 1)$ is equal to zero. Using this fact and the bounds on the deficit count from Equations (2) and (6) in Equation (16), we have,

$$\begin{aligned}
\tau_i^{(k_0+k, v)} - \tau_i &\leq \frac{W}{r} (kQ_{min}) + \frac{1}{r} \sum_{\substack{j=1 \\ j \neq i}}^n \left(\frac{v}{p}\right) w_j Q_{min} \\
& + \frac{1}{r} \left(\frac{v-1}{p}\right) w_i Q_{min} + \frac{(n-1)(m-1)}{r} \\
& - \frac{1}{r} DC_i(k_0 + k, v - 1)
\end{aligned}$$

Solving for k ,

$$\begin{aligned}
k &\geq (\tau_i^{(k_0+k, v)} - \tau_i) \frac{r}{WQ_{min}} - \frac{r}{W} \sum_{\substack{j=1 \\ j \neq i}}^n \left(\frac{v}{p}\right) w_j \\
& - \frac{r}{W} \left(\frac{v-1}{p}\right) w_i - \frac{1}{WQ_{min}} (n-1)(m-1) \\
& + \frac{1}{WQ_{min}} DC_i(k_0 + k, v - 1) \quad (17)
\end{aligned}$$

Note that the total data transmitted by flow i during the time interval under consideration can be expressed as the following summation.

$$Sent_i(\tau_i, \tau_i^{(k, v)}) = Sent_i(\tau_i, t_k) + Sent_i(t_k, \tau_i^{(k, v)}) \quad (18)$$

As explained earlier, $Sent_i(t_k, \tau_i^{(k, v)})$ is the same as $Sent_i(k, v - 1)$. $Sent_i(\tau_i, t_k)$ can be obtained by summing Equation (9) over k outer rounds starting at outer round k_0 . Substituting the result of this summation and Equation (8) in Equation (18) and using the fact that the deficit count of a newly active flow is equal to zero, we have,

$$\begin{aligned}
Sent_i(\tau_i, \tau_i^{(k_0+k, v)}) &= w_i (kQ_{min}) + \left(\frac{v-1}{p}\right) w_i Q_{min} \\
& - DC_i(k_0 + k, v - 1) \quad (19)
\end{aligned}$$

Using (17) to substitute for k in (19), we get,

$$\begin{aligned}
Sent_i(\tau_i, \tau_i^{(k_0+k, v)}) &\geq \frac{w_i r}{W} (\tau_i^{(k_0+k, v)} - \tau_i) \\
& - \frac{w_i}{W} \left(\frac{v}{p}\right) (W - w_i) Q_{min} - \frac{w_i}{W} \left(\frac{v-1}{p}\right) w_i Q_{min} \\
& - \frac{w_i}{W} (n-1)(m-1) + \frac{w_i}{W} DC_i(k_0 + k, v - 1) \\
& + \left(\frac{v-1}{p}\right) w_i Q_{min} - DC_i(k_0 + k, v - 1) \quad (20)
\end{aligned}$$

Now, since the reserved rates are proportional to the weights assigned to the flows as given by (1), and since the sum of the reserved rates is no more than the link rate r , we have,

$$\rho_i \leq \frac{w_i}{W} r \quad (21)$$

Using Equation (21) in Equation (20), and simplifying we get,

$$\begin{aligned}
Sent_i(\tau_i, \tau_i^{(k_0+k, v)}) &\geq \rho_i (\tau_i^{(k_0+k, v)} - \tau_i) \\
& - \frac{\rho_i}{r} \left(\frac{v}{p}\right) (W - w_i) Q_{min} - \frac{\rho_i}{r} \left(\frac{v-1}{p}\right) (W - w_i) Q_{min} \\
& - \frac{\rho_i}{r} (n-1)(m-1) - \frac{\rho_i}{r} DC_i(k_0 + k, v - 1) \left(\frac{W}{w_i} - 1\right)
\end{aligned}$$

