IMPROVING SCFQ TO SUPPORT BURSTY TRAFFIC

Mahmoud Daneshvar Farzanegan[1], Hossein Saidi[1] andMehdiMahdavi[1]

[1] ECE Department, Isfahan University of Technology, Isfahan, Iran
m_daneshvar@ec.iut.ac.ir,[hsaidi, m_mahdavi]@cc.iut.ac.ir

ABSTRACT

Quality of Service (QoS) provisioning in networks requires proper scheduling algorithm. Internet traffic, especially real-time multimedia applications, is bursty in nature . Hence, in addition to the service rate which is commonly used to isolate service of sessions, other parameters should be involved. In this paper a scheduling algorithm is proposed that attempts to provide a balance between bursty and non-bursty (smooth) traffic. We improve thewell-known packet scheduling algorithm, SCFQ.Our proposed algorithm is proficient to compensate an adjustable amount of missed service to each session. The average delay of the proposed algorithm is evaluated by simulation. An important advantage of our algorithm is that byselecting correct parameter setting for each session, the average delay of a bursty session can be reduced. Furthermore, compared to SCFQ our proposed algorithm does not necessitate any additional computation.

KEYWORDS

Scheduling, Fair Queuing, QoS, Bursty traffic

1. INTRODUCTION

Different kinds of services in multimedia networks need different kinds of quality of services (QoS). QoS is usually quantified by end-to-end delay, service rate, delay variation (jitter) and packet loss rate. QoS provision might be accomplished in different network layers or devices such as switches or routers. Due to their key role in QoS provisioning, scheduling algorithms have received much attention in the literature [1]-[6].

In the development of packet scheduling algorithms, with respect to internal structures, there are two categories of scheduling algorithms: sorted-priority and frame-based [1]. In a sorted-priority category, a set of potential functions or virtual clocks are defined. When a packet arrives in or departs from the server, the values of the virtual clock are updated. For each waiting packet in the queue, a time stamp is calculated from the virtual clocks. The scheduler sorts packets in order of the time stamps and serves a packet with the highest priority [7]-[8]. A useful survey can be found in [2].

In the other category i.e. frame-based, a duration of time is defined as the frame which includes time-slots. Based on the requested service rate, each session reserves some time-slots in the frame. The round robin family is a major kind of algorithms in this category [9],[10].

In both of the above categories, the reserved or requested service rate is usually applied to isolate sessions. Many kinds of packet scheduling algorithms e.g. WFQ, SCFQ, GR, DRR etc. have been proposed to distribute available bandwidth fairly among all backlogged sessions in terms of requested service rates, hence, this kind of scheduling algorithm is called Rate Proportional Server (RPS)[1]. However, most of the RPS schedulers do not have acceptable performance when bursty traffic and non-bursty (smooth) sessions compete. Suppose a bursty and a smooth session, equal in term of requested service rates are scheduled by an RPS. When

DOI : 10.5121/ijwmn.2013.5202

the bursty session starts to send data after having been idle while the smooth session received regular service, the RPS scheduler serves both sessions similarly. Because only the service rates are involved in the scheduling algorithm, unused service in idle duration cannot return back to the bursty session. In the next section, the treatment of bursty traffic and non-bursty traffic under by an RPS scheduler is studied with an example.

The bursty nature of internet traffic [11] and QoS requirements in multimedia applications motivate us to consider in addition to the requested service rate, another parameter which would describe the need for bursty service in the scheduling algorithm. However, the scheduling of bursty traffic has been an attractive subject for investigation.[12]-[28]. In these studies the authors attempt to design scheduling algorithms that provide acceptable service for bursty traffic by involving more parameters such as delay due time, unused credit or burst specification in the scheduling algorithms. By adjusting the proper parameters, a better QoS can be provided to each bursty session.

In some studies, traffic is modelled by a series of bursts, and the scheduling algorithm takes into account its special requirements and specifications [17]-[19]. This model is suitable for compressed video such as the MPEG format [17], or optical burst switches (OBS)[20]-[21].
Some scheduling algorithms which consider bursty traffic and channel disappearance in wireless networks are studied in [22]-[24]. In [22] some credit values are proposed to count number of unavailable or unused service for each session. Some other studies define a due time threshold for each session and the server tries to send the packets within the timing limit of this threshold [18]-[25].
There are also 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.

