An Extended Weighted Fair Queuing (EWFQ) algorithm for broadband applications including multicast traffic

Mudassir TUFAIL and Bernard COUSIN

IRISA, Campus de Beaulieu
35042 Rennes Cedex, France.

ABSTRACT

Ensuring end-to-end bounded delay and fair allocation of bandwidth to a backlogged session are no more the only criterias for declaring a queue service scheme good. With the evolution of packet-switched networks, more and more distributed and multimedia applications are being developed. These applications demand that service offered to them should be homogeneously distributed at all instants contrarily to back-to-back packet's serving in WFQ scheme. There are two reasons for this demand of homogeneous service:

- In feedback based congestion control algorithms, sources constantly sample the network state using the feedback from the receiver. The source modifies its emission rate in accordance to the feedback message. A reliable feedback message is only possible if the packet service is homogeneous.
- In multicast applications, where packet replication is performed at switches, replicated packets are probable to be served at different rates if service to them, at different output ports, is not homogeneous. This is not desirable for such applications as the phenomena of packet replication to different multicast branches, at a switch, has to be carried out at a homogeneous speed for the following two important reasons[1,2]: 1) heterogeneous service rates of replicated multicast packets result in different feedback informations, from different destinations (of same multicast session), and thus lead to unstable and less efficient network control. 2) in a switch architecture, the buffer requirement can be reduced if replication and serving of multicast packets are done at a homogeneous rate.

Thus, there is a need of a service discipline which not only serve the applications at no less than their guaranteed rates but also assures a homogeneous service to packets. The homogeneous service to an application may precisely be translated in terms of maintaining a good inter-packets spacing.
$EWFQ$ scheme is identical to WFQ scheme except that a packet is *stamped* with delayed value of service start time of packet in corresponding GPS scheme. This delay is meant to consider the packet slots which might be occupied by a packet of precedently served session. Then $EWFQ$ scheme serves the packets in the increasing order of their *stamp* values. It provides an end-to-end bounded delay service to applications. For multicast sessions, this scheme ensures a homogeneous service rate to all the replicated packets thus permits the replicator to work at a rather constant speed. Session's packets get distributed more accurately with low cost, moreover $EWFQ$ scheme is highly probable to perform lesser number of operations than other schemes (e.g. WF^2Q) while ensuring good inter-packets spacing.

Keywords: Weighted fair queuing, multicast, homogeneous service, integrated services, feedback control, inter-packets spacing, guaranteed service

Other author information: M. Tufail: Email: mtufail@irisa.fr; Tel: (33) 2 99 84 72 91; Fax: (33) 2 99 84 71 71
B. Cousin: Email: bcousin@irisa.fr; Tel: (33) 2 99 84 73 33; Fax: (33) 2 99 84 71 71

1. INTRODUCTION

There is a variety of distributed applications (e.g. audio and video conferencing, multimedia information retrieval, ftp, telnet, WWW, etc.) with a wide range of Quality of Service (QoS) requirements. A network meets these requirements primarily by appropriately *scheduling* its resources.
All network switches require an intelligent scheduling algorithm to select a packet from a deserving queue, among those present at an output port, at each packet slot time, where a slot is defined as:

DEFINITION 1.1. *Slot is a time interval long enough to transmit one packet completely.*
The Generalized Processor Sharing (GPS) scheme is a general form of the head-of-line processor sharing service disciplines. During any time interval, GPS scheme serves, in parallel, all the non-empty queues* in proportion to the service shares of their corresponding sessions. Obviously, GPS scheme cannot be applied to the actual packet-based traffic scenarios, where only one session can receive service at a time, and where an entire packet, must be served before another packet is picked up for the service. GPS scheme is a theoretical model so there are many propositions of queuing disciplines which emulate GPS scheme. The progress of schemes emulating GPS scheme can be presented in the following order:

- A queue service scheme is expected to provide guaranteed bounded delay services. It has been demonstrated in Refs. 3,4 that employing GPS servers at the switches, end-to-end delay can be guaranteed to a session provided its traffic is leaky bucket constrained at the source. Parekh[4] proposed Packet-by-Packet Generalized Processor Sharing ($PGPS$) which emulates the GPS server and is identical to the weighted version of Fair Queuing (WFQ). He also established several important relationships between a GPS scheme and its corresponding packet WFQ scheme:
    1. A packet will finish service in a WFQ scheme later than in the corresponding GPS scheme by no more than the transmission time of one maximum size packet. It measures how far is WFQ scheme from GPS one in terms of delay.
    2. As far as the amount of work, a session gets, is concerned, a WFQ scheme does not fall behind a corresponding GPS scheme by more than one maximum size packet.

p_i^k	the k^{th} packet on session i
a_i^k	arrival time of the k^{th} packet of session i
$d_{i,S}^k$	departure time of the k^{th} packet of session i in the S scheme
$b_{i,S}^k$	service start time of the k^{th} packet of session i in the S scheme
$stamp_i^k$	*stamp* value of the k^{th} packet of session i
L_i^k	size of the k^{th} packet of session i
$L_{i,max}$	the maximum packet size among all the packets of session i
L_{max}	the maximum packet size among all the packets of all sessions
$B(\tau)$	set of backlogged sessions at the time τ
r	link speed
r_i	guaranteed rate for session i
pos_i	position of session i in the *session order*
n	total number of packets, from all the backlogged sessions, to be transmitted at time τ
n_i	total number of packets, from a backlogged session i, to be transmitted at time τ
N	number of backlogged sessions at time τ
N_s	total number of scenarios
$dist_{i,S}^k$	inter-packets spacing in S scheme, in time units
$dist_{norm,S}$	*normalized* inter-packets spacing in S scheme

Table 1. Notations used in this paper

*There is a separate FIFO queue for each session. It is possible to have single queue for all sessions with similar QoS requirements.

