

Research Article

Packet Scheduling in High-speed Networks Using Improved Weighted Round Robin

Guikai Liu and Qing Li

School of Computer Science and Engineering, Hunan University of Science and Technology,
Xiangtan, 411201, China

Abstract: A variety of applications with different QoS requirements are supported simultaneously in the high-speed packet-switched networks, packet scheduling algorithms play a critical role in guaranteeing the performance of routing and switching devices. This study presents a simple, fair, efficient and easily implementary scheduling algorithm, called Successive Minimal-weight Round Robin (SMRR). In each round, SMRR provides the same service opportunity, which is equivalent to the minimal weight of the current round, for all active data flows. On the basis of the concept of Latency-Rate (LR) servers, we obtain the upper bound on the latency of SMRR and WRR (Weighted Round Robin) respectively and the results indicate that SMRR makes a significant improvement on the latency bound in comparison to WRR. We also discuss the fairness and implementation complexity of SMRR and the theoretical analysis shows that SMRR preserves the good implementation complexity of $O(1)$ with respect to the number of flows and has better fairness than WRR.

Keywords: Complexity, high-speed networks, latency bound, minimal weight, packet scheduling, relative fairness, Weighted Round Robin (WRR)

INTRODUCTION

There are many kinds of services with different QoS requirements in Internet. Packets belonging to different traffic flows often share links in their respective paths towards their destinations. Switches and routers want to schedule these traffics for supporting different service levels; therefore, the performances of routing-switching devices are tightly interrelated to the used packet scheduling algorithms. Packet scheduling algorithms can be broadly classified into two categories: time-stamp based scheduling and round-robin based scheduling. Time-stamp based scheduling algorithms such as WFQ (Weighted Fair Queuing) (Demers *et al.*, 1998), WF²Q (Worst-case Fair Weighted Fair Queuing) (Bennett and Zhang, 1996), VC (Virtual Clock) (Zhang, 1990), SCFQ (Self-Clocked Fair Queuing) (Golestani, 1994) and SFQ (Start-time Fair Queuing) (Goyal *et al.*, 1997) maintain two time-stamps for each packet to indicate its start-serving time and end-serving time respectively, then sort these time and send out the packet with the least end-serving time. They achieve good fairness and low latency. However, they are not very efficient due to the complexity involved in computing and sorting each packet's time. On the other hand, in round-robin based scheduling algorithms such as WRR (Weighted Round-Robin) (Hemant and Madhow, 2003), DRR (Deficit Round-Robin) (Shreedhar and Varghese, 1996) and

SRR (Smoothed Round Robin) (Guo, 2001), the scheduler simply serves all non-empty queues in a round-robin manner. These algorithms neither maintain a time-stamp for every service flow, nor perform sorting among the packets. Most of them have $O(1)$ complexity with respect to the number of flows. As a matter of fact, round-robin is used in many other fields besides packet scheduling due to its simple, applied and good efficiency. Research and application aiming at round-robin algorithm are very broad (Hu *et al.*, 2008; Lin *et al.*, 2011; He *et al.*, 2012).

Time-stamp based packet scheduling algorithms are to approximately emulate the most ideal packet scheduling algorithm-GPS (Generalized Processor Sharing) (Parekh and Gallager, 1992). However, their design complexities are increased in order to get perfect latency characteristics. So round-robin based packet scheduling algorithms are more attractive for implementation in high-speed packet networks. In this study, we present a new packet scheduling algorithm, termed Successive Minimal-weight Round Robin (SMRR), with better fairness and latency characteristic compared to Weighted Round Robin (WRR). In each round, every flow always gains the equivalent service opportunities to the least weight of the current round. It efficiently avoids the situation that some flows cannot gain service probably for a long time. Generally the opportunity of a flow, namely the number of packets which are sent to the output link with permission in a

Corresponding Author: Guikai Liu, School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, 411201, China

This work is licensed under a Creative Commons Attribution 4.0 International License (URL: <http://creativecommons.org/licenses/by/4.0/>).

```

Initialize: (Invoked when the scheduler is initialized)
ActiveFlowList=NULL;
SurplusFlowList=NULL;
VisitFlowCount=0;
for (i=1; i<=n; i=i+1) {
    PacketNumberCounti=0;
    Assigning a weight Wi to each flow i; }

Enqueue: (Invoked when a packet arrives)
i=QueueInWhichPacketArrives;
if (ExistInActiveFlowList (i)==FALSE) then
    AddToActiveFlowList (i);
    Increment SizeOfActiveFlowList;
    PacketNumberCounti=0;
end if;

Dequeue:
while (TRUE) do
    if (ActiveFlowListIsEmpty==FALSE) then
        SurplusFlowList=ActiveFlowList;
        for (i=1; i<=n; i=i+1) SurplusWeighti=Wi;
    else
        Waiting for arrival of a packet;
    end if;
    while (SurplusFlowListIsEmpty==FALSE) do
        for (i=1; i<=n; i=i+1)
            if (ExistInSurplusFlowList (i)==TRUE) then
                PacketNumberCounti=0;
            endif;
        VisitFlowCount=SizeOfSurplusFlowList;
        Wmin=MinimumOfSurplusWeights;
        while (VisitFlowCount>0) do
            i=HeadOfSurplusFlowList;
            RemoveHeadOfSurplusFlowList;
            do
                TransmitPacketFromQueue (i);
                Increment PacketNumberCounti;
                Decrement SurplusWeighti;
            while (QueueIsEmpty==FALSE)&&(SurplusWeight>0)
                &&(PacketNumberCount<Wmin)
            if (QueueIsEmpty==FALSE)&&(SurplusWeight>0) then
                AddQueueToSurplusFlowList (i);
            else
                Decrement SizeOfSurplusFlowList;
                if (QueueIsEmpty==TRUE) then
                    RemoveQueueFromActiveFlowList (i);
                endif;
            endif
            Decrement VisitFlowCount;
        end while
    end while
end while

```

Fig. 1: Pseudo-code for SMRR scheduling packets

round, is proportional to its weight. WRR sends out all the packets whose number is equivalent with a flow's weight at a time and a flow with large weight may let other flows be waiting for a long time. In contrast, SMRR sends a flow's packets whose number is equivalent with its weight through many times; only

when a flow's weight is the least, it can send out all of its packets at a time. So SMRR can assure the service opportunity of every flow and balance their waiting latency. WRR is an earlier scheduling algorithm and applied long before. In recent years, many literatures put forward many queue scheduling algorithms based

on WRR. For example, according to the current queue length, Pan and Zhang (2012) dynamically set the maximum byte number that can be sent during a service time in order to guarantee latency characteristics and relative fairness of low-weighted services. BSTLRR (Xiong and Zhang, 2012) schedules those data flows firstly, which have higher real-time requirement. TIIWRR (Zhang *et al.*, 2011) ameliorates UIWRR, also called WRR2 in (Hemant and Madhow, 2003), to solve the collision problem when UIWRR computes packet sending sequence. Based on the very limited buffering capability of optical packet switching network, Cao *et al.* (2010) introduce an algorithm called United Weight Round Robin (UWRR) for scheduling multi-services in the network. As far as a scheduling algorithm is concerned, there are three very important attributes, namely latency, fairness and complexity. These three attributes have an effect on the practical application of a scheduling algorithm. Therefore, this study will lay special stress on analyzing these three attributes of SMRR and compare them with WRR.