In this paper, we improve SCFQ[7] the famous RPS scheduling algorithm, to provide fair service to both bursty and non-bursty flows. A dominant feature of SCFQ is that its fairness index (which is based on the service rate) is less than twice the minimum value that can be achieved for a packetized scheduling algorithm [7].

The organization of the paper is as follows. Section II presents our proposed scheduling algorithm named Burst-Service Self Clock Fair Queuing (BSCFQ). In section III, we study the performance of the proposed scheduling method by a simulation model. Section IV concludes the paper.

2. BURST SERVICE SCFQ

In this section, we present a modification of a known scheduling scheme named SCFQ that provides proper service to burst traffic. First, we define some parameters and criteria which are used in the remainder of the paper.

2.1. Definitions

Definition 1-*The amount of service:*

Let $S_i(t_1, t_2)$ denote the amount of service which session f_i receives during (t_1, t_2). The amount of service is equal to the sum of all packet lengths in f_i which are completely served by the server. The amount of service is measured in term of data unit e.g. bit.

Let ρ_i denote the reserved service rate, then the *relative service* of the session f_i is shown by $\hat{s}_i(t_1, t_2)$ and defined as follows: $\frac{S_i(t_1,t_2)}{\rho_i}$.

Definition 2- *Postponed service*:

Due to lack of data, each session f_i may receive service which is less than its reserved service. We call the amount of service which is not offered to a session due to the lack of data as the "*postponed service*".

Definition 3- *Compensable service threshold*:

Compensable service is defined when a server tries to offer more service to a session which has some amount of postponed service and finally can compensate it in a limited time period. This amount of postponed service which can be compensated is called compensable service. Compensating of service should be carried out in a condition in which other sessions are backlogged. The maximum amount of service which can be compensated by a server for each session is called its *compensable service threshold*.
Let γ_i denote the compensable service threshold for session f_i in this paper.

In packet by packet scheduling algorithms (no preemptive type), the server does not start serving a new packet before the service of the last packet is finished. Therefore, it is possible to consider a condition in which one packet is served more than accords with the share of session f_i. Consequently, the amount of service may advance up to L_i (the maximum packet length in f_i). Therefore, we can say that if session f_i has a postponed service equal to L_i, the server can compensate for it in the assumed condition. With respect to definition 3, in packet by packet scheduling algorithms we can conclude that the compensable service threshold should be at least equal to L_i or: $\gamma_i \geq L_i$ (1)

2.2. Self-Clock Fair Queue

SCFQ [7] is a *sorted-priority* scheduler that defines the finishing time stamp for each packet as follows: $F_i^k = max\{F_i^{k-1}, F_{Current}\} + \frac{L_i^k}{\rho_i}$ (2)
Where F_i^k is the finishing time stamp of the kth packet of session f_i. The virtual clock of the system in SCFQ is denoted by $F_{Current}$ which is equal to the finishing time stamp of the packet which is being served when the kth packet arrives. L_i^k and r_i are the packet length of the kth packet and requested service rate in session f_i. This formula estimates the finishing service time of each packet in a fluid flow system and applies it in packet by packet scheduling. The calculating of (2) is easier than some other estimations such as those in WFQ.

2.3. Burst-Service Self-Clock Fair Queue

In addition to the request service rate, i.e. ρ_i, we apply another non-negative parameter in BSCFQ which is named *flash back* and denoted by fb_i. *Flash back* is a parameter which adjusts the amount of compensable service threshold by playing back the virtual time of the system in the session f_i's view, therefore fb_i is measured by the duration of the second.
The main difference between SCFQ and BSCFQ is in the calculation of the finishing time stamp. In BSCFQ the finishing time stamp of the kth packet in f_i is calculated as follows:

$$F_i^k \triangleq \begin{cases} 0 & k = 0 \\ max\{F_i^{k-1}, vc(a_i^k) - fb_i\} + \frac{L_i^k}{\rho_i} & k > 0 \end{cases} \quad (3)$$

Where a_i^k is the arrival time of the kth packet and fb_i is the flash back parameter in session f_i. $vc(t)$ is the overall *virtual clock* of the system and defined as:

$$vc(t) \triangleq max\{vc(t_{n-1}), F_{Current}\} \qquad t_n \leq t < t_{n+1} \quad (4)$$

$F_{Current}$ is the finishing time stamp of the packet that is being served in the server at time t and t_n is the n^{th} moment in which vc(t) is modified and $vc(t_{n-1})$ is the previous stored virtual clock . We assumed that: $vc(0)=0$.