- Once the schemes progressed well in satisfying the end-to-end delay bounds to sessions then the feedback based networks expected them to provide a homogeneous and uniform service trend to sessions. In most feedback based congestion control algorithms, source periodically samples the network state using feedback from the receiver or from the network, and tries to detect the symptoms of network congestion. In case of congestion, the source usually lowers the transmission rate to alleviate the congestion. WFQ scheme provides each session with their guaranteed rate but session packets are served back-to-back before packets on other sessions can be transmitted. This yields the ON(burst) and OFF(silence) zones in a session's packets transmission pattern. Obviously, with more sessions, the length of periods between burst and silence can be larger. Such oscillation is undesirable for feedback based congestion control algorithms as the feedback received by the source entirely depends upon an interval of network observation which is highly probable to differ in the very next interval. Jon C.R. Benett and Hui Zhang proposed Worst-case Fair Weighted Fair Queuing[3] in which the server does not serve the session packets back-to-back, if possible, rather the service to a session is distributed packet by packet during the server cycle. The session, still, gets its guaranteed rate and the work received by the session does not fall behind that in corresponding GPS scheme by more than one maximum packet size. For each session there are no more ON/OFF transmission zones and the feedback received by the source is more reliable which was interval dependent in previous methods of GPS scheme emulation.

Contribution: The technology progress requires networks to serve the packets, belonging to an application whether unicast or multicast, with an assurance of QoS required. This QoS is not assured by reserving the sources statically to the application rather the application's throughput is throttled up and down by feedback messages from the network. More precise is the feedback information, better the network can assure QoS to an application. Moreover multicast applications are more demanding for a precise feedback information as it affects the resource allocation to their packets (which ultimately changes the service allocation to packets of other applications) on all the replicated multicast branches. In order to have precise feedback information, it is necessary to maintain the inter-packets spacing closer to that in GPS scheme. WF^2Q scheme eliminates the ON/OFF periods for a session but does not have the capacity for maintaining a uniform inter-packet spacing as it may vary from one switch to other depending upon its implementation policy of WF^2Q scheme. Additionally, the implementation cost of WF^2Q scheme is significant as compare to the other approximations of GPS scheme.[5] We move ahead in the context of feedback based congestion/traffic controlled networks and propose a packet scheduling scheme named as Extended Weighted Fair Queuing ($EWFQ$) which eliminates the ON/OFF periods for a session with lesser number of operations than that in WF^2Q scheme and is independent of its implementation policy at a switch. Moreover, $EWFQ$ scheme is work conserving.

1.1. Relation to previous works

1.1.1. GPS scheme

The Generalized Processor Sharing (GPS) scheme uses an idealized fluid model and is served as the reference for comparing a proposed packet scheduling scheme. A GPS scheme server serving N sessions is characterized by N positive real numbers $\phi_1, \phi_2, \ldots, \phi_N$. The server operates at the fixed rate r and is work conserving. Let $W_i(t_1, t_2)$, be the amount of work, session i receives in the interval $[t_1, t_2]$, then GPS scheme sever is defined as one for which

$$\frac{W_i(t_1, t_2)}{W_j(t_1, t_2)} \geq \frac{\phi_i}{\phi_j} \tag{1}$$

holds for any session i that is backlogged[†] throughout the interval $[t_1, t_2]$. From the definition, it immediately follows that if $B(\tau)$, the set of backlogged sessions at the time τ, remains unchanged during any interval $[t_1, t_2]$, the service rate of session i during the interval will be exactly

$$r_i^*(t_1, t_2) = \frac{\phi_i}{\sum_{j \in B(t_1)} \phi_j} r \tag{2}$$

[†]A session is backlogged if it has one or more packets in the queue at the given instant.

where r is the link speed. Since $B(t_1)$ is a subset of all the sessions at the server, it is easy to see that:

$$r_i^*(t_1, t_2) \geq r_i \text{ where } r_i = \frac{\phi_i}{\sum_{j=1}^{N} \phi_j} r \tag{3}$$

Therefore session i is guaranteed a minimum service rate of r_i during any interval when it is backlogged.
With the fluid flow model of traffic, the service may be offered to sessions in arbitrarily small increments. Equivalently, it may be assumed that multiple sessions can receive service in parallel. As the result, it is possible to divide the service among the sessions, at all times, exactly in proportion to the specified service rates thus making it in-applicable to actual packets flow where only one session can receive service at a given time.

