A Scheduling Algorithm for Controlling of Service Rate and Burst

Mahmoud Daneshvar Farzanegan[1],Hossein Saidi[2], and Mehdi Mahdavi[3]

Department of Electrical Engineering
Najafabad Branch, Islamic Azad University
Isfahan, Iran

[1]*m_daneshvar@pel.iaun.ac.ir , m_daneshvar@ec.iut.ac.ir,*
[2]*hsaidi@cc.iut.ac.ir,* [3]*m_mahdavi@cc.iut.ac.ir*

***Abstract*—Scheduling algorithms have significant role in providing of the quality of service (QoS) in data networks. Mostly, service rate is considered as an isolating parameter in the scheduling algorithms. However, when a delay-sensitive and bursty session such as video streaming or interactive gaming is scheduled by these rate-based servers, the imposed delay may be much more than an acceptable threshold which is required for QoS provisioning. In this paper, we propose a *fluid flow* scheduling algorithm which applies *burstiness* as an additional isolating parameter. We assume that the arrival burstiness constraint is *leaky bucket*. Our proposed scheduling algorithm attempts to provide a service discipline similar to the arrival constraint. In our algorithm, the *weight* of each session and also service rate may increase when a burst arrives. Therefore, the scheduler can isolate some bursty sessions to receive much more amount of service in burst condition. The evaluation of the proposed algorithm is carried out by calculating packet delay statistics through a simulation strategy where various kinds of traffic are scheduled by the proposed algorithm.**

Index Terms—Scheduling, Quality of Service, Fairness, Bursty Traffic, GPS.

I. INTRODUCTION

Multimedia applications such as IPTV, gaming, Voice Over IP (VOIP) and Video On Demand (VOD) are going to be tremendously widespread among Internet and Intranet users. These applications can provide a wide range of services simultaneously. Some applications such as VOIP, generate a constant bit rate (CBR) flow of traffic while others such as compressed video transmission and web browser produce a stream of traffic with variable bit rate (VBR). In addition, real-time applications (e.g. VoIP and IPTV) are delay-sensitive while other non-real time services (e.g. FTP and Email) do not need any guaranteed upper bound of the end-to-end delay. Therefore, service providing for real-time variable bit rate (rt-VBR) sessions which require a bounded delay is more challenging in comparison with other applications.

Different applications have different traffic natures which imply different QoS requirements. QoS is usually quantified by a single or combination of measurable parameters such as end-to-end delay, bandwidth, delay jitter (variation in delays) and packet loss. To provide guaranteed QoS in a network, not only the cooperation among different layers inside a network device is required, but also these appliances should collaborate with each other in the network in order to appropriately assign their available resources to the demanding sessions. .In network infrastructure, routers and data switches are the most important devices in QoS provisioning. One of the components of a switch/router that plays a significant role in providing QoS is the scheduling module. A scheduling algorithm is a method that chooses the most appropriate packet among competing packets for transferring to a specified output port. A QoS scheduling algorithm is the one that considers QoS parameters in its decision making process. Thus, coincident with the emergence of multimedia applications in recent years, many researches have been conducted to achieve high performance QoS scheduling algorithms on multimedia applications [1]-[6].

A well-known class of scheduling algorithms is *Rate Proportional Server* (RPS) which isolates different sessions from each other based on their service rates [7]. An ideal scheduling algorithm in this class is *Generalized Processor Sharing* (GPS) [8] which is used as a benchmark in evaluation of practical algorithms. Some other types of packet-by-packet scheduling disciplines such as SCFQ [9] and FFQ [10], try to approximate GPS in real life.

When all the incoming traffic to the scheduler is constant bit rate, an RPS server, say GPS, ideally share bandwidth among them. But when an RPS server needs to schedule a mixture of VBR and CBR flows, the provided QoS for rt-VBR (bursty and delay-sensitive) sessions may be unacceptable. The reason is that the packet delay in a VBR flow may be much more than the packet delay in a CBR flow. We show this fact by an example in section II.

The *Smoothing* or *traffic shaping* methods are used for controlling of the burstiness of the VBR flows. The *Leaky Bucket* and *Token Bucket* techniques are two well-known traffic smoothing approaches [11] which are developed to reduce the burstiness value of bursty flows. However, decreasing the burstiness in VBR traffic makes each packet to experience even more delay in the buffer of the traffic shaper

[12] and thus, QoS guaranteeing for VBR sessions still remains a challenging issue.

In this paper, we propose a scheduling algorithm that provides a more desirable service to rt-VBR sessions by considering predefined parameters for each session. These parameters are the *required service rate*, r_i, and a pair of thresholds which indicate the amount of extra service for each bursty session. The thresholds are MBT_i and BSP_i which are defined in section III through definitions 2 and 4. Using these thresholds, a specific session may receive much more amount of service than its share in a short time when a burst arrives and so we call it *burst service*. Our algorithm which is termed by Regulated Burst Service Scheduling (RBSS) is established on a fluid-flow model, where we assume that all packets are infinitely divisible.

In RBSS, associated to each session, a parameter named *instantaneous weight* is assigned which directly affects the rate of service at each time. Unlike GPS, the assigned weights to sessions are not fixed and can be changed depending on queue length variation.

In this paper we aim to create a baseline for evaluation of scheduling algorithms that try to handle bursty traffic. To this end, we focus on GPS. Recently, some studies also have been carried out on GPS with respect to bursty traffic such as [13]-[15].We believe that the devised idea in our proposed algorithm can be applied in other implantable packet-by-packet scheduling algorithms in order to make them capable of providing appropriate service to rt-VBR flows.

The organization of the paper is as follows. Section II describes the related works. Section III presents our proposed algorithm, RBSS. The performance of RBSS in a single node is simulated in section IV, and then a comparison with GPS is performed. Finally, section V concludes the paper.