For each session also we define a virtual clock as follow:

$$vc_i(t) \triangleq F_i^{HoL} - \frac{L_i^{HoL}}{\rho_i} \quad (5)$$

Where F_i^{HoL} and $L_i^{HoL}$are the finishing time stamp and length of the packet which is *head of line* in session f_i at time t. If a session is not backlogged, i.e. there is no packet as a head of line packet, then the time stamp of the previous packet is considered as $vc_i(t)$. We also assumed that: $vc_i(0)=0$. Let $vc(t_1, t_2)$ denote $vc(t_2) - vc(t_1)$.

Similar to SCFQ, we assume that by ending a busy period, i.e. when the server has no more packet in the queue, the algorithm reinitialized by setting all the vc ,vc_i and packet counts in each session f_i to zero.

Note that when a session becomes backlogged after an idle period, (3) indicates that F_i^kmay be less than $vc(a_i^k)$. Therefore, it is possible that $F_{Current}$ will be less than the finishing time of the last served packet . By defining (4), in this case the value of the vc(t) remains constant.

In acondition in which all sessions are regularly backlogged $F_{Current}$is less than or equal to the finishing stamp of the head of line packet in each session due to the packet selection mechanism which a packet by the minimum value of the finishing time stamp is selected.

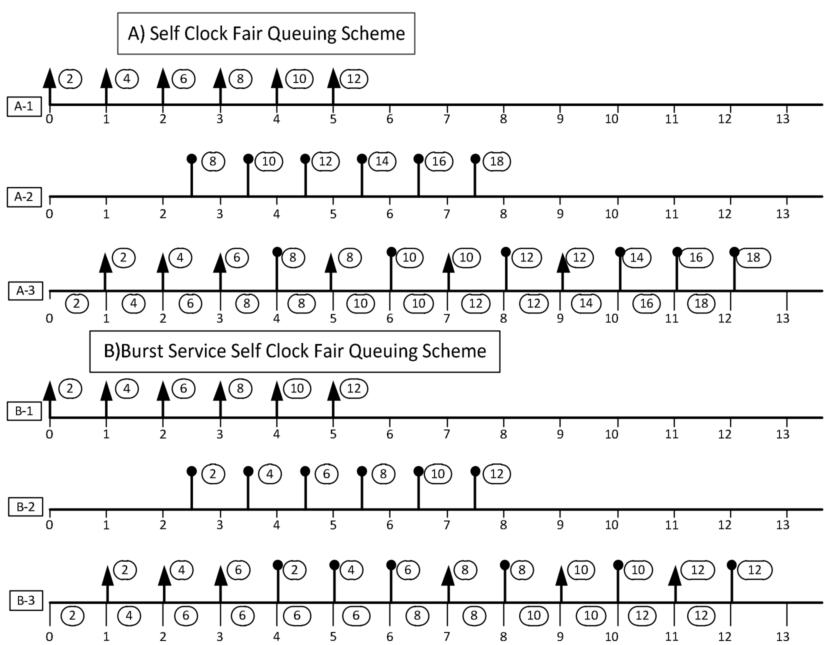


Figure 1. Packet Arrival and Departure Times in Ex.1

2.4. Example

In the following examples, we attempt to illustrate the mechanism of our algorithm. BSCFQ and SCFQ results are compared to indicate the benefits of this modification.

It is assumed that there are 2 sessions, all packets lengths are the same and the requested service rates, (ρ_1 and ρ_2) are equal to 0.5 packet/sec in both Ex. 1 and Ex.2. The flash back of the sessions f_1 and f_2 (fb_1 and fb_2) are equal to 0 and 6 sec respectively in both examples. Arrivals in example1 and example2 are different.

Figure 1 indicates the arrivals and departures in Ex.1 for both SCFQ and BSCFQ algorithms. In each algorithm (for example part A), arrows in the diagrams A-1 and A-2, indicate the arrival times of packets in f_1 and f_2. The finishing time stamps of each packet are also indicated above each packet's arrow. Diagrams A-3 and B-3 indicate the moments of packet departure from the system. Meanwhile, we indicate the finishing time stamp of the packet above each arrow and the virtual clock of the system between both departure times is shown in diagrams A-3 and B-3.

In both examples, if the service of f_2 is postponed then the BSCFQ scheduler is able to compensate for only 3 postponed packets. The packet arrivals in Ex.1 indicate that when 3 packets are served from f_1, session f_2 is backlogged. In SCFQ the service is equally divided between two sessions after the 3rd second and cannot compensate for the postponed service. But in BSCFQ, first the postponed service is served to f_2 then the service is equally divided between the two sessions. Ex.2 shows that the amount of service which can be compensated for by BSCFQ is limited even if the postponed service is more than 3 packets.