1.1.2. WF^2Q scheme

In WF^2Q scheme, when the server is ready to transmit the next packet at time τ, rather than selecting it from among all the packets at the server (as in the WFQ scheme)[‡] the server only considers the set of packets that have started

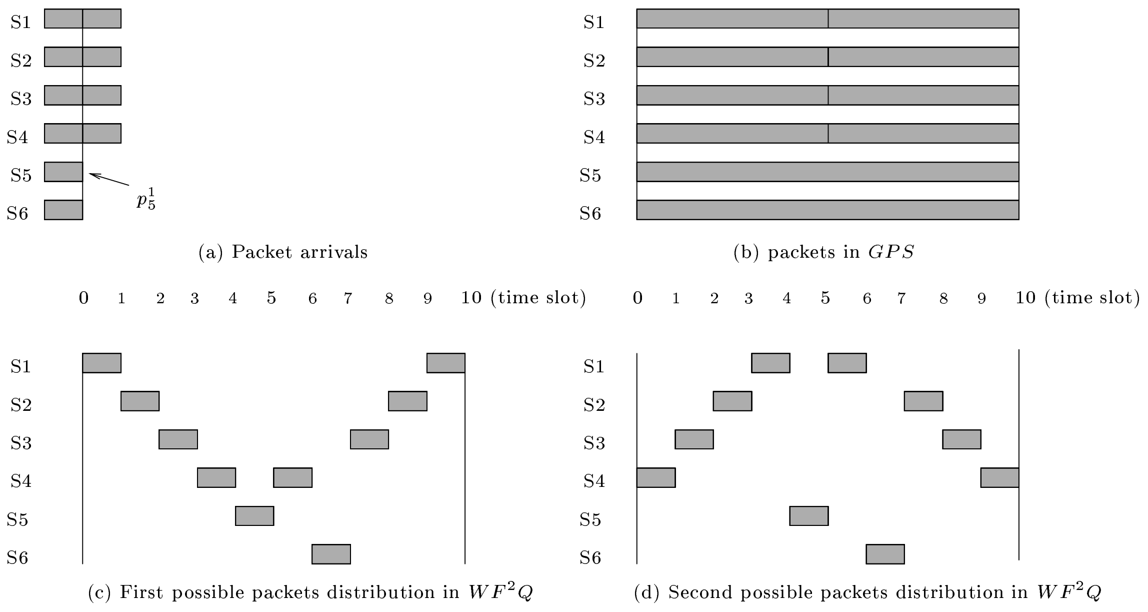
S1
S2
S3
S4
S5
S6
p_5^1
(a) Packet arrivals
(b) packets in GPS
0 1 2 3 4 5 6 7 8 9 10 (time slot)
(c) First possible packets distribution in WF^2Q
(d) Second possible packets distribution in WF^2Q

Figure 1. Possible packet's distributions in WF^2Q scheme

(and possibly finished) receiving service in the corresponding GPS scheme at time τ and picks up the packet among them that would complete service first in the corresponding GPS scheme.[3] The packets distribution by WF^2Q scheme depends upon its implementation policy at a switch. Obviously, in a real packet-switched network, different switches on a link may implement differently the WF^2Q scheme especially if they come from different vendors.
Consider the example of Fig. 1. Sessions are allocated with the following service share proportions:

$$\phi_i = 0.2, 0.2, 0.2, 0.2, 0.1, 0.1 \text{ where } i = 1, 2 \ldots, 6$$

The server speed r is 1 packet per time slot. The packets in GPS scheme depart as determined by the equation: $d_{i,GPS}^k = \frac{L_i^k}{r_i} + max(d_{i,GPS}^{k-1}, a_i^k)$ and are shown in Fig. 1.b. For simplicity, we assume, in Fig. 1, that all packets are of same size.
There are many possible service distributions to packets for a given scenario in WF^2Q scheme, where a scenario is defined in the following.

[‡]Recall that in a WFQ scheme, when the server chooses the next packet for transmission at the the time τ, it selects among all the packets that are backlogged at τ, the first packet that would finish service in the corresponding GPS scheme, if no additional packets were to arrive after time τ.

Definition 1.2. *A scenario constitutes a set of sessions with specific service share values and with specific server speed.*
In other words, at the given instant there may be more than one packet which satisfy the WF^2Q's principles. Hence, packets distribution depends upon the implementation policy of WF^2Q scheme, which may vary, considerably, the inter-packet spacing from one implementation to other. At time $\tau = 0$, there are four valid packets (p_1^1, p_2^1, p_3^1 and $, p_4^1$) as they all have the same service finish time in corresponding GPS scheme and have started receiving service at time $\tau = 0$ in corresponding GPS scheme. So a random selection of a packet among the valid ones may result in different packets distribution. Refer to Fig. 1.c and Fig. 1.d and observe these packets distribution in two possible implementations of WF^2Q scheme. A similar situation is also observed at time $\tau = 5$ where there are more than one valid packets and packets distribution may differ. Observe that there is a considerable variation in inter-packet spacing for session 1 in Fig. 1.c and Fig. 1.d which are two packets distribution by WF^2Q scheme among the possible ones.
In this article we present Extended Weighted Fair Queuing ($EWFQ$) scheme which maintains good inter-packet spacing and is work-conserving. $EWFQ$ is an advanced version of Weighted Fair Queuing (WFQ). It eliminates the back-to-back packet flow, if possible, thus avoids the ON/OFF periods of packet transmission. We will consider Self-Clocked Fair Queuing $SCFQ$ scheme, proposed by Golestani,[6] for emulating the WFQ part of our proposed $EWFQ$ scheme. $SCFQ$ eliminates the need of simulating events in the hypothetical GPS scheme, at every instant τ, by generating a *virtual time* which is extracted from the packet being served at the moment, thus lowers the computational complexity considerably.