II. RELATED WORKS

Scheduling algorithms for VBR traffic have been an active field with rich background in research literatures. The most common approach in this context is that other parameters rather than service rate (e.g. delay or jitter) are engaged in design of the algorithm in order to differentiate among different sessions with different requirements [16]-[27]. We briefly review some of these scheduling algorithms as follows.

When a session with bursty traffic has no packet to send, the session becomes absent for a while, so it may lose its service shares. In addition to bursty traffic, this situation can also occur in wireless networks due to channel disappearance [17]-[18]. In some studies, the amount of lost service share is calculated by a credit parameter and the scheduling algorithm tries to compensate the lost share. In [17], Lee et al. have suggested a modified version of Weighted Fair Queuing (WFQ) algorithm. In their algorithm, some part of the lost service share can be compensated by applying credit to each serving session. Therefore, it is necessary to calculate and update the credit values for sessions after serving each packet, this requirement is a challenging issue in practical implementation of the algorithm.

In the other category of algorithms, a due time threshold for each session is defined and the scheduler acts as an *earliest deadline first* (EDF) paradigm. In some papers, the due time threshold have been used for each burst and the scheduler should serve each burst of packets so that a guaranteed feature is achieved [19]-[22].In the algorithm of Sariowan et al [19], the amount of service in each session attempt to be more than a pre-determined curve which is named *Service Curve*. In this strategy, not only service rate but also the burstiness has impact on the scheduling rules. Considering the expansion of the networks and number of concurrent sessions, it would be difficult to implement these algorithms.

In [20], Golestani specifies an average service rate *r*, and a window time *T*, for each flow or session. Hence, the amount of service provided by the scheduler during this window time should not exceed *r.T*. This method controls the burstiness of output traffic while providing a fair service .Hamadoui et al. also propose another algorithm in [21] which guarantees that at least *m* packets out of *k* consecutive packets (a known value which represents a frame of data), will receive service before a predefined deadline. This method is suitable for scheduling a video traffic which consists of encoded video frames. Both of these methods require many calculations for updating the deadlines which makes the overall scheduler to suffer from a high computational complexity.

In [23], Hanada et al have used Leaky Bucket as a traffic shaper component to smooth the arriving traffic. To reduce the delay bound, they use bucket size, σ, as their controlling parameter along with another bucket for each flow which is named *service token bucket*. Their system is non-work conserving, since if all service token buckets were empty, then they cannot get more than their assigned rate.

In [24], Kwon et al. have proposed a scheduling algorithm which tries to schedule real time variable bit rate (rt-VBR) traffic (e.g. video) and constant bit rate (CBR) traffic (e.g. audio) while observing quality of service criteria. The increase of a session's weight in this algorithm is performed by considering the queue length values and comparing it with a single threshold; therefore the busy state of the session is determined.

There are other methods which use the delay or jitter (i.e. delay variation) parameter as their main scheduling parameter [26]-[27], due to jitter importance in multimedia applications. In this group of schedulers, bursty traffic is treated implicitly. These scheduling algorithms are designed for guaranteeing per-hob, per-class relative service where there are few flows.

On the other hand, many approaches have been carried out to obtain a real VBR traffic model [28]-[35]. *Burstiness* as a single parameter which describes bursty behavior is one of the

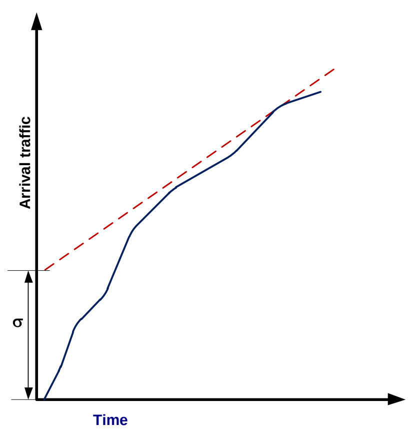


FIGURE 1- LEAKY BUCKET (A BURSTINESS CONSTRAINT)

most famous parameter in this regard. It is notable that in the literature the concept of burstiness has been differently defined such as: *The ratio of the maximum to the average data rate*[28], *the Indexes of Dispersion*, the *Peakedness*[29], the *leaky bucket size* (σ) [30], *the ratio of ON-time average value to the sum of ON-time and OFF-time average values*[32] and the *Hurst* parameter in the *self-similarity* model [33].

In this paper, the minimum value of the *leaky bucket size* (σ) that satisfies *Definition1* [31] is considered as the burstiness parameter (See Figure 1).

Definition1: Let A_i (t_1, t_2), be the amount of data which is produced by the session f_i in an arbitrary interval [t_1, t_2). The session f_i is (σ_i, ρ_i) *Leaky Bucket* shaped if the following property is conformed for any t_1 and t_2 (which $t_1 \le t_2$):

$$A_i(t_1,t_2) \le \rho.(t_2\text{-}t_1) + \sigma \qquad (1)$$

In [30], it is shown that if a leaky bucket-shaped traffic is served by an ideal server with service rate ρ, the maximum length of the backlogged traffic in the queue is not greater than the bucket size.

In this paper, we use bucket size σ, as the burstiness parameter because not only this parameter has been widely used in network flow control techniques, but also it is feasible to measure or approximate the burstiness of traffic by leaky bucket definition by means of measuring the queue length. As an example we refer to a real-time algorithm by Kodama et al. in [36] where the burstiness is measured by calculating the virtual queue lengths. It is notable that leaky bucket is a special case of *Burstiness Constraint* curve or *Arrival* curve in the Network Calculus theory [31], and we can use it to calculate upper bounds of delay or queue length in the network.

Compared with the reviewed algorithms, our proposed algorithm is a general and fluid flow based algorithm which uses traditional leaky bucket shaper. Our algorithm attempts to provide a service discipline that is similar to the arrival curve or leaky bucket constraint. For implementing the proposed algorithm, only queue length is required to be measured.