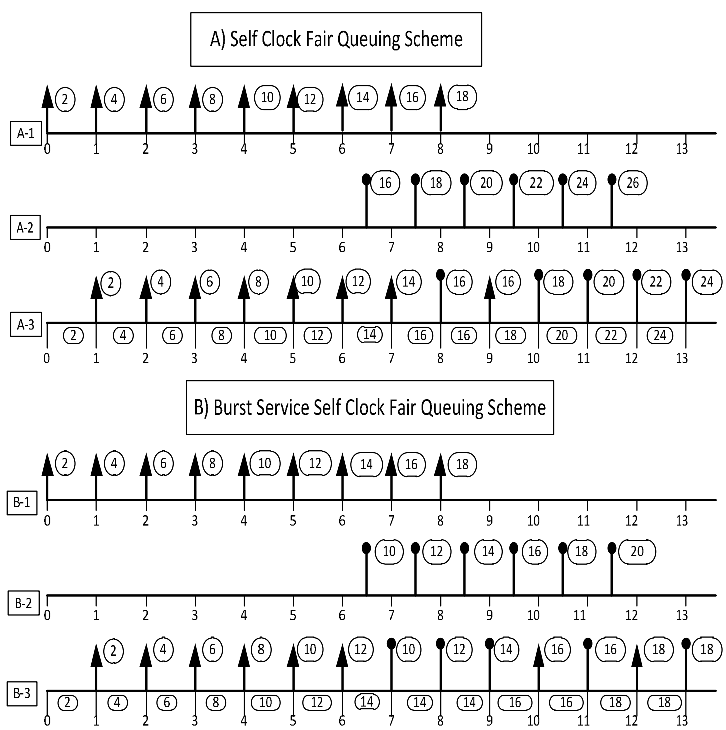


Figure 2. Packet Arrival and Departure Times in Ex.2

Average packet delays in Ex.1 and Ex.2 are collected in Table 1. The average delay of f_2 in both examples is decreased by BSCFQ. The results indicate that it is possible to adjust BSCFQ to compensate for a desired value of postponed service and thus, to decrease its average packet delay.

Table 1. Average packet delay in SCFQ and BSCFQ

Example		Ex.1		Ex.2	
method		SCFQ	BSCFQ	SCFQ	BSCFQ
Average Delay	f_1	1.00	2.00	0.13	0.88
	f_2	2.50	1.50	2.33	1.33

3. SIMULATION MODEL

With the aim of evaluating the average packet delay in the performance of the BSCFQ algorithm, we create a simulation model using SIMULINK® in MATLAB. The model includes 4 sessions with equal arrival rates but different burstiness parameters. The arrival traffic in the f_1 and f_4 sessions is considered to be smooth but for the f_2 and f_3 sessions it is bursty.

In the first scenario, we assumed that the f_1 and f_4 sessions have a constant packet rate with packet inter arrival time equal to 4 sec while sessions f_2 and f_3 have switched (ON-OFF) traffic models. During ON intervals a constant packet rate is used with inter arrival time equal to 0.1 sec, while no packet is generated during OFF intervals. ON and OFF intervals are constant but their periods are different in f_2 and f_3. We assume that the ON interval is 0.7 sec with a 28 sec period in f_2 while it is 0.3 sec with 12 sec period in f_3. The mean arrival rates in all sessions are the same and equal to 0.25 pkt/sec. The capacity of the output link is considered as 1 pkt/sec and all packets are to have the same length. Figure 3 indicates the arrivals in the first scenario. It is obvious that the postponed service in sessions f_2 and f_3 may be respectively 7 and 3 in each period.

In order to study the effect of random arrival in our discussion, we build another scenario in which the arrival process in one of the sessions (e.g. f_4) is Poisson. In the second scenario, we assume that in session f_4 packet inter arrival time is exponential (with parameter 4) instead of being constant. All other assumptions are the same as in the first scenario.