2. EXTENDED WEIGHTED FAIR QUEUING SCHEME

In Extended Weighted Fair Queuing ($EWFQ$) scheme, we develop a *session order* and a *service order* for all the sessions, whether backlogged or not, which are guaranteed a non-zero bandwidth share. The *session order* is represented by pos_i which indicates the position of session i, in the decreasing order of service shares, among all sessions which are supposed to share the available bandwidth. It means that among N sessions, session i attributed with $pos_i = 1$ has the largest service share where as another session j with $pos_j = N$ has the minimum service share.
The *service order* is the order with which the sessions are served by $EWFQ$ server. Before constructing the *service order*, we *stamp* each packet assuming that all the sessions are backlogged, regardless of their actual status. Packets of session i, whose $pos_i = 1$, are *stamped* with their respective $b_{i,GPS}^k$ values. In order to avoid back-to-back transmission of packets belonging to a session, packets of any other session i (i.e. $pos_i > 1$), instead of being *stamped* with $b_{i,GPS}^k$ values, are *stamped* with $b_{i,GPS}^k$ values delayed by time slots which might be occupied by a packet of precedent session in the *session order*. This act of delaying $b_{i,GPS}^k$ values is based on the fact that packets in a real system, contrarily to GPS scheme, are served one after other requiring the *stamp* values be deplaced accordingly. Moreover this *stamping* policy ensures a good inter-packet spacing. The *service order* is, then, constructed by arranging the packets in increasing order of their *stamp* values. Before moving further, we would like to define a server cycle as:

Definition 2.1. *A cycle in the minimum time interval sufficiently long enough to transmit as many packets, of all the sessions, of a scenario, with non-zero service shares, as to assure each session with, at least, its guaranteed service share, provided that each packet is fully transmitted.*
Cycle length may vary with variable packet's size. In this article, we assume packets of same length thus rending cycle computations simpler. The *service order* has the following characteristics:

- It is independent of the fact that one or more sessions is not backlogged at the given instant.
- The *service order* once constructed stays valid unless there is a change in any session's service share (ϕ_i).
- The *service order* is independent of the instant τ at which it is being consulted.
- It repeats itself after every server cycle.

The *service order* avoids scheduling all the packets present at the given instant (WFQ and WF^2Q schemes do so) rather it helps the scheduler to select the packet for service among those present at the given instant. At first, a pointer is placed on the first slot of the *service order*. The slot represented by the pointer indicates the session whose packet (present at the head of session's queue) is to be served at the given time. The pointer is, then, moved forward to the next slot in the *service order* which, when consulted at the next packet slot time, indicates the session to be served. If at a given instant the session indicated by the pointer of the *service order* is not backlogged then pointer is moved forward till it points to a slot of the *service order* representing a backlogged session at the given instant[§]. Obviously other backlogged sessions get more than their guaranteed share.

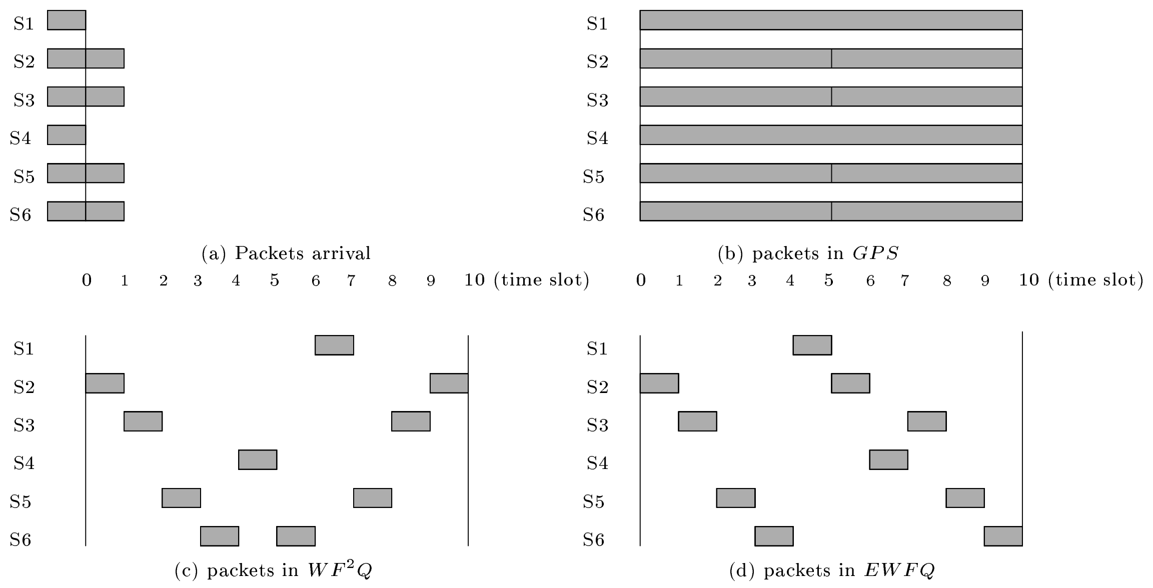

Figure 2. An example of *EWFQ*

2.1. Algorithm

1. Create *session order*: Arrange sessions in decreasing order of their respective service shares (ϕ_i) and attribute them pos_i such that $\phi_{i_1} \geq \phi_{i_2} \geq \ldots \geq \phi_{i_N}$ which implies that $pos_{i_k} = k$.