III. Proposed Algorithm

In this section, we firstly define some required parameters and then a new scheduling algorithm called *Regulated Burst Service Scheduling* (RBSS) will be introduced. Then, we compare GPS with the proposed algorithm in terms of service delay. This comparison between GPS and RBSS is due to the fact that both of these algorithms are based on the fluid flow model and their structures are similar. This comparison shows that for a session with a high burstiness value, RBSS imposes less delay in comparison to GPS.

A. Definitions

Consider a set of flows or sessions, f_i where $i \in \{1..N\}$, that share an output link with the bandwidth (rate) of R bps.

The following terminologies are generally used and also we use them throughout the paper:

Backlogged session: At any time, a *backlogged* session is a session that its queue is not empty at that time. At any time t, BF(t) denotes the set of backlogged sessions.

Busy period: A *busy period* is the maximal-length interval of a period of time such that the output link is not idle during that period. Assume the system is studied during one busy period. Without loss of generality we may assume that busy period starts at time 0.

Queue Length: At any time t, *Queue length* of an active session f_i is denoted by $Q_i(t)$ and is defined as the total amount of data that has not been transmitted yet. For session f_i, let $A_i(t)$ and $S_i(t)$ be the amount of data that is received and served until time t respectively. Therefore we can write: $Q_i(t) = A_i(t) - S_i(t)$.

We specifically use the following definitions in RBSS:

Definition 2- Minimum Burst Threshold (MBT): For each session, a threshold is defined to indicate the occurrence of a burst condition, i.e. whenever the queue length of the session exceeds this threshold; it is assumed that the condition is bursty. The value of MBT could be identically defined for every session.

Definition 3- Burst Degree (BD): At any time t, the queue length $Q_i(t)$ normalized by MBT_i is defined as the Burst Degree based on MBT_i. Larger value of BD_i means more noticeable burst circumstances while $BD_i < 1$ indicates that there is not a bursty condition.

$$BD_i(t) = \frac{Q_i(t)}{MBT_i} \qquad (2)$$

Definition4- Burst Service Parameter (BSP): BSP is the maximum permissible ratio of the instant weight, $w_i(t)$, in accordance with constant weight, ω_i. This ratio would exceed due to burst condition.

For any session f_i which is shaped by a (σ_i, ρ_i)*-leaky bucket* shaper, the BSP_i value is selected as follows:

$$BSP_i = \left\lceil \frac{\sigma_i}{MBT_i} \right\rceil - 1 \qquad (3)$$

where $\lceil . \rceil$ is the ceiling function.

The value of MBT_i is selected according to the burst service priority. Selecting smaller value for MBT_i, makes the scheduler to pay more attention to session f_i.

The value of ω_i.is calculated based on the value r_i as follows: $\omega_i = \frac{r_i}{R}$ (4)

B. Regulated Burst Service Scheduler Algorithm

At each time and for each session f_i, the value of $BD_i(t)$ should be measured and then, the *instantaneous weight* which is denoted by $w_i(t)$, is assigned as follows:

$$w_i(t) = \omega_i.\min\{\lceil BD_i(t) \rceil, BSP_i + 1\} \qquad (5)$$

Figure 2 shows how buffer is scaled by MBT and how the instantaneous weight is obtained from BD_i. When BD_i is between 2 and 3, the instantaneous weight is $3\omega_i$. Thus, the instantaneous weight of session f_i varies from ω_i to $\omega_i.(BSP_i + 1)$ in respect to the queue length.

The instantaneous service rate $r_i^s(t)$, which is associated to a backlogged session f_i at any time t, is computed by (6):

$$r_i^s(t) = \frac{w_i(t)}{W(t)}.R \qquad (6)$$

which: $W(t) = \sum_{f_j \in BF(t)} w_j(t)$

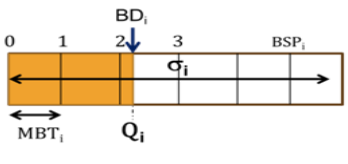


Figure 2- Scaling of the buffer size by MBT in RBSS

The service provisioning in RBSS is in the fluid flow treatment such as GPS. The main difference between the GPS and RBSS is the selection of session's weight. If the instantaneous weight in RBSS does not change both of algorithms do as the same. With $BSP_i=0$ this goal can be met.

RBSS attempts to provide service to a session f_i which is shaped by (σ_i, ρ_i)-leaky bucket shaper in which behavior that arrivals are received. When the arrivals rate instantaneously exceeds, RBSS increases $w_i(t)$ according to increasing the queue length or BD_i(see Eq. (5)). Using this technique, the instantaneous service rate of session f_i exceeds gradually. Due to constraint which is leaky bucket forces on the amount of arrival, it is expected that the arrival rate remains below the r_i in long term. Hence, the queue length would be decreased and consequently, both of $BD_i(t)$ and $r_i^s(t)$ come down.

RBSS algorithm is summarized as follows:

Arrival event:
When the k^{th} packet with length L_i^k arrives to session f_i:

- The queue length is increased:
  $Q_i(t^+) = Q_i(t^-) + L_i^k$
- The Burst Degree is updated: $BD_i(t^+) = \frac{Q_i(t^+)}{MBT_i}$
- The weight of session f_i, $w_i(t)$, is updated according to (11) and the increase of the session's weight, dw_i, is calculated as follows:

$$dw_i = w_i(t^+) - w_i(t^-)$$

- The sum of all weights, W, is updated as follows:

$$W(t^+) = W(t^-) + dw_i$$

Rate assignment:

- The instantaneous service rate in time t is obtained as follows: $r_i(t) = \frac{w_i(t)}{W(t)}.R$

Associated Service:

- If t_1 is the last event time (arrival or departure) and t_2 is any time before the next event, the amount of associated service to the backlogged session f_i between t_1 and t_2 is:
  $S_i(t, t_1) = \frac{w_i(t_1^+)}{W(t_1^+)}.R.(t - t_1)\ t \in (t_1, t_2)$

Departure event:
When a packet with length L_i^k departs from session f_i:

- The queue length is decreased:
  $Q_i(t^+) = Q_i(t^-) - L_i^k$
- The Burst Degree is updated: $BD_i(t^+) = \frac{Q_i(t^+)}{MBT_i}$
- The weight of session f_i, $w_i(t)$, is updated according to (11) and the decrease of the session's weight, dw_i, is calculated as follows:
  $dw_i = w_i(t^+) - w_i(t^-)$.
- The sum of all weights, W, is updated :
  $W(t^+) = W(t^-) + dw_i$

In the rest of the paper, it is assumed that the weight updating performs when a packet completely arrives or departs scheduler.