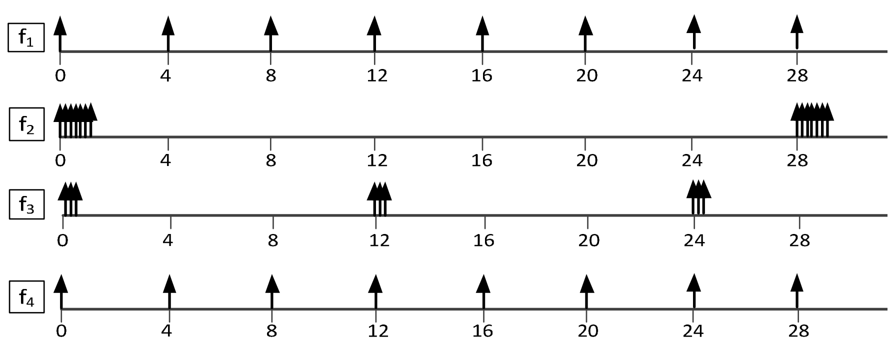


Figure 3.Arrivals in simulation model

Table 2. Values of Flash Back in f_2 and f_3 sessions

	Ex. 1									Ex.2	
Session	$BSCFQ_1$	$BSCFQ_2$	$BSCFQ_3$	$BSCFQ_4$	$BSCFQ_5$	$BSCFQ_6$	$BSCFQ_7$	$BSCFQ_8$	$BSCFQ_9$	$BSCFQ_1$	$BSCFQ_2$
f_2	7	14	21	28	0	0	0	14	28	14	28
f_3	0	0	0	0	4	8	12	6	12	6	12

We also examine SCFQ and BSCFQ with various *flash back* parameters for f_2 and f_3. A summary of the parameters for each different case is given in Table 2. For the first example, the value of flash back for session f_2 is increased in 4 steps ($BSCFQ_1$ to $BSCFQ_4$); it is also increased in another session, f_3, in 3 steps ($BSCFQ_5$ to $BSCFQ_7$). Finally, the flash back values in both f_2 and f_3 sessions are changed in 2 steps in the first example ($BSCFQ_8$ and $BSCFQ_9$) and also in the second example ($BSCFQ_1$ and $BSCFQ_2$).

3.1. Simulation Results

The average and maximum delays of each session are computed for 1000 sec in the first scenario. The delay of each packet is computed during simulation time and simulation results are shown in Table 3 and depicted by Figures4 and 5 respectively which show the average delay for different values of flash back in f_2 and f_3.

Table 3Average and maximum of packet delay in the first scenario

Scheduling Algorithm	**Average of delay ineach session**				**Maximum delay of each session**			
	f_1	**f_2**	**f_3**	**f_4**	**f_1**	**f_2**	**f_3**	**f_4**
SCFQ	3.53	9.30	6.49	1.97	10.05	21.40	10.80	4.05
$BSCFQ_1$	3.55	9.09	6.66	1.99	10.05	21.40	10.80	4.05
$BSCFQ_2$	4.59	6.59	7.41	2.69	8.05	18.40	14.80	6.05
$BSCFQ_3$	5.26	4.29	8.16	3.56	9.05	10.40	14.80	7.05
$BSCFQ_4$	5.42	3.81	8.34	3.69	12.05	8.40	14.80	7.05
$BSCFQ_5$	3.10	10.09	6.03	2.07	6.05	22.40	10.80	4.05
$BSCFQ_6$	2.75	13.06	3.72	1.75	4.05	25.40	6.80	3.05
$BSCFQ_7$	3.03	13.36	2.87	2.03	4.05	26.40	4.80	3.05
$BSCFQ_8$	5.02	8.59	4.69	3.00	7.05	21.40	8.80	5.05
$BSCFQ_9$	5.94	8.50	4.14	2.71	11.05	21.40	8.80	5.05

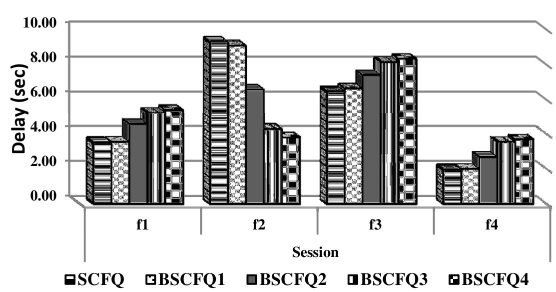


Figure 4The packet delay averagein eachsession at different cases
(The flash backof session $f_2$is changed).