2. Calculate $b^k_{i,GPS}$ for all packets as:
$$b^k_{i,GPS} = max(a^k_i, d^{k-1}_{i,GPS}) \tag{4}$$

3. Stamp each packet as:
$$stamp^k_i = \begin{cases} b^k_{i,GPS} & \text{if } pos_i = 1 \\ b^k_{i,GPS} + \sum_{l|pos_l=1}^{l|pos_l=pos_i-1} \frac{L_{l,max}}{r} & \text{else} \end{cases}$$

4. Create *service order*: Arrange packets in increasing order of their *stamp* values.
   - If two or more packets have the same *stamp* value, arrange them in increasing order of their respective session's pos_i value.

Example: Consider the example shown in Fig. 2. There are six sessions sharing the bandwidth of link server and their respective service share values are $\phi_i = 0.1, 0.2, 0.2, 0.1, 0.2, 0.2$ where $i = 1, 2, \ldots, 6$. The server speed r is 1 packet per time slot and one server cycle measures 10 time slots in the example. First the sessions are arranged in the decreasing order of their respective service share values as: $S_2, S_3, S_5, S_6, S_1, S_4$, which makes their respective *session order* pos_i values as: $1, 2, 3, 4, 5, 6$. As the sessions are supposed to be backlogged so they are shown, in Fig. 2.a disposing enough packets at $\tau = 0$ to get their bandwidth share in one server cycle (i.e. 10 time slots). Following the

[§]The guaranteed share of bandwidth, which is not being used by an non-backlogged session (at the given instant), may be utilized for serving the packets of a session with zero service share e.g. best effort traffic.

		$b_{i,GPS}^k$		$stamp_i^k$	
pos_i	session i	$k=0$	$k=1$	$k=0$	$k=1$
1	2	0	5	0	5
2	3	0	5	1	6
3	5	0	5	2	7
4	6	0	5	3	8
5	1	0		4	
6	4	0		5	

Table 2. Calculating *stamp* values for each packet

$EWFQ$ algorithm, we calculate $b_{i,GPS}^k$, the service start time for a packet p_i^k in corresponding GPS scheme, then *stamp* each packet accordingly. The packets of session i, whose $pos_i = 1$, are *stamped* with their respective $b_{i,GPS}^k$ values. For any other session i (i.e. whose $pos_i > 1$), packets are *stamped* with respective $b_{i,GPS}^k$ values delayed by 1 (recall that all packets are assumed, in the article, to be of equal size which implies that $\frac{L_{l,max}}{r} = 1$) than those of its precedent session l in *session order*. These values are shown in table 2. The *service order* is constructed by arranging the packets in increasing order of their *stamp* values and is shown in Fig. 3. The packets p_2^2 and p_4^1 have the same *stamp* value (i.e. $stamp_2^2 = stamp_4^1 = 5$). We arrange them in increasing order of their respective session's pos_i values ($pos_2 < pos_4$) which makes session 2 to come earlier than session 4 in *service order*. The $EWFQ$ scheme

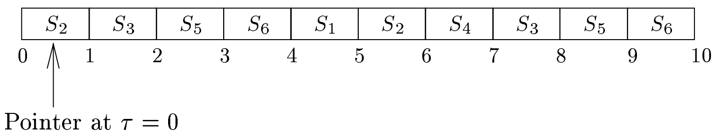

Figure 3. The *service order.*

server consults the *service order* at every packet slot time and transmits the packet present at the head of queue of the session indicated by the *service order*. Refer to Fig. 2.d for the packets progress in $EWFQ$ scheme.
Note that $EWFQ$ scheme is independent of its implementation policy (contrarily to WF^2Q scheme) as there is only one packet which is declared eligible for service at a given instant.

3. ANALYSIS OF $EWFQ$ SCHEME

LEMMA 3.1. *If all sessions, of a given scenario, are backlogged then* $stamp_i^k$ *value, associated to any packet* p_i^k, *in EWFQ scheme will always be smaller than its service finish time,* $d_{i,GPS}^k$, *in the corresponding GPS scheme.*
It is to prove that:

$$stamp_i^k < d_{i,GPS}^k \tag{5}$$

For packets of session i, whose $pos_i = 1$, $stamp_i^k = b_{i,GPS}^k$ as per $EWFQ$ scheme principles. Since $d_{i,GPS}^k = b_{i,GPS}^k + \frac{L_i^k}{r_i}$ thus a packet, p_i^k $\forall k$ and $pos_i = 1$, always satisfies the relation 5.
For all other sessions ($pos_i > 1$), packets are *stamped* as:

$$stamp_i^k = b_{i,GPS}^k + \sum_{l \| pos_l = 1}^{l \| pos_l = pos_i - 1} \frac{L_{l,max}}{r} \tag{6}$$