C. RBSS Compared with GPS

In this part, GPS and RBSS behavior under bursty traffic is exemplified. We consider four sessions with equal average rates, three of which have smooth traffic and the last session is bursty. It is assumed that all packets have the same length and the average arrival rate of all sessions is 0.25 packet/s. Figure 3 and Figure 4 show packet arrival and departure during an observation period of 16 seconds for GPS and RBSS. We further assume that when the last bit of a packet is received (or served), and then the packet is received (or served). Packet arrival times are shown by upward arrows while a downward arrow represents the time when a packet is completely served. The delay of each packet is defined as the difference between its departure and arrival. The arrival and departure arrows of each packet have the same color.

Since GPS discipline is based on a fluid flow model, it is assumed that the scheduler serves all backlogged sessions simultaneously. In this example, session 4 has no packet till the 6^{th} second, when a burst is received in the scheduler. In the GPS scheduler all sessions are served similarly, therefore the second and third packets in session f_4 experience about 3.5second and 5.5second delay due to the presence of other sessions' packets in the queue.

The averages of packet delay in each session in this example are collected in Figure 3. As can be seen, the average delay of f_4 is approximately 2.4 times (4.4 second versus 1.8 second) greater than the average delay of all of the other sessions (namely f_1 to f_3). If session f_4 is delay sensitive, then it is not satisfied its desired QoS.

In order to reduce the packet delay of the bursty session, we may assign a higher priority or constant weight to this session. But, this idea results in unfairness and in receiving more service by the bursty session.

Using the same example, the RBSS scheduler is simulated in C language. Figure 4illustrates the behaviour of the RBSS scheduler.

Flows f_1 to f_3 are leaky bucket shaped by (1.25, 0.25) and flow f_4 is leaky bucket shaped by (3.81, 0.25) these value can be calculated by the arrivals in each sessions. . In this simulation, the *BSP* values for f_1 to f_3 are selected to be zero and the *BSP* value is equal to 3 for f_4. The *MBT* value is chosen equal to 1 for all sessions.

TABLE I-AVERAGE AND VARIANCE OF PACKET DELAY IN THE EXAMPLE

Session(s)	Average of Delay		Variance of Delay	
	Algorithms			
	GPS	RBSS	GPS	RBSS
f_1 to f_3	1.8	2.2	1	2.1
f_1	1.85	2.35	0.963	2.757
f_2	1.975	2.35	1.309	2.51
f_3	1.7	1.975	0.76	1.869
f_4	4.45	3.65	8.67	7.483

Up to the 7^{th} second, the RBSS procedure is the same as that of the GPS, because only one session is backlogged at any time during this period. At this moment, by having an arrival from session 1, the procedures will be different. At this time, the queue length of session f_4 and f_1 are 3 and 1, thus the weight of session f_4 will be 3 times greater than that of f_1. Consequently, between the$7^{th}$and 8^{th} seconds, 75% of the 2^{nd}

packet of session f_4 has been served, but only 25% of the packet from session f_1 is served. At second 8, another packet is received from session f_2 and as a result, the rates of sessions f_1,