SUCCESSIVE MINIMAL-WEIGHT ROUND ROBIN

A pseudo-code implementation of the SMRR scheduling algorithm is shown in Fig. 1, including Initialize Module, Enqueue Module and Dequeue Module. The Initialize Module is to initialize the SMRR packet scheduler when network node has demand on scheduling packets. Enqueue Module is called whenever a new packet arrives at a flow. The packet will enter into the correlative queue and wait for being sent. Dequeue Module schedules packets from the queues corresponding to different flows.

Therefore, we can define the scheduling framework of SMRR as: suppose there are n (n is a natural number and $n > 0$) data flows sharing one output link; packets of each flow enter into correlative queue before being sent to the output link; packet scheduling module uses SMRR algorithm to scheduling each packet so that the bandwidth resource of the output link is allocated fairly.

A data flow is defined as active if its queue is not empty or its packets are being scheduling. All the active flows are put into a list and this list is called "Active Flow List". When a flow changes from active to inactive, it will be removed from Active Flow List. When a flow changes from inactive to active, it will be appended at the end of Active Flow List.

Every data flow is designated a "weight" through which indicates its priority, QoS requirements and so on. When a flow's weight (namely its service opportunity) is not used up, we say that its "Surplus Weight" is more than zero. All the flows which have positive Surplus Weight are put into a list which is called "Surplus Flow List". When a flow's weight is used up, namely its weight already equals 0; this flow

will be removed from Surplus Flow List. If a data flow belongs to Surplus Flow List, it belongs to Active Flow List definitely, but not vice versa.

There are two types of rounds in SMRR-primary round and secondary round. A primary round is defined as the process during which the data flows, included in Active Flow List at a time instant $T1$ ($T1 > 0$), are accessed by packet scheduling module. The newcomer flows or those become active once again can be appended at the end of Active Flow List, but they will be accessed in the next round. The definition of secondary round, similar with primary round, is the process during which the data flows included in Surplus Flow List are accessed by packet scheduling module. A primary round may involve one or more secondary rounds.

In order to express the number of data flows wanted to be accessed in a round, SMRR introduces a counter, which is called "Visit Flow Count", to record the number of data flows. At the beginning of a round, Visit Flow Count equals the number of data flows in Surplus Flow List. When packet scheduling module finishes accessing a data flow, the value of Visit Flow Count will be minus 1. At last, when Visit Flow Count equals 0, it means that the current round is over.

Another counter, called "Packet Number Count", is used to record the number of packets that a data flow has sent out totally in a round and every data flow has a Packet Number Count. The initial value of Packet Number Count is 0 before a round begins and then Packet Number Count will be increased by 1 after the corresponding flow sends out one packet.

SMRR LATENCY ANALYSIS

Stiliadis and Verma (1998) defined a general class of schedulers, called Latency-Rate (LR) Servers. Here schedulers are those servers or devices which run specific scheduling algorithm. The authors also developed and defined a notion of latency and determined an upper bound on the latency for a number of schedulers that belong to the class of LR servers. There into the notion of latency is based on the length of time it takes a new flow to begin receiving service at its guaranteed rate and pointed out latency is directly relevant to the size of the playback buffers required in real-time streaming applications. In this study, SMRR's latency characteristic will be analyzed by using the concept of Latency-Rate Servers.

Consider an output link of transmission rate r , access to which is controlled by the SMRR scheduler. Let n be the total number of flows and let ρ_i be the scheduling rate for flow i . Let ρ_{min} be the lowest of these scheduling rates. Since all the flows share the same output link, the sum of the scheduling rates is no more than the transmission rate of the output link, namely $\sum_{i=1}^n \rho_i \leq r$. In order that each flow receives service

proportional to its guaranteed service rate, the SMRR scheduler assigns a weight to each flow. The weight assigned to flow i , w_i is given by:

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

Apparently, for $1 \leq i \leq n$, $w_i \geq 1$.

Let Φ_i represent the amount of data that flow i is permitted to send during each round robin service opportunity and let Φ_{\min} be the amount of data permitted to the flow with the lowest scheduling rate. The amount of data permitted to flow i is given by $\Phi_i = w_i \Phi_{\min}$. Thus, the amounts of data permitted to the flows are in proportion of their scheduling rates. In addition, let $A_i(\tau, t)$ represent the number of arrived packets in flow i during the time interval (τ, t) .

Definition 1: Define W as the sum of the weights of all active flows that are being served by the SMRR scheduler.

Definition 2: An active period of a flow is defined as the maximal interval of time during which it has at least one packet awaiting service or in service. A flow remains continuously active during its active period.

Definition 3: A busy period of flow i is defined as the maximal interval of time (τ_1, τ_2) such that at any time $t \in (\tau_1, \tau_2)$ the accumulated arrivals of flow i since the beginning of the interval do not fall below the total service received during the interval at a rate of exactly ρ_i . That is:

$$A_i(\tau_1, t) \geq \rho_i(t - \tau_1) \quad (2)$$

A flow is active during its busy period, but the active period is different from the busy period of a flow, in the sense that it reflects the actual behavior of the scheduler because the instantaneous service offered to the flow varies according to the number of active flows. Suppose $\text{Sent}_i(t_1, t_2)$ represent the amount of service received by flow i during the time interval (t_1, t_2) . Let the time instant α_i be the start of a busy period for flow i . Suppose $t > \alpha_i$ and flow i is continuously busy during the time interval (α_i, t) . Define $S_i(\alpha_i, t)$ as the number of packets in flow i that arrive after time α_i and are scheduled during the interval of time (α_i, t) . Obviously, the busy period of a flow must be its active period, but the active period is not always its busy period and thus $S_i(\alpha_i, t)$ is not necessarily the same as $\text{Sent}_i(\alpha_i, t)$.

Definition 4: The latency of flow i is defined as the minimum non-negative constant Θ_i that satisfies the following for all possible busy period of the flow:

$$S_i(\alpha_i, t) \geq \max \{0, \rho_i(t - \alpha_i - \Theta_i)\} \quad (3)$$

As defined in Stiliadis and Verma (1998), a scheduler which satisfies Eq. (3) for some non-negative constant value of Θ_i is said to belong to the class of Latency Rate (LR) servers.