Since all the packets are assumed to be of same size then Eq. 6 becomes:

$$stamp_i^k = b_{i,GPS}^k + (pos_i - 1)\frac{L}{r} \qquad\qquad L_i^k = L \quad \forall i,k \tag{7}$$

For a packet p_i^k of session i ($pos_i > 1$), the service finish time corresponding GPS scheme is written as:

$$d_{i,GPS}^k = b_{i,GPS}^k + \frac{L_i^k}{r_i} = b_{i,GPS}^k + \frac{L}{r_i} \tag{8}$$

Calculating $b_{i,GPS}^k$ from Eq. 8 rewriting it in Eq. 7 for a packet of session i ($pos_i > 1$), we get:

$$stamp_i^k = d_{i,GPS}^k - \frac{L}{r_i} + (pos_i - 1)\frac{L}{r} \tag{9}$$

Putting $stamp_i^k$ value from Eq. 9 in Eq. 5 we get:

$$\begin{aligned} (pos_i - 1)\frac{L}{r} &< \frac{L}{r_i} & (10)\\ r &> (pos_i - 1) * r_i & (11) \end{aligned}$$

Thus proving Eq. 11 validates the relation 5. We know that $r = \sum_{j=1}^{N} r_j$ which implies that $r \geq \sum_{j=1}^{i} r_j$. In the *session order* we have $\phi_{i_1} \geq \phi_{i_2} \geq \ldots \geq \phi_{i_N}$ with $pos_{i_k} = k$ which implies that $r_{i_1} \geq r_{i_2} \geq \ldots \geq r_{i_N}$, so it can be written as:

$$r \geq pos_i * r_i \Rightarrow r > (pos_i - 1) * r_i \tag{12}$$

The proof of Eq. 12 validates the relation $stamp_i^k < d_{i,GPS}^k$.

COROLLARY 1. *In EWFQ scheme, the $stamp_i^k$ value associated to a packet p_i^k always satisfies the following:*

$$stamp_i^k \leq d_{i,GPS}^k - \frac{L_i^k}{r} \tag{13}$$

Modifying Eq. 9, we get

$$\begin{aligned} stamp_i^k &= d_{i,GPS}^k - \frac{L}{r} + pos_i * \frac{L}{r} - \frac{L}{r_i} & (14)\\ stamp_i^k &= d_{i,GPS}^k - \frac{L}{r} + L * \frac{pos_i * r_i - r}{r * r_i} & (15) \end{aligned}$$

Knowing that $r \geq pos_i * r_i$ (refer Eq. 12) we can deduce from Eq. 15 that:

$$stamp_i^k \leq d_{i,GPS}^k - \frac{L}{r} \tag{16}$$

Since $L_i^k = L \; \forall i, k$ thus above Eq. 16 proves the relation 13.
These results may also be verified for scenarios having variable packets size.

3.1. Simulation results

Proposed $EWFQ$ scheme is destined to serve the packets of a session with a good inter-packets spacing. Among the earlier proposed queue service schemes, WF^2Q scheme has been developed for the same goal. In this section, we present the comparative study of $EWFQ$ and WF^2Q schemes on the basis of resulting inter-packets spacing in two schemes.

DEFINITION 3.2. *We define the inter-packets spacing for a session, served by server of S scheme, as the difference between the service start times of two consecutive packets of the session. For a packet p_i^k, the inter-packet spacing, $dist_{i,S}^k$, is given as:*

$$dist_{i,S}^k = b_{i,S}^k - b_{i,S}^{k-1} \tag{17}$$

In GPS scheme, $b_{i,GPS}^k = max(a_i^k, d_{i,GPS}^{k-1})$. If for a packet p_i^k of a backlogged session i, $b_{i,GPS}^k = d_{i,GPS}^{k-1}$, then $dist_{i,GPS}^k$ for p_i^k is given as:

$$dist_{i,GPS}^k = b_{i,GPS}^k - b_{i,GPS}^{k-1} = \frac{L_i^{k-1}}{r_i} \tag{18}$$

Since inter-packets spacing is session dependent and moreover it is given in absolute time, so it should be *normalized* for a reliable comparative study. We define, in the following, the different quality measuring parameters to determine a scheme's performance.

DEFINITION 3.3. *Quality variation for a scenario, in S scheme, is given by:*

$$quality\ variation = \frac{1}{N}\sum_{i=1}^{i=N}\frac{1}{n_i}\sum_{k=1}^{k=n_i}\frac{|dist_{i,GPS}^k - dist_{i,S}^k|}{dist_{i,GPS}^k} \tag{19}$$

The Eq. 19 gives a mean value of normalized difference of inter-packets spacings from corresponding values in GPS scheme for all the packets in all the sessions. Note that an equal weight is given to all sessions, regardless of their respective service share, in measuring the quality variation for a scenario in S scheme. This permits every session to contribute equally in quality variation measure and, thus, makes it more realistic.