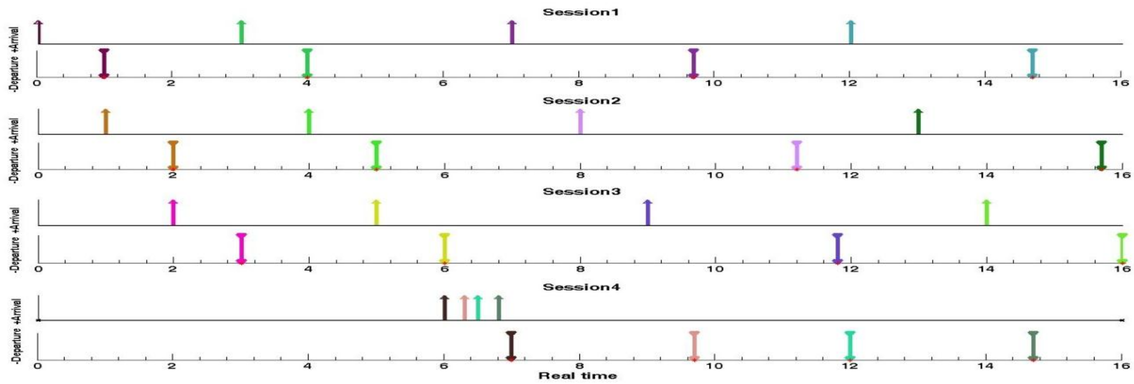


FIGURE 3-THE GPS TREATMENT WITHIN AN EXAMPLE

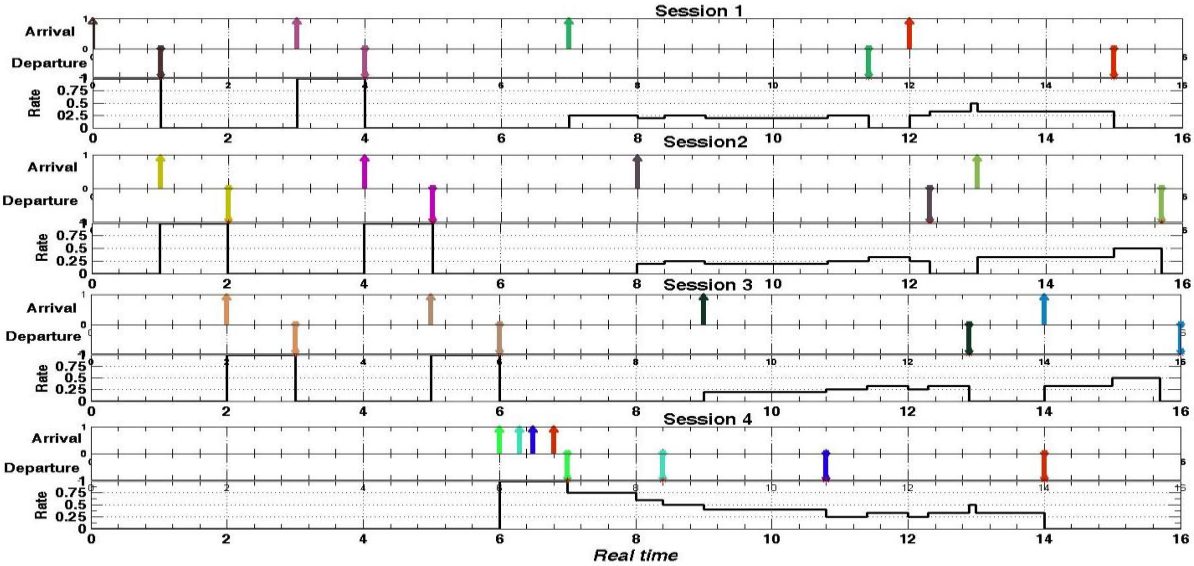


FIGURE 4-THE RBSS TREATMENT WITHIN AN EXAMPLE

$f_2$and f_4 will change to 0.2, 0.2, 0.6 packet/s respectively. The service of the 2nd packet of f_4 will be complete at 8.42 second, because its remaining 25% service is served at rate of 0.6 packet/s.

For the above examples, the average and variance of packet delays of each session are presented in Table I.

The average delay of session f_4 is decreased by 20% (from 4.4 second in GPS to 3.6 second in RBSS). It should be clear that the decrease in delay of session f_4, is achieved at the expense of additional delay for other sessions which is about 20% in this example (from 1.8 second in GPS to 2.2 second in RBSS).

To evaluate RBSS vs. GPS, let us compare the ratio of the average delay of session f_4 to those of other sessions (f_1 to f_3). This ratio is 1.63 for RBSS while it is 2.4 for GPS, which shows a decrease by 30% for RBSS.

The variance of the packets delay might be used to evaluate jitter criteria. It is obvious that for session f_4 this parameter is also decreased from 8.7 in GPS to 7.5 in RBSS.

IV. SIMULATION AND EVALUATION

In this section we simulate and compare the performance of RBSS and GPS schedulers in a single node. In this simulation, four sessions are scheduled in two scenarios S_1 and S_2.

The first scenario is designed to show the performance of the scheduler when there is one bursty session among three non bursty sessions. The second scenario illustrates the treatment of different sessions with different burstiness parameters by RBSS. In order to compare the performances, the average packet delay, packet delay variance and maximum packet delay for each session are computed in each case.

In the next two parts, the parameters of the traffic generator are described first, and then the simulation results are presented.

A. Simulation Parameters

In both scenarios S_1 and S_2, it is assumed that there are four active sessions and the distributions of packet inter arrival time

TABLE II TRAFFIC PARAMETERS

Scenario	Parameter	Session # (i)			
		1	2	3	4
S_1	a_i (sec)	0.09	0.09	0.09	0.05
	c_i	10	10	10	2
	σ_i(Kbit)	4	4	4	40
	ρ_i (Kbps)	10	10	10	10
S_2	a_i (sec)	0.09	0.08	0.05	0.05
	c_i	10	5	2	2
	σ_i (Kbit)	4	20	40	100
	ρ_i (Kbps)	10	10	10	10

are *Pareto*. Pareto is common to use for modeling of bursty traffic [37]. This distribution has two parameters: *a* and *c*. Parameter *a* indicates the minimum value of the random variable and *c* is the shape parameter which determines the probabilistic distribution function, *pdf*. Parameter *c* changes the burstiness of the traffic. Different traffic patterns have been generated with similar characteristics by changing the seed values. Traffic generator parameters are summarized in Table II. It is assumed that all packets have the same size which equals 1 Kbits. The generated traffic passes through the LB shaper with (σ_i, ρ_i) for each session shown in Table II. The simulation time is considered to be 500 sec for all scenarios.

The S_1 and S_2 scenarios are divided further into 3 cases each using different BSP_i and MBT_i parameters. The values of the RBSS parameters are shown in Table III. It is assumed that in S_{11} and S_{21}, the values of BSP_i are equal to zero for all sessions, therefore the scheduler would be the same as that of GPS. It is assumed that the requested service rate of all sessions in all simulation cases are equal to 10 Kbps and the output link rate equals to 40 Kbps.

B. Simulation results

In this part, we present and compare the average, maximum and variance of the packet delay which is calculated from the various scenarios in the simulation.

a. Simulation results for the scenario S_1

The simulation results for scenarios S_1 are shown in Figure 5. These results indicate that the average, maximum and variance of the delay parameter for bursty session f_4 are significantly reduced by RBSS (scenariosS_{12} and S_{13}) in comparison with GPS (scenario S_{11}). In this scenario using the GPS scheduler, the average delay of session f_4 is more times greater than the delay of those other sessions. However using RBSS, the average delays of all sessions are almost the same.