Note that even if the definition of the latency is based on flow busy periods, actually it is easier to analyze scheduling algorithms based on the active period of a flow.

Lemma 1: Suppose there is a scheduling server S and flow i becomes active at the instant of time τ_i . Let $t > \tau_i$ be some instant of time 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 is satisfied for all t :

$$\text{Sent}_i(\tau_i, t) \geq \max \{0, \rho_i(t - \tau_i - \Theta_i')\} \quad (4)$$

Even though (τ_i, t) may not be a continuously busy period for flow i , the latency as defined by Eq. (3) is bounded by Θ_i' .

Proof: Since a busy period must be an active period, it will be proven so long as any one busy period (α_k, β_k) during active periods satisfies Eq. (3):

If $\alpha_k = \tau_i$, $t \in (\alpha_k, \beta_k)$, then $S_i(\alpha_k, t) = \text{Sent}_i(\alpha_k, t) \geq \max \{0, \rho_i(t - \alpha_k - \Theta_i')\}$

If $\alpha_k > \tau_i$, $t \in (\alpha_k, \beta_k)$, then $S_i(\alpha_k, t) = \text{Sent}_i(\tau_i, t) - \text{Sent}_i(\tau_i, \alpha_k)$

Since (τ_i, α_k) is not the busy period of flow i , therefore, $\text{Sent}_i(\tau_i, \alpha_k) \leq A_i(\tau_i, \alpha_k) < \rho_i(\alpha_k - \tau_i)$ and then $S_i(\alpha_k, t) = \text{Sent}_i(\tau_i, t) - \text{Sent}_i(\tau_i, \alpha_k) > \max \{0, \rho_i(t - \tau_i - \Theta_i')\} - \rho_i(\alpha_k - \tau_i) = \max \{0, \rho_i(t - \alpha_k - \Theta_i')\}$.

Synthetically, $S_i(\alpha_k, t) \geq \max \{0, \rho_i(t - \alpha_k - \Theta_i')\}$. Hence S is Latency-Rate server and the latency is less than or equals Θ_i' .

Lemma 1 allows us to determine the latency bound of a scheduler by considering only those periods during which a flow is continuously active.

Lemma 2: The latency experienced by flow i during the active period (τ_i, t) will reach its upper bound, Θ_i' , only if the time instant, τ_i , at which flow i becomes active, is the start time of serving another flow.

Proof: Assume that flow j ($j \neq i$) receives its service opportunity during the time interval (t_1, t_2) , $t_1 \leq \tau_i < t_2$. If τ_i is not the start time that j receives service opportunity, then $\tau_i > t_1$. That is, the time interval (t_1, τ_i) is a part of the service opportunity of flow j , but it is not included in the latency experienced by flow i . On the other hand, when $\tau_i = t_1$, τ_i is the start of the service opportunity of flow j . So, the time for which flow i has

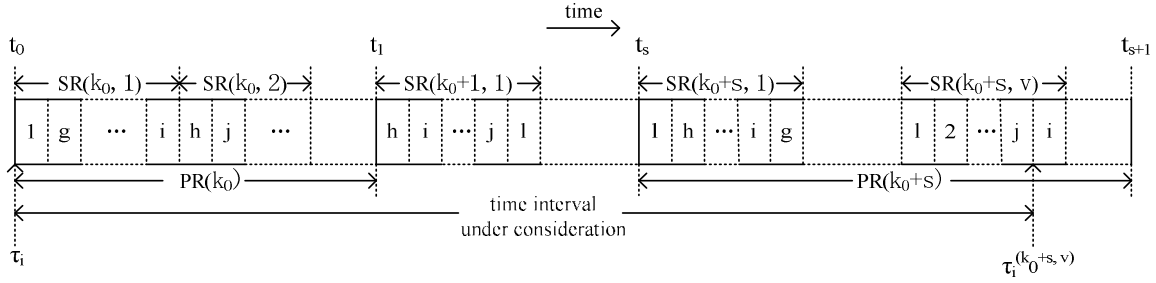


Fig. 2: SMRR time interval under consideration

to wait before receiving any service will involve the time interval (t_1, t_2) . Obviously, when τ_i is the start time of the service opportunity of another flow, the latency experienced by flow i is always greater.

Lemma 3: The latency experienced by flow i during the active period (τ_i, t) will reach its upper bound, Θ_i' . t belongs to such a time set T_i and T_i is the set of all time instants at which the scheduler begins serving flow i .

Proof: Assume that t_1 and t_2 are any two consecutive time instants belong to T_i . Consider an instant of time t , $t_1 < t < t_2$. When flow i is receiving service at time instant t , the amount of service received by flow i is $r(t - t_1)$ during the time interval (t_1, t) , where r is the link rate. Clearly, flow i is receiving service at the rate ρ_i or higher. Thus, the worst-case latency of flow i is no worse than that until time t_1 . On the other hand, when some other flow is receiving service at time instant t , flow i does not receive any service during the time interval (t, t_2) . So the latency experienced by flow i includes the time interval (t, t_2) . In this case, the worst-case latency of flow i is no worse than that until time t_2 . The above two cases show that the worst-case latency of a flow during (t_1, t_2) is equal to the latency until either t_1 or t_2 . The lemma is proved.

Just as mentioned in above section, when flow i receives service from SMRR scheduler, rounds can be primary and secondary and a primary round may involve one or more secondary rounds. The primary round measures the amount of data Φ_i , which is permitted by flow i during a primary round. Service in a secondary round equals the lowest amount of data among all flows in every secondary round, namely W_{\min} in SMRR algorithm description. Service in a primary round equals the sum of service of all nested secondary rounds. If let k be the primary round and let v be the secondary round, then (k, v) expresses that a flow receives the v -th service in the k -th primary round and let $\tau_i^{(k, v)}$ be the time instant at which flow i begins to receive the (k, v) -th service. Assume that flow i becomes active at time instant τ_i , therefore, in order to determine the latency bound of SMRR, according to Lemma 3, we need to only consider time interval $(\tau_i, \tau_i^{(k, v)})$ for all (k, v) in which flow i receives service.

For analyzing the latency bound, we need to select a suitable time interval $(\tau_i, \tau_i^{(k, v)})$ such that the size of this time interval is the maximum possible. The start time instant τ_i may or may not coincide with the start of a new primary round. Assume that the k_0 -th primary round is in progress or which starts exactly at time instant τ_i and let the time instant t_h represent the start time of the $(k_0 + h)$ -th primary round. Consider the case when τ_i does not coincide with the time instant t_0 , the start of primary 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 $(\tau_i, \tau_i^{(k, v)})$ is maximal. According to Lemma 2, we assume that the τ_i coincides with the start of the k_0 -th primary round. Figure 2 illustrates the time interval under consideration supposing that (k, v) is equal to $(k_0 + s, v)$. Note that in Fig. 2, PR (a) represents the a -th primary round and SR (a, b) denotes the secondary round (a, b) in the execution of SMRR scheduler. In addition, the set G_{FIN} includes those flows that have completed receiving their allocated service in the duration of the $(k_0 + s)$ -th primary round at the time instant $\tau_i^{(k_0+s, v)}$.