DEFINITION 3.4. *Normalized inter-packets spacing, $dist_{norm,S}$, for a session i $\forall i$ served by S scheme server, is given by:*

$$dist_{norm,S} = \frac{dist_{i,S}^k}{dist_{i,GPS}^k} \tag{20}$$

DEFINITION 3.5. *Quality distribution of a S scheme, for a given set of scenarios (i.e. a simulation testbed), is a distribution of normalized inter-packets spacings of all the packets in all the scenarios.*
Eq. 20 is used to measure the quality distribution of a S scheme. The mean value of quality distribution of GPS scheme is 1 whereas its standard variation is zero. Naturally, a scheme, maintaining a good inter-packets spacing, is expected to have lesser standard deviation in quality distribution with its mean value closer to 1¶.
We have simulated $EWFQ$ and WF^2Q schemes for different scenarios of a testbed. In the testbed, different scenarios are constructed by testing all possible service shares (ϕ_i's), a session i can have with in these limits: $\frac{1}{26} \leq \phi_i \leq \frac{1}{2}$ $\forall i$. We take six sessions, which are backlogged at the instant $\tau = 0$.

3.1.1. Quality variation

We calculate quality variation of all scenarios of the simulation testbed. These measurements are taken for $EWFQ$ scheme as well as for WF^2Q scheme. In WF^2Q scheme, there are many possible packet's distributions for a given scenario. For each scenario, we calculate an averaged performance of two possible fairly apart, in terms of inter-

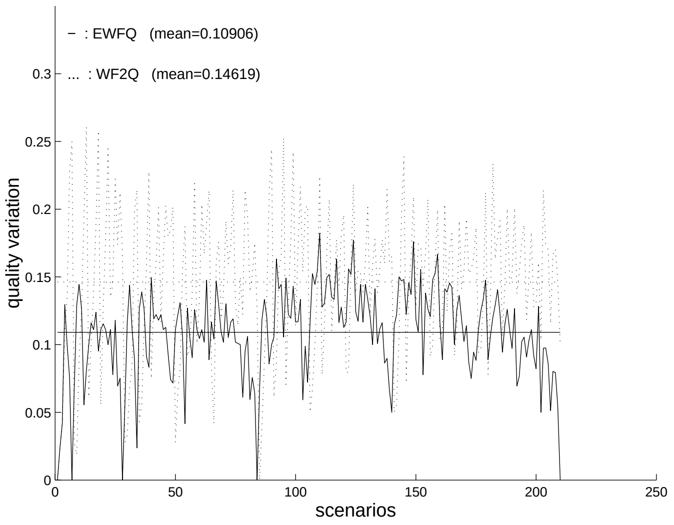

Figure 4. Quality variation in $EWFQ$ and WF^2Q schemes.

¶$dist_{norm,S}$ is 1 in an ideal packet's service scheme.

packets spacing, packet's distributions by WF^2Q scheme and then calculate the corresponding quality variation. In Fig. 4, a scenario's quality variation, in two schemes, is shown. As explained earlier, curve for WF^2Q scheme represents its average performance. We calculate the mean quality variation for both schemes. This mean value is the ratio of sum of quality measures of all the scenarios to the total number of scenarios. It is shown in Fig. 4, numerically as well as graphically (with straight horizontal line) for the two schemes.
The mean quality variation of $EWFQ$ scheme is lesser than that of WF^2Q scheme. In other words, $EWFQ$ scheme ensures a quality variation closer to zero thus more efficient in maintaining a good inter-packet spacing. It can be observed, for certain scenarios, that there is significant difference between the quality variations in two schemes.

3.1.2. Quality distribution

For each scenario in simulation, we *normalize* inter-packets spacings, using Eq. 20, generated by $EWFQ$ and WF^2Q schemes. Note that there are as many $dist_{norm,S}$ values as total number of packets for a given scenario in S scheme. This way we calculate $dist_{norm,EWFQ}$ and $dist_{norm,WF^2Q}$ for all the scenarios of the testbed. Recall that for GPS

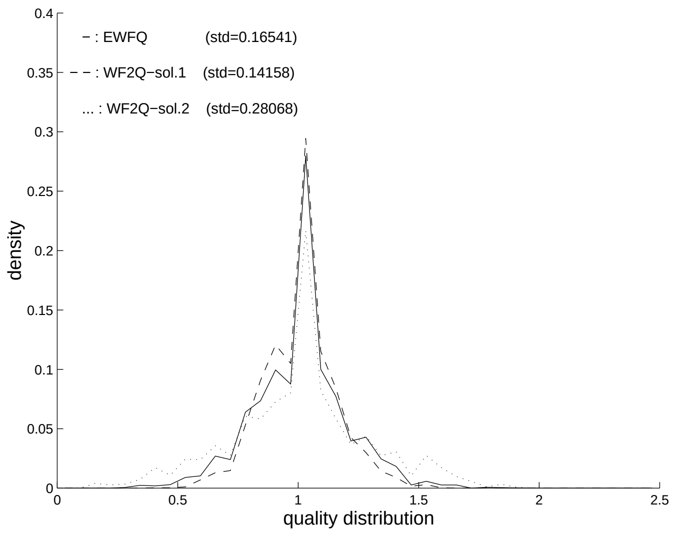

Figure 5. Quality distribution in $EWFQ$ and WF^2Q schemes.