Additionally, it is clear in Figure 5 that the delay variance of session f_4 is extremely reduced in RBSS.

b. Simulation results for the scenario S_2

The simulation results of scenarios S_2 are shown in Figure *6*. Again, the results of this scenario indicate that the average,

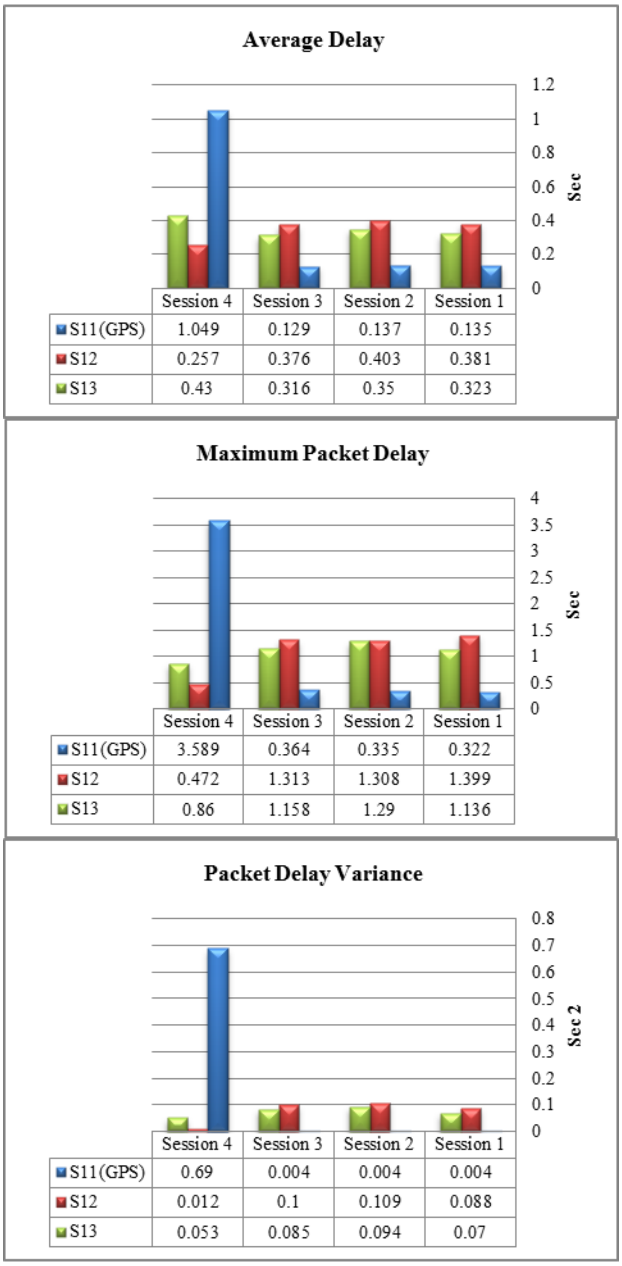


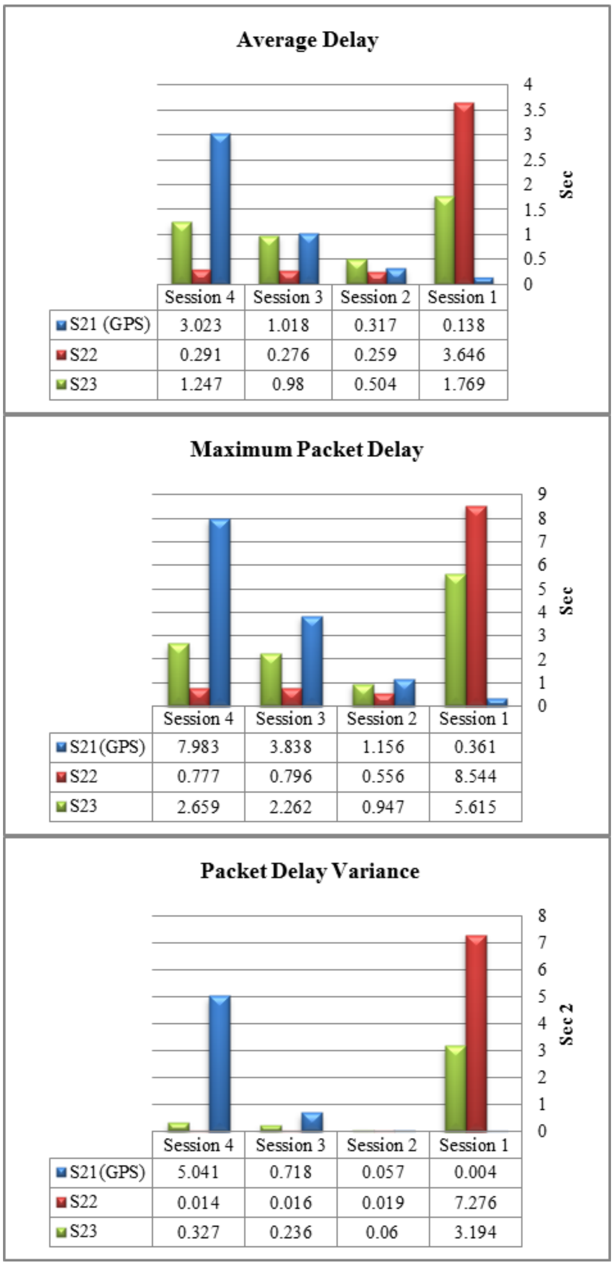
FIGURE 5- THE SIMULATION RESULTS FOR 1ST SCENARIO

maximum and variance of the delay of the bursty sessions (i.e. sessions f_2- f_4) are reduced by RBSS compared to GPS results. Sessions with more burstiness experience more decrease in their average and maximum delay.

In this scenario, when GPS is used, the average delay of session f_4 is very more times greater than session f_1, while in RBSS , the average delay of session f_4 is extremely reduced. But in RBSS , unlike session f_4, the average delay of session 1 has been increased, especially for case S_{22}. This is due to the excessive increase of the BSP_4 parameter in this case. This effect will be balanced by reducing this parameter as shown for case S_{23} in Table III.