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

$$\theta_i \leq \frac{1}{r} \sum_{j \in G_{FIN}} w_j \Phi_{\min} + \frac{1}{r} \sum_{\substack{j \in (n-G_{FIN}) \\ j \neq i}} w_j \Phi_{\min} - \Psi \left(\frac{1}{\rho_i} - \frac{1}{r} \right) \Phi_{\min} \quad (5)$$

Hereinto, the value of the parameter Ψ as follows:
 $\Psi = 0$, when w_i is the smallest among n weights
 $\Psi = 1$, other cases

Proof: According to Lemma 1, the latency of an LR server can be estimated based on its behavior in the flow active periods, so we will prove the theorem by showing that:

$$\theta_i' \leq \frac{1}{r} \sum_{j \in G_{FIN}} w_j \Phi_{\min} + \frac{1}{r} \sum_{\substack{j \in (n-G_{FIN}) \\ j \neq i}} w_j \Phi_{\min} - \Psi \left(\frac{1}{\rho_i} - \frac{1}{r} \right) \Phi_{\min}$$

On the basis of the discussion above, the time interval $(\tau_i, \tau_i^{(k_0+s, v)})$ can be split into two sub-intervals:

- **(τ_i, t_s) :** This sub-interval includes s primary rounds of execution of SMRR scheduler starting at primary round k_0 . Now assume that primary round $(k_0 + h)$ is in progress during the time interval (t_h, t_{h+1}) . For all n flows:

$$t_{h+1} - t_h = \frac{W}{r} \Phi_{\min} \quad (6)$$

Summing the above over s rounds beginning with primary round k_0 :

$$t_s - \tau_i = \frac{W}{r} (s\Phi_{\min}) \quad (7)$$

- **$(t_s, \tau_i^{(k_0+s, v)})$:** This sub-interval includes the part of the $(k_0 + s)$ -th primary round prior to the start of the service of flow i during the v -th secondary round. In the worst-case, flow i will be the last flow to receive service among all the flows. In this case, during the sub-interval $(t_s, \tau_i^{(k_0+s, v)})$, the service received by flow i comes from the former $v-1$ secondary rounds, whereas the other $n-1$ flows have completed the service of v secondary rounds. If $v = 1$, then flow i does not receive service in this sub-interval.

Let $Sent_i(k, v)$ represent the total service received by flow i since the start of the k -th primary round until the time instant when SMRR scheduler finishes the service of the v -th secondary round, then:

$$\tau_i^{(k_0+s, v)} - t_s = \frac{1}{r} [Sent_i(k_0 + s, v-1) + \sum_{\substack{j=1 \\ j \neq i}}^n Sent_j(k_0 + s, v)] \quad (8)$$

Sorting all the n weights in ascending order and let w_{v-1} and w_v represent the $(v-1)$ -th and the v -th weight respectively, we have:

$$\tau_i^{(k_0+s, v)} - t_s = \frac{1}{r} [\sum_{j \in G_{FIN}} w_j \Phi_{\min} + \sum_{\substack{j \in (n-G_{FIN}) \\ j \neq i}} w_v \Phi_{\min} + w_{v-1} \Phi_{\min}] \quad (9)$$

Since $w_v \leq w_i$, therefore:

$$\tau_i^{(k_0+s, v)} - t_s \leq \frac{1}{r} [\sum_{j \in G_{FIN}} w_j \Phi_{\min} + \sum_{\substack{j \in (n-G_{FIN}) \\ j \neq i}} w_i \Phi_{\min} + w_{v-1} \Phi_{\min}] \quad (10)$$

Combining Eq. (7) and (10), we have:

$$\tau_i^{(k_0+s, v)} - \tau_i \leq \frac{W}{r} (s\Phi_{\min}) + \frac{1}{r} [\sum_{j \in G_{FIN}} w_j \Phi_{\min} + \sum_{\substack{j \in (n-G_{FIN}) \\ j \neq i}} w_i \Phi_{\min} + w_{v-1} \Phi_{\min}] \quad (11)$$

Solving for s :

$$s \geq (\tau_i^{(k_0+s, v)} - \tau_i) \frac{r}{W\Phi_{\min}} - \frac{1}{W} [\sum_{j \in G_{FIN}} w_j + \sum_{\substack{j \in (n-G_{FIN}) \\ j \neq i}} w_i + w_{v-1}] \quad (12)$$

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

$$Sent_i(\tau_i, \tau_i^{(s, v)}) = Sent_i(\tau_i, t_s) + Sent_i(t_s, \tau_i^{(s, v)}) \quad (13)$$

As explained earlier, $Sent_i(t_s, \tau_i^{(s, v)})$ is the same as $Sent_i(s, v-1)$. $Sent_i(\tau_i, t_s)$ can be obtained by summing the amount of data of flow i over s primary rounds starting at primary round k_0 . We get:

$$Sent_i(\tau_i, \tau_i^{(k_0+s, v)}) = w_i (s\Phi_{\min}) + w_{v-1} \Phi_{\min} \quad (14)$$

Using (12) to substitute for s in (14), we have:

$$\begin{aligned} Sent_i(\tau_i, \tau_i^{(k_0+s, v)}) &\geq \frac{w_i r}{W} (\tau_i^{(k_0+s, v)} - \tau_i) \\ &- \frac{w_i \Phi_{\min}}{W} [\sum_{j \in G_{FIN}} w_j + \sum_{\substack{j \in (n-G_{FIN}) \\ j \neq i}} w_i + w_{v-1}] + w_{v-1} \Phi_{\min} \\ &= \frac{w_i r}{W} (\tau_i^{(k_0+s, v)} - \tau_i) - \frac{w_i \Phi_{\min}}{W} \\ & [\sum_{j \in G_{FIN}} w_j + \sum_{\substack{j \in (n-G_{FIN}) \\ j \neq i}} w_i] - \frac{w_i \Phi_{\min}}{W} w_{v-1} + w_{v-1} \Phi_{\min} \\ &= \frac{w_i r}{W} (\tau_i^{(k_0+s, v)} - \tau_i) - \frac{w_i \Phi_{\min}}{W} \\ & [\sum_{j \in G_{FIN}} w_j + \sum_{\substack{j \in (n-G_{FIN}) \\ j \neq i}} w_i] + \frac{(W - w_i) \Phi_{\min}}{W} w_{v-1} \end{aligned} \quad (15)$$