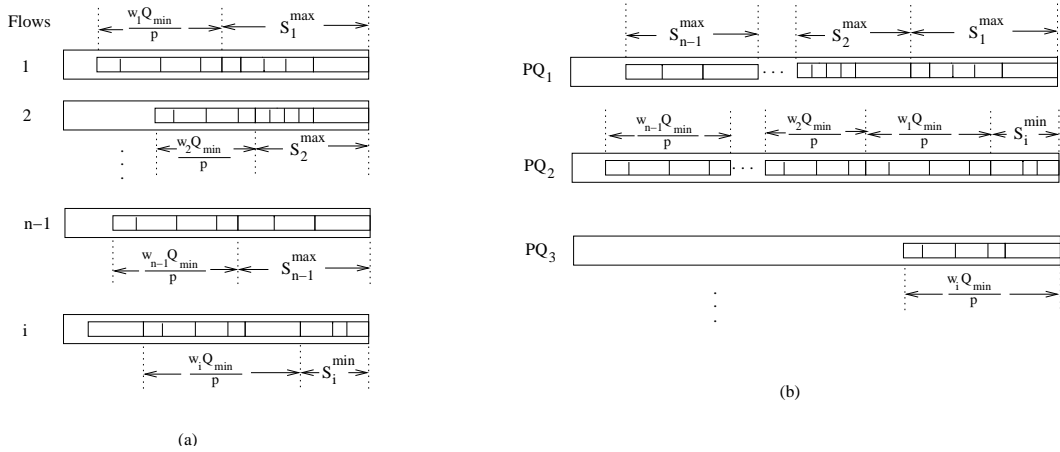


Figure 2. (a) Input Pattern (b) Packet Classification in the priority queues

Simplifying further, and noting that the latency bound reaches the upper bound when $DC_i(k_0 + k, v - 1)$ equals $(m-1)$,

$$\begin{aligned}
 Sent_i(\tau_i, \tau_i^{(k_0+k, v)}) &\geq \max \left\{ 0, \rho_i \left(\tau_i^{(k_0+k, v)} - \tau_i \right. \right. \\
 &\quad \left. \left. - \frac{1}{r} \left(\frac{(W - w_i)Q_{min}}{p} \right. \right. \right. \\
 &\quad \left. \left. \left. + (m - 1) \left(\frac{W}{w_i} + n - 2 \right) \right) \right) \right\} \quad (22)
 \end{aligned}$$

As discussed earlier, flow i will experience its worst latency during an interval $(\tau_i, \tau_i^{(k_0+k, v)})$ for some inner round $(k_0 + k, v)$. Therefore, from Equation (22), the statement of the theorem is proved. \square

We now proceed to show that the latency bound given by Theorem 1 is tight by illustrating a case when the bound is actually achieved. Let \mathbf{F} represent the set of all n flows. Assume that flow i becomes active at a certain time instant τ_i which also coincides with the start of certain outer round k_0 . Since the arrival of a packet into the empty buffer of a flow signals the start of a busy period of the flow, τ_i is also the start of its busy period. Assume that for any time instant $t, t > \tau_i$, a total of n flows, including flow i , are active. Also, assume that the summation of the reserved rates of all the n flows is equal to the transmission rate of the output link, r . Therefore, we have, $\rho_i = \frac{w_i}{W}r$. Since flow i became active at time τ_i , its deficit count at the start of outer round k_0 is 0. Let the deficit count of all the other $(n - 1)$ flows be equal to the maximum value of $(m - 1)$. Using Equations (6) and (7), it is seen that the maximum service received by a flow j during an inner round, S_j^{max} is given by,

$$S_j^{max} = \frac{w_j Q_{min}}{p} + (m - 1) \quad (23)$$

On a similar note, the minimum service received by flow j during an inner round, S_j^{min} , provided it is present in the priority queue being served, is given by,

$$S_j^{min} = \frac{w_j Q_{min}}{p} - (m - 1) \quad (24)$$

Fig. 2(a) illustrates a part of the input traffic present in the queues of the n flows at the start of outer round k_0 . Fig. 2(b) shows how the *Classifier* module of the Pre-order DRR scheduler classifies these packets into the priority queues using Equation (4). From Fig. 2(b) it can be seen that, except for flow i , all the other $(n - 1)$ flows have packets classified into the highest priority queue PQ_1 . Prior to the service of the first packet of flow i , each flow $j, j \in \mathbf{F}, j \neq i$, transmits S_j^{max} worth of data. Hence the cumulative delay until flow i receives service, X , is given by,

$$X = \sum_{\substack{j \in \mathbf{F} \\ j \neq i}} \frac{S_j^{max}}{r}$$

Substituting for S_j^{max} from Equation (23), we have,

$$X = \frac{1}{r} \left(\frac{(W - w_i)Q_{min}}{p} + (n - 1)(m - 1) \right) \quad (25)$$

Also the total flow i data that is served from PQ_2 equals S_i^{min} .