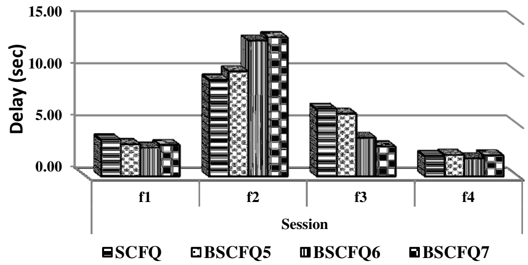


Figure 5The packet delay averagein eachsession at different cases
(The flash back of session $f_3$is changed).

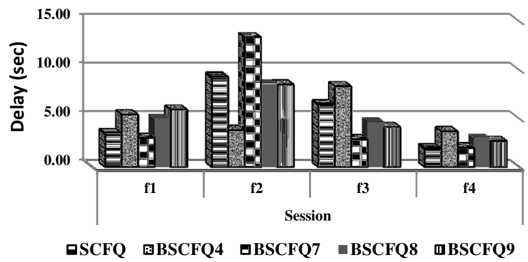


Figure 6The packet delay averagein different sessions at different cases
(The flash back values of both sessions f_2 and f_4 are changed)

Figure 6shows the average delay of each session when the flash back of both f_2 and f_3 sessions are increased ($BSCFQ_8$ and $BSCFQ_9$). SCFQ, $BSCFQ_4$ and $BSCFQ_7$ are shown in this figure for comparison.

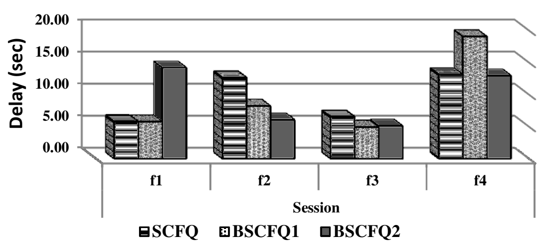


Figure 7 Average delays in the second scenario

Table 4 Average packet delay in the second scenario

Scheduling Algorithm	**Average delay of each session**			
	f_1	f_2	f_3	f_4
SCFQ	5.78	12.69	6.56	13.18
BSCFQ1	5.80	8.23	4.93	19.06
BSCFQ2	14.21	6.09	5.11	12.98

Table 4 and Figure 7 show the average delay of the second scenario. We consider only two cases for this scenario where the value of the flash back in sessions f_2 and f_3 are non-zero.

3.2. Discussion

With respect to Figure 4 and Figure 5, we can see that by increasing the value of flash back of a session, (our proposed parameter for the adjustment of the amount of compensable service threshold) the corresponding average delay is decreased. When the flash back of a session increases, however, average delays in other sessions may either increase (e.g. f_1 , f_3 and f_4 in Figure4 and f_2 in Figure5) or show no noticeable change (e.g. f_1 and f_4 in Figure 5). This effect is predictable due to the fact that overall average packet delay during a busy period is constant when the server is in work-conserve mode. A work-conserve server is a server which is busy during every time period that at least one session is backlogged. Therefore, when the average delay of a session during a busy period decreases, the average delay in other sessions should be increased or kept constant to ensure the overall average delay remains constant.

Figure 6 indicates the average delay when we want to provide burst service to both of the bursty sessions. Delay average in both bursty sessions f_2 and f_3 are reduced by increasing the flash back value (compare $BSCFQ_8$ and $BSCFQ_9$ with SCFQ). Although average delay in other sessions (e.g. f_1 and f_4) increase in $BSCFQ_8$ and $BSCFQ_9$ in comparison with SCFQ, the growing of average delay in other sessions is slighter than that in $BSCFQ_4$ and $BSCFQ_7$ where the server provides burst service only to one session.

These results are satisfied when the arrivals in session f_4 are random. The results in Figure 7 show as is the case in the first scenario, the average delays of f_2 and f_3 are reduced in comparison with the delay in SCFQ.

We can also observe that the average delay of f_4 is increased in scenario2 as compared to scenario1. The random nature of the arrival process can be considered as the main factor in the increase in the average delay of f_4. Therefore, we conclude that BSCFQ can reduce the average delay of bursty sessions even when the arrival is random.

4. CONCLUSIONS

In this paper, we have proposed BSCFQ to improve SCFQ, as a known RPS scheduling algorithm. In addition to *request service rate*,we introduce a new parameter called *flash back*for each session. The *flash back* parameter has beenused to measure the amount of postponed service that should be compensated by the BSCFQ server. Therefore, BSCFQ provides better service to bursty sessions rather than SCFQ.