In (15), if $v = 1$ then w_i is the least among all weights and. $w_{v-1} = 0$; if $v > 1$ then $w_{v-1} \geq 1$. Thereupon, introducing a parameter Ψ and its values as follows: $\Psi = 0$ when w_i is the least among n weights; $\Psi = 1$ in other cases. So:

$$\begin{aligned} Senti(\tau_i, \tau_i^{(k_0+s, v)}) &\geq \frac{w_i r}{W} (\tau_i^{(k_0+s, v)} - \tau_i) \\ &- \frac{w_i \Phi_{\min}}{W} [\sum_{j \in G_{FIN}} w_j + \sum_{\substack{j \in (n-G_{FIN}) \\ j \neq i}} w_i] + \Psi \frac{(W - w_i) \Phi_{\min}}{W} \end{aligned} \quad (16)$$

Now, since the sum of the scheduling rates is no more than the link rate r and combing (1), we have:

$$\rho_i \leq \frac{w_i r}{W} \tag{17}$$

Using Eq. (17) in (16), we get:

$$\begin{aligned} Sent_i(\tau_i, \tau_i^{(k_0+s,v)}) &\geq \rho_i(\tau_i^{(k_0+s,v)} - \tau_i) \\ &- \frac{\rho_i \Phi_{\min}}{r} \left[\sum_{j \in G_{FIN}} w_j + \sum_{\substack{j \in (n-G_{FIN}) \\ j \neq i}} w_j \right] + \Psi \left(1 - \frac{\rho_i}{r}\right) \Phi_{\min} \\ &= \rho_i(\tau_i^{(k_0+s,v)} - \tau_i) - \frac{\Phi_{\min}}{r} \\ &\left[\sum_{j \in G_{FIN}} w_j + \sum_{\substack{j \in (n-G_{FIN}) \\ j \neq i}} w_j \right] + \Psi \left(\frac{1}{\rho_i} - \frac{1}{r}\right) \Phi_{\min} \\ &= \rho_i(\tau_i^{(k_0+s,v)} - \tau_i) - \\ &\left(\frac{1}{r} \sum_{j \in G_{FIN}} w_j \Phi_{\min} + \frac{1}{r} \sum_{\substack{j \in (n-G_{FIN}) \\ j \neq i}} w_j \Phi_{\min} - \Psi \left(\frac{1}{\rho_i} - \frac{1}{r}\right) \Phi_{\min}\right) \end{aligned}$$

And then:

$$Sent_i(\tau_i, \tau_i^{(k_0+s,v)}) \geq \max \{0, \rho_i(\tau_i^{(k_0+s,v)} - \tau_i) - \left(\frac{1}{r} \sum_{j \in G_{FIN}} w_j \Phi_{\min} + \frac{1}{r} \sum_{\substack{j \in (n-G_{FIN}) \\ j \neq i}} w_j \Phi_{\min} - \Psi \left(\frac{1}{\rho_i} - \frac{1}{r}\right) \Phi_{\min}\right)\} \tag{18}$$

Therefore:

$$\theta_i' \leq \frac{1}{r} \sum_{j \in G_{FIN}} w_j \Phi_{\min} + \frac{1}{r} \sum_{\substack{j \in (n-G_{FIN}) \\ j \neq i}} w_j \Phi_{\min} - \Psi \left(\frac{1}{\rho_i} - \frac{1}{r}\right) \Phi_{\min}$$

The statement of the theorem is proved.

Now, analyzing the latency of SMRR under two boundary conditions:

Case I: for $\forall j, 1 \leq j \leq n, j \neq i, \rho_i \ll \rho_j$

In this case, $w_i = 1$ and $w_i \ll w_j$. Also $W \gg w_i$, the set G_{FIN} is empty and $\Psi = 0$. Hence:

$$\theta_i \leq \frac{1}{r} \sum_{\substack{j \in (n-G_{FIN}) \\ j \neq i}} w_j \Phi_{\min} = \frac{1}{r} (n-1) \Phi_{\min} \ll \frac{1}{r} (W - w_i) \Phi_{\min} \tag{19}$$

Case II: for $\forall j, 1 \leq j \leq n, j \neq i, \rho_i \gg \rho_j$

In this case, $w_i \gg w_j$. $\Psi = 1$ and the set G_{FIN} includes all $(n-1)$ flows except flow i . Hence:

$$\begin{aligned} \theta_i &\leq \frac{1}{r} \sum_{j \in G_{FIN}} w_j \Phi_{\min} - \left(\frac{1}{\rho_i} - \frac{1}{r}\right) \Phi_{\min} \\ &= \frac{1}{r} (W - w_i) \Phi_{\min} - \left(\frac{1}{\rho_i} - \frac{1}{r}\right) \Phi_{\min} \end{aligned}$$

Since $\rho_i < r$, therefore:

$$\theta_i < \frac{1}{r} (W - w_i) \Phi_{\min} \tag{20}$$

THE LATENCY BOUND OF WRR

Theorem 2: The WRR scheduler belongs to the class of LR servers, with an upper bound on the latency θ_i for flow i given by:

$$\theta_i \leq \frac{1}{r} (W - w_i) \Phi_{\min} \tag{21}$$

Proof: According to Lemma 1, the latency of an LR server can be estimated based on its behavior in the flow active periods, so we will prove the theorem by showing that:

$$\theta_i' \leq \frac{1}{r} (W - w_i) \Phi_{\min}$$

Assume that flow i becomes active at time instant τ_i and let τ_i^s be the start time of the s -th service opportunity of flow i . According to Lemma 3, in order to determine upper bound on the latency of WRR, we need to only consider time interval (τ_i, τ_i^s) for all s . Figure 3 illustrates the time interval under consideration for a given s . 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.

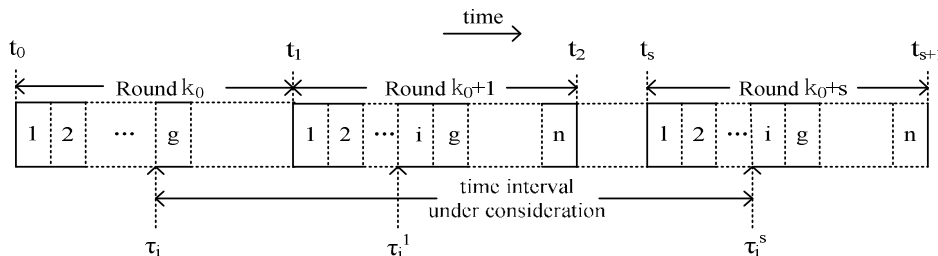


Fig. 3: WRR time interval under consideration

As shown in Fig. 3, assume that the time instant when flow i becomes active coincides with the time instant when some flow g is about to start its service opportunity during the k_0 -th round. Let G_{i+} denote the set of flows which receive service after flow i becomes active, i.e., during the time interval (τ_i, t_1) . Similarly, let G_{i-} denote the set of flows which are served by the WRR scheduler before flow i becomes active, i.e., during the time interval (t_0, τ_i) . 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 the start of the $(k_0 + 1)$ -th round, then the set G_{i+} will be empty and all the $n-1$ flows will be included in the set G_{i-} . 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 time interval (τ_i, τ_i^s) can be split into the following three sub-intervals:

- **(τ_i, \mathbf{t}_1) :** This sub-interval includes the part of the k_0 -th round during which all the flows belonging to the set G_{i+} will be served by the WRR scheduler and then:

$$t_1 - \tau_i = \frac{1}{r} \sum_{j \in G_{i+}} w_j \Phi_{\min} \quad (22)$$

- **$(\mathbf{t}_1, \mathbf{t}_s)$:** This sub-interval includes $s-1$ rounds of the WRR scheduler starting from round $(k_0 + 1)$. Assume that round $(k_0 + h)$ is in progress during the time interval (t_h, t_{h+1}) , we get:

$$t_{h+1} - t_h = \frac{W}{r} \Phi_{\min} \quad (23)$$

Summing Eq. (23) over $s-1$ rounds beginning with round k_0+1 :

$$t_s - t_1 = \frac{W}{r} (s-1) \Phi_{\min} \quad (24)$$

- **(\mathbf{t}_s, τ_i^s) :** This sub-interval includes the part of the $(k_0 + s)$ -th round during which all the flows belonging to the set G_{i-} will be served by the WRR scheduler and then:

$$\tau_i^s - t_s = \frac{1}{r} \sum_{j \in G_{i-}} w_j \Phi_{\min} \quad (25)$$

Combining Eq. (22), (24) and (25), we have:

$$\tau_i^s - \tau_i = \frac{W}{r} (s-1) \Phi_{\min} + \frac{1}{r} \sum_{j \in G_{i+}} w_j \Phi_{\min} + \frac{1}{r} \sum_{j \in G_{i-}} w_j \Phi_{\min}$$

$$= \frac{W}{r} (s-1) \Phi_{\min} + \frac{1}{r} (W - w_i) \Phi_{\min} \quad (26)$$

Solving for $s-1$:

$$s-1 = \frac{r}{W \Phi_{\min}} (\tau_i^s - \tau_i) - \frac{W - w_i}{W} \quad (27)$$

Note that during the time interval under consideration, (τ_i, τ_i^s) , flow i receives service in $s-1$ rounds starting at round $(k_0 + 1)$. We get:

$$Sent_i(\tau_i, \tau_i^s) = w_i (s-1) \Phi_{\min} \quad (28)$$

Using (27) to substitute for $(s-1)$ in (28), we have:

$$Sent_i(\tau_i, \tau_i^s) = \frac{w_i r}{W} (\tau_i^s - \tau_i) - \frac{w_i (W - w_i)}{W} \Phi_{\min} \quad (29)$$

Since $\rho_i \leq \frac{w_i r}{W}$, hence:

$$\begin{aligned} Sent_i(\tau_i, \tau_i^s) &\geq \rho_i (\tau_i^s - \tau_i) - \frac{\rho_i}{r} (W - w_i) \Phi_{\min} \\ &= \rho_i (\tau_i^s - \tau_i - \frac{1}{r} (W - w_i) \Phi_{\min}) \end{aligned} \quad (30)$$

We get:

$$Sent_i(\tau_i, \tau_i^s) \geq \max \{0, \rho_i (\tau_i^s - \tau_i - \frac{1}{r} (W - w_i) \Phi_{\min})\} \quad (31)$$

And then:

$$\theta_i' \leq \frac{1}{r} (W - w_i) \Phi_{\min}$$

The theorem is proved.

COMPARATIVE ANALYSIS OF LATENCY UPPER BOUND

At first, we present a comparison of the latency upper bound of WRR derived in Theorem 2 with the latency bound derived in (Stiliadis and Verma, 1998). Let Θ_i^{new} represent the latency bound derived in Theorem 2. Thus:

$$\theta_i^{new} = \frac{1}{r} (W - w_i) \Phi_{\min} \quad (32)$$

Let Θ_i^{old} represent the latency bound of WRR as derived in (Stiliadis and Verma, 1998), namely:

Table 1: Comparison of latency bounds

Scheduler	Latency bound
θ_i^{old}	$\frac{1}{r}(W - w_i)\Phi_{\min} + \frac{L_c}{r}$
θ_i^{new}	$\frac{1}{r}(W - w_i)\Phi_{\min}$
θ_i^{smrr}	$\frac{1}{r} \sum_{j \in \mathcal{G}_{\text{WRR}}} w_j \Phi_{\min} + \frac{1}{r} \sum_{\substack{j \in (\mathcal{N} - \mathcal{G}_{\text{WRR}}) \\ j \neq i}} w_j \Phi_{\min} - \Psi \left(\frac{1}{\rho_i} - \frac{1}{r} \right) \Phi_{\min}$
$\theta_i^{smrr-low}$	$\frac{1}{r} \sum_{j \in \mathcal{G}_{\text{WRR}}} w_j \Phi_{\min}$
$\theta_i^{smrr-high}$	$\frac{1}{r}(W - w_i)\Phi_{\min} - \left(\frac{1}{\rho_i} - \frac{1}{r} \right) \Phi_{\min}$

$$\theta_i^{old} = \frac{F - \Phi_i + L_c}{r} \quad (33)$$

In Eq. (33), F denotes the size of a WRR frame which is equal to $W\Phi_{\min}$, the summation of the data permitted to be sent by all the active flows. L_c is the size of an ATM cell. Substituting for F in Eq. (33) we get:

$$\theta_i^{old} = \frac{W\Phi_{\min} - \Phi_i + L_c}{r} = \frac{1}{r}(W - w_i)\Phi_{\min} + \frac{L_c}{r} \quad (34)$$

Comparing Eq. (32) and (34), we have $\theta_i^{new} < \theta_i^{old}$. It illuminates that the upper bound on the latency of WRR derived in Theorem 2 is lower.

Next, we compare the latency bound of SMRR analyzed earlier with the latency bound of WRR. The latency of SMRR, under two boundary conditions, are shown in Eq. (19) and (20). Both of them are less than $\frac{1}{r}(W - w_i)\Phi_{\min}$ obviously, so the latency characteristic of SMRR is better than that of WRR. Note that there is a special case: SMRR and WRR have the same latency upper bound when all the n flows have the same weight. Table 1 summarizes each latency upper bound and also making a comparison to these latency bounds through Table 1. Hereinto, θ_i^{smrr} represents the latency upper bound of SMRR; $\theta_i^{smrr-low}$ and $\theta_i^{smrr-high}$ represent the latency upper bounds of SMRR under boundary conditions I and II respectively.