TABLE III THE RBSS PARAMETERS

Scenario	Parameters	Session #			
		1	2	3	4
S_{11}(GPS)	**MBT_i (Kbit)**	4	4	4	40
	BSP_i	0	0	0	0
S_{12}	**MBT_i(Kbit)**	4	4	4	4
	BSP_i	0	0	0	9
S_{13}	**MBT_i (Kbit)**	4	4	4	8
	BSP_i	0	0	0	4
S_{21}(GPS)	**MBT_i (Kbit)**	4	20	40	100
	BSP_i	0	0	0	0
S_{22}	**MBT_i (Kbit)**	4	4	4	4
	BSP_i	0	4	9	24
S_{23}	**MBT_i (Kbit)**	2	4	8	10
	BSP_i	1	4	4	9



Average Delay

	Session 4	Session 3	Session 2	Session 1
S21 (GPS)	3.023	1.018	0.317	0.138
S22	0.291	0.276	0.259	3.646
S23	1.247	0.98	0.504	1.769

Maximum Packet Delay

	Session 4	Session 3	Session 2	Session 1
S21(GPS)	7.983	3.838	1.156	0.361
S22	0.777	0.796	0.556	8.544
S23	2.659	2.262	0.947	5.615

Packet Delay Variance

	Session 4	Session 3	Session 2	Session 1
S21(GPS)	5.041	0.718	0.057	0.004
S22	0.014	0.016	0.019	7.276
S23	0.327	0.236	0.06	3.194

FIGURE 6-THE SIMULATION RESULTS FOR 2ND SCENARIO

Additionally, Figure 6 indicates that the packet delay variances of bursty sessions have been decreased. But, for sessions with more burstiness, the decrease of delay variance is more. Also it indicates that in RBSS there is a tradeoff between providing burst service and rate-fair service. In RBSS, while a burst from a bursty session is being served other sessions have to tolerate more delays, therefore the delay and delay variation of sessions with less burstiness will be increased.

V. CONCLUSION

In this paper, a novel scheduling algorithm (RBSS) has been presented to provide a better service for rt-VBR applications where the session has a delay-sensitive bursty traffic. In RBSS scheduling algorithm, we assign triple parameters (r, MBT and BSP) to each session. The *Burstiness* parameter of traffic is given from the *leaky bucket* concept. By precisely assigning the parameters in RBSS, the received service curve can coordinate with the arrival constraint curve.

In RBSS instantaneous weight of a bursty session will be increment when the queue length of the session goes beyond a specified threshold (MBT). Thus, instantaneous service rate may be increased by growth of queue length. Larger amount of BSP indicates that when a session receives a burst, the amount of instantaneous weight can be increased much more. Thus, the service provided to the bursty session can be increased and the average service time of the bursty session decreases compared with other sessions. The proposed algorithm is useful to provide quality of service in special sessions like video streams, which are not only inherently bursty but also delay- sensitive.

The simulation results show that the service provided to a bursty session in RBSS outperforms GPS in terms of packet delay. For a bursty session with higher priority (BSP), maximum and average of packet delay are less than GPS system in the same condition. However, the average and maximum packet delays in sessions with lower amount of BSPs, would be increased. This is justifiable according to the parameter selection. The increment in other sessions delay may be small if burst arrivals in different sessions do not occur simultaneously. After burst duration of a typical session elapsed, it is expected that its arrival rate reduces for a while. Hence, although other sessions are served less than usual in the burst duration time, the reduction in their service will be compensated when the bursty session becomes quiet. A suitable connection admission control (CAC) unit can be design to determine how many numbers of bursty sessions can be accepted to guarantee a minimum service rate in all sessions. We will study this issue in our future works.

Finally, this paper shows that in addition to fairness indexes which are necessary to evaluate schedulers in term of the service rate, another metric is required to evaluate the reaction of the scheduler at the presence of bursty traffic. We plan to analyze the details of the metric in our future works.

ACKNOWLEDGMENT

The authors would like to thanks Ali Ghiasian for editing the paper and for useful comments.

REFERENCES

[1] A. Sabri; H. El-Badawy, K. Shehata, A. Ali, "Video Scheduling in the Uplink over WiMAX Networks by the Novel Concept Adaptive Minimum Service Interval"" in Proceeding of International conference on computer engineering and applications, ICCEA, Vol.2 pp.3-7, 2010.

[2] R. Kalle, U.C.Devi, S.Kalyanaraman, "Multi-tiered, burstiness-aware bandwidth estimation and scheduling for VBR video flows", in Proceeding of IEEE International Workshop on Quality of Service (IWQoS), pp.1-9, 2011.

[3] J. F. Shortle, M. J. Fischer," Approximation for a two-class weighted fair queueing discipline", Performance Evaluation, Elsevier Journal, Vol.67 pp.946-958, 2010.

[4] Z. Dwekat, G.N. Rouskas, "A practical fair queuing scheduler: Simplification through quantization", Computer Networks, Elsevier Journal, 55 pp.2392–2406, 2011.

[5] I.Vaishnavi, D.Bulterman, Estimate and serve: scheduling soft real-time packets for delay sensitive media applications on the internet", in Proceeding of NOSSDAV, pp. 109-114, 2009.

[6] F. Sbrina, "A novel resource scheduling algorithm for QoS-aware services on the Internet", Journal of Computer and Electrical Engineering, Vol.36 No.4, pp. 718–734, 2010,

[7] H. Zhang and S. Keshav, "Comparison of rate-based service disciplines", in Proceedings of the conference on Communications architecture & protocols, ACM, 1991, p. 121.

[8] A. Parekh and R. Gallager, “A generalized processor sharing approach to flow control in integrated services networks: the single-node case”, IEEE/ACM Transactions on Networking (TON), Vol. 1, No. 3, pp. 344–357, 1993.

[9] S. Golestani, “A self-clocked fair queueing scheme for broadband applications”, in Proceedings of IEEE INFOCOM, Vol. 94, pp. 636–646, 1994.

[10] D. Stiliadis and A. Varma, “Efficient fair queueing algorithms for packet-switched networks,” IEEE/ACM Transactions on Networking (TON), Vol. 6, No. 2, p. 185, 1998.