An important advantage of BSCFQ vs. SCFQ is that the average delay of a burstysession can be reduced. To study the average delay in BSCFQ, we constructed a simulation model. Our simulation results demonstrate that in compare with SCFQ, BSCFQ reduces theaverage delay in bursty sessions. It is worth pointing out that QoSs of most of multimedia applications e.g. real-time video streaming depends on such a feature. An upper bound of delay will be derived in our upcoming publications.in BSCFQ scheduling algorithm.The computational complexity of our proposed algorithm, is the same as SCFQ. This is because BSCFQ only adds a subtraction function in calculating the finishing-tag and also a comparison during virtual clock computation.

REFERENCES

[1] D. Stiliadis and A. Varma, (1998),“Rate-Proportional Servers: A design methodology for fair queueing algorithms”, *IEEE/ACM Transactions on Networking*, Vol.6, No.2 , pp175-185.

[2] YaserMiaji and Suhaidi Hassan,(2009), “A survey on the chronological evolution of timestamp schedulers in packet switching networks”, in proceedings of *IC-BNMT*.

[3] C. Annadurai, (2011),“Review of packet scheduling algorithms in mobile AdHoc networks”, *International Journal of Computer Applications*,Vol. 15, No.1, February.

[4] H. Zhang and S. Keshav, (1991),“Comparison of rate-based service disciplines”, In proceedings of the *ACM Conference on Communications Architecture & Protocols*, pp113-121.

[5] P. Goyal and H. Vin., (1997), “Generalized guaranteed rate scheduling algorithms: a framework”, *IEEE/ACM Transaction on Networking*, Vol.5 , No.4, pp561–571.

[6] D. Stiliadis and A. Varma, (1998),“Latency-Rate servers: a general model for analysis of traffic scheduling algorithms”, *IEEE/ACM Transactions on Networking (ToN)*, Vol.6, No.5, pp611–624.

[7] S. Golestani, (1994),“A self-clocked fair queueing scheme for broadband applications”, In proceedings of *INFOCOM*, pp. 636–646,.

[8] A. Parekh and R. Gallager, (1993),“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), pp344–357,.

[9] D. Stiliadis and A. Varma, (1998),“Efficient fair queueing algorithms for packet-switched networks”, IEEE/ACM Transactions on Networking (TON), Vol.6, No.2, pp175-185.

[10] M. Katevenis, S. Stefanos, C. Courcoubetis, (1991),“Weighted round-robin cell multiplexing in a general-purpose ATM switch chip*”, IEEE Journal on Selected Areas in Communications*, Vol.9, No.8, pp1265-1279.

[11] K. Siriwong, L. Lipsky, and R. Ammar, (2007), “Study of bursty internet traffic”, In proceedings of *IEEE International Symposium on Network Computing and Applications*, pp. 53–60.

[12] M. M. N. Aldeer, (2010), “Performance comparison of packet-level multiplexing algorithms with bursty traffic”, *Journal of Engineering Science and Technology*, pp46-52.

[13] M Hanada, H. Nakazato, (2007),“ORC-GPS: output Rate-Controlled scheduling policy for delay guarantees”, Proceedings of *ICC*, pp530 – 535.

[14] R. Kalle, U.C. Devi, S. Kalyanaraman, (2011), “Multi-tiered, burstiness-aware bandwidth estimation and scheduling for VBR video flows”, In Proceeding of*IEEE International Workshop on Quality of Service (IWQoS)* , pp1-9.

[15] Z. Dwekat, G. N. Rouskas, (2011),“A practical fair queuing scheduler: Simplification through quantization”, *Journal of Computer Networks*, Elsevier, Vol.55,pp2392–2406.

[16] F. Sbrina,(2010), “A novel resource scheduling algorithm for QoS-aware services on the Internet”, *Journal of Computer and Electrical Engineering*, Vol.36, No.4,pp718–734.

[17] S. S. Lan and G. G. Xie, (1995), “Burst Scheduling: architecture and algorithm for switching packet video”, *IEEE/ACM Transaction on Networking*, Vol.3, No.6, pp683-690.

[18] S. J. Golestani,(1991), “Congestion-free communication in high-speed packet network”,*IEEE Transactions on Communications*, Vol.39, No.12, pp1802-1812.

[19] M. Hamadoui and P. Ramanathan,(1995), “A dynamic priority assignment technique for streams with (m, k)-firm deadlines”, *IEEE Transactions on Computers Communication*, Vol.44, No.12, pp1443-1451.