FAIRNESS

For fairness analysis, the fairness metric proposed in Golestani (1994) is more commonly employed. This metric, known as the Relative Fairness Bound (RFB), is defined as the maximum difference in the service received by any two flows over all possible intervals of time.

Definition 5: For any two flows i and j of all the flows which have continuous packets queuing during the time interval (t_1, t_2) , let $FM(t_1, t_2)$ represent the maximum value of the following formula (35). FM is defined as

the maximum value of $FM(t_1, t_2)$ over all possible time intervals (t_1, t_2) :

$$\left| \frac{Sent_i(t_1, t_2)}{\rho_i} - \frac{Sent_j(t_1, t_2)}{\rho_j} \right| \quad (35)$$

If a scheduling algorithm has $FM = 0$, it is obvious that this algorithm has good relative fairness. For example, GPS (Generalized Processor Sharing) (Parekh and Gallager, 1992) is proven to possess this property. However, this condition cannot be met by any packet-by-packet algorithm since packet must be served exclusively. In such algorithms, we can only require that their FM s are bounded by a constant. Therefore, for a scheduling algorithm running on server S , it is considered close to fair if its FM is a constant. In particular, $FM(t_1, t_2)$ should not depend on the size of the time interval. This constant is called F^S in (Stiliadis and Verma, 1998) and F^S is called the fairness of server S . Apparently, $FM \leq F^S$.

Theorem 3: The SMRR schedulers have $FM \leq \frac{\Phi_{\min}}{\rho_{\min}}$.

Proof: Consider any two flows i and j that are continuously backlogged in a time interval (t_1, t_2) . Since the maximal difference of the service opportunity between i and j is one secondary round, without loss of generality, assume that two flows have passed through k primary rounds. In addition, flows i and j have passed through $v-1$ and v secondary rounds respectively. We have:

$$Sent_i(t_1, t_2) = w_i(k\Phi_{\min}) + w_{v-1}\Phi_{\min}$$

$$Sent_j(t_1, t_2) = w_j(k\Phi_{\min}) + w_v\Phi_{\min}$$

And then:

$$\left| \frac{Sent_i(t_1, t_2)}{\rho_i} - \frac{Sent_j(t_1, t_2)}{\rho_j} \right| = \left| \frac{w_i(k\Phi_{\min})}{\rho_i} - \frac{w_j(k\Phi_{\min})}{\rho_j} + \frac{w_{v-1}\Phi_{\min}}{\rho_i} - \frac{w_v\Phi_{\min}}{\rho_j} \right|$$

According to (1), we have $\frac{w_i}{\rho_i} = \frac{w_j}{\rho_j}$, therefore:

$$\left| \frac{Sent_i(t_1, t_2)}{\rho_i} - \frac{Sent_j(t_1, t_2)}{\rho_j} \right| = \left| \frac{w_{v-1}\Phi_{\min}}{\rho_i} - \frac{w_v\Phi_{\min}}{\rho_j} \right| \quad (36)$$

The following case discussion:

When $v = 1$:

$$\left| \frac{Sent_i(t_1, t_2)}{\rho_i} - \frac{Sent_j(t_1, t_2)}{\rho_j} \right| = \frac{w_v \Phi_{min}}{\rho_j} \leq \frac{w_j \Phi_{min}}{\rho_j} = \frac{\Phi_{min}}{\rho_{min}} \quad (37)$$

When $v > 1$, if:

$$\frac{w_{v-1} \Phi_{min}}{\rho_i} \geq \frac{w_v \Phi_{min}}{\rho_j}$$

then:

$$\left| \frac{Sent_i(t_1, t_2)}{\rho_i} - \frac{Sent_j(t_1, t_2)}{\rho_j} \right| = \frac{w_{v-1} \Phi_{min}}{\rho_i} - \frac{w_v \Phi_{min}}{\rho_j} \leq \frac{w_{v-1} \Phi_{min}}{\rho_{v-1}} - \frac{w_v \Phi_{min}}{\rho_j} < \frac{\Phi_{min}}{\rho_{min}} \quad (38)$$

If:

$$\frac{w_{v-1} \Phi_{min}}{\rho_i} \leq \frac{w_v \Phi_{min}}{\rho_j}$$

then:

$$\left| \frac{Sent_i(t_1, t_2)}{\rho_i} - \frac{Sent_j(t_1, t_2)}{\rho_j} \right| = \frac{w_v \Phi_{min}}{\rho_j} - \frac{w_{v-1} \Phi_{min}}{\rho_i} \leq \frac{w_j \Phi_{min}}{\rho_j} - \frac{w_{v-1} \Phi_{min}}{\rho_i} < \frac{\Phi_{min}}{\rho_{min}} \quad (39)$$

Combining (37), (38) and (39), we get $FM \leq \frac{\Phi_{min}}{\rho_{min}}$.

The theorem is proved. Next, analyzing the fairness of WRR:

Theorem 4: The WRR schedulers have $FM \leq \frac{\Phi_{min}}{\rho_{min}}$.

Proof: Consider any two flows i and j that are continuously backlogged in a time interval (t_1, t_2) . Let k be the number of service opportunities given to flow i in interval (t_1, t_2) and let k' be the number of service opportunities given to flow j in the same interval, then $|k - k'| \leq 1$, namely $k \geq k' - 1$ and $k \leq k' + 1$. Thus,

When,

$$\frac{k\Phi_i}{\rho_i} \geq \frac{k'\Phi_j}{\rho_j}$$

we have:

$$\left| \frac{Sent_i(t_1, t_2)}{\rho_i} - \frac{Sent_j(t_1, t_2)}{\rho_j} \right| = \frac{k\Phi_i}{\rho_i} - \frac{k'\Phi_j}{\rho_j} \leq \frac{k\Phi_i}{\rho_i} - \frac{(k-1)\Phi_j}{\rho_j} = \frac{\Phi_j}{\rho_j} = \frac{\Phi_{min}}{\rho_{min}} \quad (40)$$

When,

$$\frac{k\Phi_i}{\rho_i} \leq \frac{k'\Phi_j}{\rho_j}$$

we have:

$$\left| \frac{Sent_i(t_1, t_2)}{\rho_i} - \frac{Sent_j(t_1, t_2)}{\rho_j} \right| = \frac{k\Phi_i}{\rho_i} - \frac{k'\Phi_j}{\rho_j} \leq \frac{k'\Phi_j}{\rho_j} - \frac{(k'-1)\Phi_i}{\rho_i} = \frac{\Phi_i}{\rho_i} = \frac{\Phi_{min}}{\rho_{min}} \quad (41)$$

Combining (40) and (41), we get $FM \leq \frac{\Phi_{min}}{\rho_{min}}$. The theorem is proved.