[11] J. S. Turner, “New directions in communications (or which way to the information age?)”, IEEE Communications Magazine, Vol.24, No.10, pp.50-57, 1986.

[12] Raymond W. Yeung, “Fundamental Tradeoff between Burstiness and Delay in Traffic Shaping”, Information Theory, 2005

[13] X. Jin, G. Min, “Analytical queue length distributions of GPS systems with long range dependent service capacity”, Simulation Modeling Practice and Theory, Elsevier Journal, Vol.17 pp. 1500–1510, 2009.

[14] P. Giacomazzi, G. Saddemi, “End to end delay approximation in cascades of Generalized Processor Sharing Schedulers”, in Proceeding of IEEE ICC, pp. 1-6, 2009.

[15] M. Lelarge, “Asymptotic behavior of generalized processor sharing queues under subexponential assumptions”, journal of Queueing systems, Vol. 62, No 1-2 ,pp. 51-73, 2009.

[16] S. Singh, “Round-robin with credits: an improved scheduling strategy for rate-allocation in high-speed packet-switching”, in Proceedings of IEEE GLOBECOM, Vol.1, pp.334-338, 1994.

[17] D. Sh. Lee, Ch. M. Chen, and Ch.-Y. Tang, “Weighted Fair Queueing and Compensation Techniques for Wireless Packet Switched Networks”, IEEE Transactions on Vehicular Technology, Vol.56 ,No.1, pp.887–891, 2007.

[18] M. M. N. Aldeer, “Performance comparison of packet-level multiplexing algorithms with bursty traffic, Journal of Engineering Science and Technology, pp.46-52, 2010.

[19] H. Sariowan, RL. Cruz, GC. Polyzos, “Scheduling for Quality of Service Guarantees via Service Curves ” , in Proceedings of ICCCN, pp.512-520, 1995.

[20] S. J. Golestani, “Congestion-free communication in high-speed packet network”, IEEE Transactions on Communications, Vol.39, No.12, pp.1802-1812, 1991.

[21] M. Hamadoui and P. Ramanathan, “A dynamic priority assignment technique for streams with (m, k)-firm deadlines”, IEEE Transactions on Computers Communication, Vol.44 , No.12, pp.1443-1451,1995.

[22] Y.-J. Choi, H.-M. Kim, “A new scheduling scheme: for high-speed packet networks Earliest-virtual-deadline-first, Computer Communications Elsevier Journal,Vol.30,pp.2291–2300, 2007.

[23] M Hanada, H. Nakazato, “ORC-GPS: Output Rate-Controlled Scheduling Policy for Delay Guarantees”, in Proceedings of ICC, 2007 , pp. 530 – 535.

[24] T.-G. Kwon, S.-H. Lee, J.-K. Rho, “ Scheduling Algorithm for Real-time Burst Traffic using Dynamic Weighted Round Robin”, in Proceeding of IEEE International Symposium on Circuits and Systems, No. 6, pp. 506-509, 1998.

[25] H. Zhu, “On Improving Service Differentiation under Bursty Data traffic in Wireless Network”, in Proceedings of IEEE INFOCOM , Vol.2 , pp. 871 – 881, March 2004.

[26] T. Quynh, H. Karl, A. Wolisz, and K. Rebensburg, “Relative jitter packet scheduling for differentiated services”, in Proceedings of IFIP conference performance Modeling and Evaluation of ATM and IP networks, 2001.

[27] M. Yaghmaei, M. bahekmat, and T. Khojasteh, “A Novel Fuzzy Logic Base Scheduling Mechanism for Service Differentiation in IP Networks”, Journal of Iranian Association of Electrical and Electronics Engineers, 2008.

[28] V. S. Frost and B.Melamed,“ Traffic Modeling For Telecommunications Networks”, IEEE Communications Magazine, pp.70-81 March 1994.

[29] D. Jagerman and B. Melamed, “Burstiness descriptors of traffic streams: Indices of Dispersion and Peakedness”, in Proceedings of Annual Conference on Information Sciences and Systems, pp. 24–28,1994.

[30] H. M. Au. Tat-Ming, “Delay performance of a leaky bucket regulated multiplexer based on burstiness characterization”, in Proceeding of IEEE Global Telecommunications Conference, vol. 2, No. 13–17, pp. 843–847, 1995.

[31] R. Cruz, “A calculus for network delay, part I: Network elements in isolation”, IEEE transactions on Information Theory, vol. 37, No. 1, pp.114–131, 1991.

[32] L. Wang, G. Min, and I. Awan, “Stochastic Modeling and Analysis of GRED-I Congestion Control for Differentiated Bursty Traffic,” in Proceeding of International Conference on Advanced Information Networking and Applications, pp. 1022–1030, 2007.

[33] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, “On the self-similar nature of Ethernet traffic (extended version)”, IEEE/ACM Transactions on Networking (ToN), Vol. 2, No. 1, pp. 1–15, 1994.

[34] K. Siriwong, L. Lipsky, and R. Ammar, “Study of bursty internet traffic”, in Proceeding of IEEE International Symposium on Network Computing and Applications, pp. 53–60, 2007.

[35] A. Neidhardt and J. Wang, “The concept of relevant time scales and its application to queuing analysis of self-similar traffic (or is Hurst naughty or nice?),” in Proceedings of ACM Joint International Conference on Measurement and Modeling of Computer Systems. pp. 222–232, 1998.

[36] Y. Kodama, T. Kudoh, M. Matsuda, F. Okazaki, “Real-time burstiness measurement”,in Proceeding of International. Workshop on Protocols for Fast Long-Distance Networks (PFLDnet), 2006.

[37] M. P.Farrera, M.Fleury, K. Guild, and M.Ghanbari, “Measurement and analysis study of congestion detection for Internet Video Streaming”, Journal of communications, Vol. 5, No. 2, pp.169-177, February 2010.