[20] F. Callegati, A. Campi, W. Cerroni, (2010),“A practical approach to scheduler implementation for optical Burst/Packet Switching”, In proceeding of *ONDM*, pp.1-6 .

[21] Z. Rosberg, Li Ji, Li Fan, M. Zukerman, (2011), “Flow Scheduling in Optical Flow Switched (OFS) Networks Under Transient Conditions” , *Journal of Light way Technology*, Vol.29, No.21, pp. 3250 – 3264.

[22] H. Zhu, G. Cao, (2004),“On improving service differentiation under bursty data traffic in wireless networks”, in Proceeding of *INFOCOM*, pp. 871-881.

[23] D. Sh. Lee, Ch. M. Chen, and Ch.-Y. Tang,(2007), “Weighted Fair Queueing and Compensation Techniques for Wireless Packet Switched Networks”, *IEEE Transactions on Vehicular Technology*, Vol. 56 , No.1), pp887–891.

[24] T.-G. Kwon, S.-H. Lee, J.-K. Rho, (1998), "Scheduling algorithm for real-time burst traffic using Dynamic Weighted Round Robin", In proceeding of *IEEE International Symposium on Circuits and Systems*, pp506-509.

[25] Young-Jin Choi, Hyung-Myung Kim,(2007), "A new scheduling scheme: for high-speed packet networks Earliest-virtual-deadline-first", *Journal of Computer Communications, Elsevier*,Vol.30 ,pp2291–2300.

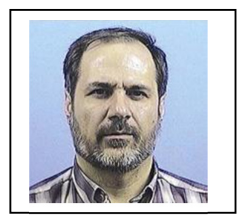
[26] T. Quynh, H. Karl, A. Wolisz, and K. Rebensburg, (2001), "Relative jitter packet scheduling for differentiated services", In proceeding of *IFIP Conference PerformanceModeling and Evaluation of ATM and IP Networks*, pp1-13.

[27] M. Yaghmaei, M. Bahekmat, and T. Khojasteh, (2008), "A Novel Fuzzy Logic Base Scheduling Mechanism for Service Differentiation in IP Networks", *Journal of Iranian Association of Electrical and Electronics Engineers*, pp. 54-65.

[28] Xiaobo Zhou, Dennis Ippoliti, Liqiang Zhang, (2008) "Fair bandwidth sharing and delay differentiation: Joint packet scheduling with buffer management",*Journal of Computer Communications, Elsevier*, Vol.31, No.17, pp4072-4080.

Authors

Mahmoud DaneshvarFarzanegan:.received his B.Sc. degree in electrical engineering from Isfahan University of Technology (IUT), in 1994 and the M.Sc. degree in electrical engineering (Communication Networks) from Amir-Kabir (Polytechnique) University of Technology, Tehran, Iran, in 1997. He joined the Electrical Engineering Islamic Azad University branch of Najafabad (IAUN)in 1999 where he is a lecturer. He is currently working toward his Ph.D. His research interests include: Packet scheduling algorithms, QoS and networks performance evaluations, Network calculus, and Broadband Power Line Communication.



HosseinSaidi: received the B.S. and M.S. degrees in electrical engineering (Electronics and Communication Eng.) from Isfahan University of Technology (IUT), in 1986 and 1989, respectively, and the Ph.D. degree in electrical engineering from the Washington University in St. Louis, MO in 1994.From 1994 to 1995, he was a research associates at Washington University St. Louis, and in 1995 he joined the Electrical and Computer Engineering of IUT, where he is an Associate Professor. His Research interest includes ATM, high speed networking, QoS guarantees, routing, algorithms and information theory.

Mehdi Mahdavi: received his B.Sc. degree in electrical (Electronics) engineering from Isfahan University of Technology, the M.Sc. degree in electrical engineering from Amir-Kabir (Polytechnique) University of Technology, Tehran, Iran, and the Ph.D. degree in electrical engineering (Mobile Communication Networks) from the University of Sheffield, Sheffield, U.K. In 2004, he joined the ICT faculty, Ministry of ICT, Tehran, where he was an Assistant Professor working on Tele-traffic theory and network quality of service in the next generation of mobile cellular and wireless networks. Since 2005, he has been with the Department of Electrical and Computer Engineering, Isfahan University of Technology, as an Assistant Professor. His research interests include tele-traffic theory, network quality of service, networks performance evaluations, network calculus, traffic source modelling and measurements, scheduling algorithms, routing algorithms and telecommunication in general.