Even though X represents the time for which flow i has to wait until it starts receiving service, Equation (10) does not hold true if we substitute X as Θ_i . This is because in time interval $(\tau_i, \tau_i + X)$ flow i has not yet started receiving at its guaranteed rate. We assume that the latency, Θ_i is given by,

$$\Theta_i = X + Y \quad (26)$$

A plot of the service received by flow i against time is illustrated in Fig. 3. In order to determine the value of Y we

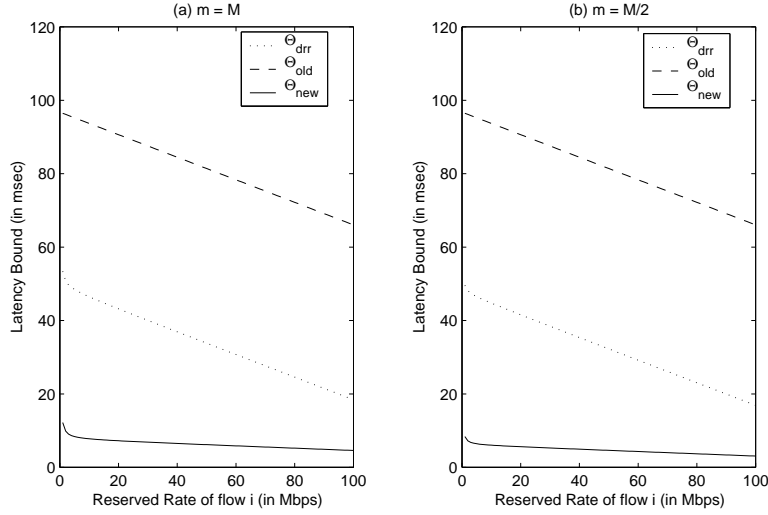


Figure 4. Comparison of the latency bounds

the latency bound of Pre-order DRR scheduler is significantly lower than that of DRR.

In order that the reader may fully appreciate the difference between the new and the old latency bounds, we provide a comparison of these two latency bounds of Pre-order DRR within the context of a practical example. Θ_i^{drr} is also included to illustrate the improvement in latency achieved by Pre-order DRR. Let us assume that a total of 100 flows are multiplexed onto an output link whose transmission rate, r is 150 Mbps. Assume that M is equal to 576 bytes and let Q_{min} be equal to M . Also assume that ρ_{min} is equal to 0.1Mbps and that the output link is completely utilized, i.e. $\sum_{i=1}^n \rho_i = r$. Note that this implies that sum of all the weights is $150/0.1 = 1500$. Let the number of queues in the priority queue module of the Pre-order DRR scheduler, p be equal to 10. We compare the latency bounds, Θ_i^{old} , Θ_i^{new} and Θ_i^{drr} for flow i as a function of its reserved rate, ρ_i , for two values of m : (a) $m = M$, (b) $m = M/2$. Fig. 4 illustrates a plot of these latency bounds of flow i for both values of m . Note that the expressions for Θ_i^{old} , Θ_i^{new} and Θ_i^{drr} depend on the sum of the weights of all the flows other than flow i . Therefore, the weights of the flows other than flow i are not discussed in the context of this illustration. From Fig. 4, it can be seen that Θ_i^{old} is such a loose bound that it is in fact even greater than Θ_i^{drr} . On the other hand, Θ_i^{new} is a much tighter bound.

6 Concluding Remarks

We conclude the paper with Table 2, a summary comparison of the work complexity and the latency bounds of several guaranteed-rate scheduling disciplines. Note that GPS is

an ideally fair but unimplementable scheduler with a latency bound of exactly 0. The latencies in Table 2, except ERR, DRR and Pre-order DRR, are derived in [3]. The latency of ERR is derived in [6] and the latency of DRR is derived in [7]. In this table, as in the rest of this paper, M is the size of the largest packet that may potentially arrive during the execution of a scheduling algorithm. Recall that m is the size of the largest packet that *actually* arrives during the execution of the scheduler. Typically, $M \gg m$, since in most networks including the Internet, the vast majority of the packets are of much smaller size than the maximum possible size of a packet [13, 15].