From Theorem 3 and 4, we can see that both SMRR and WRR schedulers have $FM \leq \frac{\Phi_{min}}{\rho_{min}}$. The reason is that there is a special case mentioned above, namely all the n flows have the same weights, at which both SMRR and WRR schedulers have the same FM. Nevertheless, when flows have different weights, SMRR scheduler has better fairness than WRR, which can be derived by comparing (37)-(39) with (40)-(41). For example, in (38), we have:

$$\frac{w_{v-1} \Phi_{min}}{\rho_{v-1}} - \frac{w_v \Phi_{min}}{\rho_j} = \frac{\Phi_{min}}{\rho_{min}} - \frac{w_v \Phi_{min}}{w_j \rho_{min}} = \left(1 - \frac{w_v}{w_j}\right) \frac{\Phi_{min}}{\rho_{min}} \quad (42)$$

The result of (42) is less than $\frac{\Phi_{min}}{\rho_{min}}$ absolutely.

IMPLEMENTATION COMPLEXITY

Implementation complexity is also an important index to evaluate a scheduling algorithm besides latency and fairness. Since a scheduling algorithm is wanted to be implemented in high-speed networks generally, implementation complexity of a scheduling algorithm will affect its practical application. In addition, it is better that the implementation complexity does not depend on the number of the active flows.

When a SMRR scheduler serves n flows, its implementation complexity is defined as the order of the time complexity, with respect to n , of enqueueing and then dequeuing a packet for transmission.

Theorem 5: The implementation complexity of a SMRR scheduler is $O(1)$.

Proof: The theorem will be proved if we can prove that enqueueing and dequeueing a packet are each of time complexity $O(1)$.

When a new packet arrives, enqueueing process of SMRR is executed. The operations include: determining the queue at which the new packet arrives and it is an $O(1)$ operation. If the flow to which the new packet belongs is not already in the Active Flow List, it will be added to the end of the list. This addition is also an $O(1)$ operation.

Now analyzing the time complexity of dequeueing a packet. Since SMRR scheduler transmits at least one packet during each service opportunity given to a flow, the time complexity of dequeueing a packet is equal to or less than the time complexity of all the operations performed in enqueueing process of SMRR during each service opportunity. These operations involve: determining the next flow to be served; removing a flow from the head of the Surplus Flow List and possibly adding a flow back at the tail. All of these operations on a list can be finished in $O(1)$ time. In addition, each service opportunity involves updating the values of Visit Flow Count, Packet Number Count and Surplus Weight. All of these operations can be executed in constant time, therefore the time complexity of dequeueing a packet is $O(1)$. The theorem is proved.

CONCLUSION

In this study, we have presented an improved version of weighted round-robin called Successive Minimal-weight Round Robin (SMRR). A primary round consists of one or more secondary rounds and in every round each flow always gains the equivalent service opportunities to the least weight of the current round. Firstly, we prove the relevant theorems to determine the upper bound on latency of SMRR and WRR; hereinto the latency upper bound of WRR is smaller than that derived in other literature. Then, we discuss the fairness of SMRR and WRR. As a result of comparative analysis, it shows that SMRR almost always has better latency property and fairness than WRR. We also prove that the implementation complexity of SMRR is $O(1)$. Therefore, it can be easily implemented in high-speed networks with large number of flows and carry out fair and efficient packet scheduling.

ACKNOWLEDGMENT

This study is supported by the S&T Plan Projects of Hunan Province (No. 2010GK3045), Scientific Research Fund of Hunan Provincial Education Department (No. 10C0687).

REFERENCES

- Bennett, J.C.R. and H. Zhang, 1996. WF2Q: Worst-case fair weighted fair queueing. Proceedings of IEEE 15th Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation (INFOCOM '96). San Francisco, CA, pp: 120-128.
- Cao, C., Y. Zhang and R. Chen, 2010. UWRR: A novel scheduling algorithm for self-similar multi-services in optical packet switching network. Proceedings of the 2010 Symposium on Photonics and Optoelectronic (SOPO 2010). Chengdu, China, pp: 1-4.
- Demers, A., S. Keshav and S. Shenker, 1998. Design and analysis of a fair queueing algorithm. Proceedings of the ACM SIGCOMM. Austin, pp: 1-12.
- Golestani, S.J., 1994. A self-clocked fair queueing scheme for broadband applications. Proceedings of IEEE INFOCOM'94. Toronto, pp: 636-646.
- Goyal, P., H.M. Vin and H. Cheng, 1997. Start-time fair queueing: A scheduling algorithm for integrated services packet switched networks. IEEE T. Network., 5(5): 690-704.
- Guo, C., 2001. SRR: An $O(1)$ time complexity packet scheduler for flows in multi-service packet networks. Proceedings of the 2001 SIGCOMM. San Diego, CA, pp: 211-222.
- He, M., D. Zhao and L. Bao, 2012. An energy-efficiency polling access control protocol for wireless sensor networks. CAAI T. Intell. Syst., 7(3): 265-270, (Chinese).
- Hemant, M. and U. Madhow, 2003. Fair scheduling with tunable latency: A round robin approach. IEEE/ACM T. Network., 11(4): 592-601.
- Hu, H., Y. Peng and Y. Guo, 2008. A fair service and dynamic round robin scheduling algorithm. J. Software, 19(7): 1856-1864, (Chinese).
- Lin, D., Y. Jiang and M. Hamdi, 2011. Selective-request round-robin scheduling for VOQ packet switch architecture. Proceedings of the IEEE International Conference on Communications (ICC), pp: 1-5.
- Pan, C. and J. Zhang, 2012. An improved queue scheduling algorithm based on WRR. Telecommun. Inform., 1: 34-37, (Chinese).
- Parekh, A.K. and R.G. Gallager, 1992. A generalized processor sharing approach to flow control the single node case. Proceedings of INFOCOM'92. Florence, pp: 915-924.
- Shreedhar, M. and G. Varghese, 1996. Efficient fair queueing using deficit round-robin. IEEE T. Network., 4(3): 375-385.
- Stiliadis, D. and A. Verma, 1998. Latency-rate servers: A general model for analysis of traffic scheduling algorithms. IEEE-ACM T. Network., 6(5): 611-624.

- Xiong, L. and S. Zhang, 2012. The optimization of the WRR algorithm in multi-class real-time data scheduling. *Comput. Eng. Sci.*, 34(7): 35-38, (Chinese).
- Zhang, L., 1990. Virtual clock: A new traffic control algorithm for packet switching networks. *Proceedings of the ACM Symposium on Communications Architectures and Protocols (SIGCOMM '90)*. Philadelphia, pp: 19-29.
- Zhang, X., G. Ni and F. Jin, 2011. A new WRR algorithm based on ideal packet interval time. *Proceedings of the International Conference on Intelligent Computation Technology and Automation (ICICTA)*. Shenzhen, China, pp: 1039-1042.