scheme the quality distribution is always one. We divide the quality distribution scale, represented on x-axis of Fig. 5, into small intervals[‖]. For each interval, we count the number of quality distributions, whose values fall within the interval, in all the scenarios of the testbed. This number of quality distributions per interval is then converted into a proportional figure, termed as density. The density, shown on the y-axis of Fig. 5, is the ratio of number of quality distributions per interval to total number of quality distributions in all scenarios of a testbed. Fig. 5 also displays standard deviation values, represented by 'std', of quality distribution values in $EWFQ$ and WF^2Q schemes. Lesser is the 'std' value, better is the scheme in maintaining good inter-packets spacing. Standard deviation value for S scheme is calculated as:

$$std = \sqrt{\frac{1}{N_s}\sum_{l=1}^{l=N_s}\frac{1}{N}\sum_{i=1}^{i=N}\frac{1}{n_i}\sum_{k=1}^{k=n_i}\left(\frac{dist_{i,GPS}^{k} - dist_{i,S}^{k}}{dist_{i,GPS}^{k}}\right)^2} \tag{21}$$

Note that the standard deviation (refer to Eq. 21) is variance of a scheme's quality distribution from the quality distribution of GPS scheme[**] (which is always one), thus making it more realistic.
We have simulated two possible packet's distributions by WF^2Q scheme for all scenarios of the testbed. These two solutions of WF^2Q scheme are shown as WF^2Q-sol.1 and WF^2Q-sol.2 in Fig. 5. WF^2Q-sol.1 is among the good possible packet distributions (may be the best one for certain scenarios) by WF^2Q scheme, in terms of inter-packets

[‖]Taking intervals of very small size, though present more precisely the results, makes the curve abundantly fluctuating thus rending it difficult to understand.
[**]Normally standard deviation is measured in terms of data variance from their mean value.

spacing, which a scenario may experience but with a very low probability. In order to ensure it on all switches on a link, we need to add certain conditions (e.g. creating a *session order* in WF^2Q scheme), which increases, further, the operational cost of already costlier WF^2Q scheme.[5] WF^2Q-sol.2 is another possible packet's distribution by WF^2Q scheme.
By comparing the 'std' values for three curves of Fig. 5, we find that 'std' value of $EWFQ$ scheme is much closer to that of WF^2Q-sol.1 than that of WF^2Q-sol.2 which shows that $EWFQ$ scheme ensures a good inter-packets spacing which, being independent of its implementation policy, is guaranteed on all the switches on a link. The performance curves and standard deviation values show that $EWFQ$ scheme, though being lesser complex, is better in service distribution to packet's of a session and ensures a good inter-packets spacing.

4. CONCLUSION

We propose $EWFQ$ scheme which, in addition to providing the guaranteed bounded delay service, has the following important properties.

4.1. Inter-packets spacing

The simulation results in section 3.1 shows that $EWFQ$ scheme ensures a good inter-packets spacing and may be recommended for packet-switched based networks where the sessions' rate are feedback controlled. WFQ and $EWFQ$ are identical except their packet *stamping* strategy. The idea behind the packet's *stamping* with delayed values of $b^k_{i,GPS}$, in $EWFQ$ scheme, is to take care of packet slots which might be occupied by a packet of precedently served session thus making a more realistic order in *stamp* values. The *session order*, where sessions are arranged in decreasing order of their service proportions, takes care of the fact that the sessions with large share values are more sensible to inter-packets spacing.

4.1.1. Complexity and implementation issue

As said earlier that $EWFQ$ and WFQ schemes are identical except their packet's *stamping* strategy. In other wards, $EWFQ$ scheme has a WFQ part. This WFQ part of the $EWFQ$ scheme can be efficiently implemented by Self-Clocked Fair Queuing ($SCFQ$) scheme,[6] proposed by Golestani. $SCFQ$ scheme can be implemented with a complexity of order O(log N) and has very low computational cost[††], refer to Ref. 5. As for the *session order* and the *service order*, they, once constructed, remain valid till service shares of sessions are modified. These modifications do not occur very often in a real packet-switched networks thus reducing considerably the probable increase in computational cost of $EWFQ$ scheme.

REFERENCES

1. M. Tufail and B. Cousin, "Proposing a new queue service scheme for ABR multicast flow," in *Proc. International Conference on Parallel and Distributed Computing and Networks (to appear), IASTED* , Aug. 1997.
2. M. Tufail and B. Cousin, "Timer Imposed and Priority Supported (TIPS) congestion control scheme for point-to-multipoint connections in ATM," in *Large Bande:Internet ou RNIS*, G. Pujolle, ed., *Proc.* 11^{th} *Conference of De nouvelle Architecture pour les Communications (DNAC)* , pp. 65–79, Dec. 1996.
3. H. Z. Jon C.R. Bennett, "WF^2Q: Worst-case Fair Weighted Fair Queuing," *IEEE INFOCOM* , pp. 120–128, Mar. 1996.
4. A. Parekh, *A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks*, Ph.D. dissertion, Massachusetts Institute of Technology, Massachusetts, 1992.
5. F. Toutain, *Gestion préemtive et équitable de la qualité de service dans les réseaux de paquet à intégration de services*, Ph. D. dissertation, pp. 98, University of Rennes 1, Rennes, 1997.
6. J. Golestani, "A Self-Clocked Fair Queuing Scheme for Broadband Applications," *IEEE INFOCOM* , pp. 636–646, Jun. 1994.

[††]The number of operations per packet required by $SCFQ$ scheme is as low as 9 for 100 sessions arranged in multiple queues which, for WF^2Q scheme, is 36 for same set of sessions.[5]