References

- [1] J. C. R. Bennett and H. Zhang. WF²Q : Worst-case fair weighted fair queueing. In *Proceedings of IEEE INFOCOM*, pages 120–128, San Francisco, CA, March 1996.
- [2] A. Demers, S. Keshav, and S. Shenker. Design and analysis of a fair queuing algorithm. In *Proceedings of ACM SIGCOMM*, pages 1–12, Austin, September 1989.
- [3] D. Stiliadis. *Traffic Scheduling in Packet-Switched Networks: Analysis, Design and Implementation*. PhD thesis, University of California, Santa Cruz, 1996.
- [4] S. J. Golestani. A self-clocked fair queuing scheme for broadband applications. In *Proceedings of IEEE INFOCOM*, pages 636–646, Toronto, Canada, June 1994.
- [5] S. S. Kanhere and H. Sethu. Fair, efficient and low-latency packet scheduling using nested deficit round robin. In *Proceedings of the IEEE Workshop on High Performance Switching and Routing*, pages 6–10, Dallas, TX, May 2001.
- [6] S. S. Kanhere and H. Sethu. Low-latency guaranteed-rate scheduling using elastic round robin. *Computer Communication*, 25(14):1315–1322, 2002.

Table 2. A Comparison between Scheduling Disciplines

Scheduling Discipline	Complexity	Latency Bound for flow i
GPS [9]	-	0
Weighted Fair Queueing [2]	$O(\log n)$	$\frac{m}{r} + \frac{m}{\rho_i}$
Self-Clocked Fair Queueing [4]	$O(\log n)$	$\frac{(n-1)m}{r} + \frac{m}{\rho_i}$
Virtual Clock [16]	$O(\log n)$	$\frac{m}{r} + \frac{m}{\rho_i}$
Frame-based Fair Queueing [12]	$O(\log n)$	$\frac{m}{r} + \frac{m}{\rho_i}$
Deficit Round Robin [10]	$O(1)$	$\frac{1}{r} \left\{ (W - w_i)M + (m - 1) \left(\frac{W}{w_i} + n - 2 \right) \right\}$
Elastic Round Robin [8]	$O(1)$	$\frac{(W - w_i)m + (n - 1)(m - 1)}{r}$
Pre-order Deficit Round Robin [14]	$O(\log p)$	$\frac{1}{r} \left\{ \frac{(W - w_i)M}{p} + (m - 1) \left(\frac{W}{w_i} + n - 2 \right) \right\}$

- [7] S. S. Kanhere and H. Sethu. On the latency bound of deficit round robin. In *Proceedings of the International Conference on Computer Communications and Networks*, Miami, FL, October 2002.
- [8] S. S. Kanhere, H. Sethu, and A. B. Parekh. Fair and efficient packet scheduling using elastic round robin. *IEEE Transactions on Parallel and Distributed Systems*, 13(3):324–336, March 2002.
- [9] A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control—the single node case. In *Proceedings of IEEE INFOCOM*, pages 915–924, Florence, Italy, May 1992.
- [10] M. Shreedhar and G. Varghese. Efficient fair queuing using deficit round-robin. *IEEE Transactions on Networking*, 4(3):375–385, June 1996.
- [11] D. Stiliadis and A. Varma. Latency-rate servers: A general model for analysis of traffic scheduling algorithms. *IEEE Transactions on Networking*, 6(3):611–624, October 1996.
- [12] D. Stiliadis and A. Varma. Efficient fair queuing algorithms for packet-switched networks. *IEEE Transactions on Networking*, 6(2):175–185, April 1998.
- [13] K. Thompson, G. J. Miller, and R. Wilder. Wide-area internet traffic patterns and characteristics. *IEEE Network*, 11(6):10–23, November/December 1997.
- [14] S. Tsao and Y. Lin. Pre-order deficit round robin: a new scheduling algorithm for packet-switched networks. *Computer Networks*, 35(2-3):287–305, February 2001.
- [15] I. Widjaja and A. I. Elwalid. Performance issues in vc-merge capable switches for multiprotocol label switching. *IEEE Journal on Selected Areas in Communications*, 17(6):1178–1189, June 1999.
- [16] L. Zhang. Virtual clock: A new traffic control algorithm for packet switching networks. In *Proceedings of ACM SIGCOMM*, pages 19–29, Philadelphia, PA, September 1990.