EWFQ: Enhanced Weighted Fair Queuing Scheme ensuring accurate inter-packets spacing

Mudassir TUFAIL and Bernard COUSIN
IRISA, Campus de Beaulieu
35042 Rennes Cedex, France.
{mtufail, bcousin}@irisa.fr
tel: (33) 2 99 84 71 00
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 a time interval long enough to transmit one packet. 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 [1] 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 scheme is expected to provide guaranteed bounded delay services. It has been demonstrated in [**?**, **?**] 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 [**?**] 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.

- 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 packet transmission pattern. Obviously, with more sessions, the length of periods between bursting and silence can be larger. Such oscillation is undesirable for feedback based congestion control

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

a_i^k	arrival time of the k^{th} packet on session i
$d_{i,s}^k$	departure time of the k^{th} packet on session i in the s scheme
$b_{i,s}^k$	service start time of the k^{th} packet on session i in the s scheme
$W_{i,s}(t_1, t_2)$	the amount of work received by the session i during the time interval $[t_1, t_2]$ in the s scheme
$Q_{i,s}(\tau)$	the queue size of the session i at time τ in the s scheme
L_i^k	size of the k^{th} packet on session i in number of bits
L_{max}	the maximum packet size in number of bits
$B_s(\tau)$	the set of backlogged sessions at the time τ in the s scheme
r	link speed
r_i	guaranteed rate for session i
$\hat{}$	any notation with $\hat{}$ as overhead represents its *virtual* value

Table 1: Notations used in this paper

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 [**?**] in which the server does not serve the session packets back to back 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.

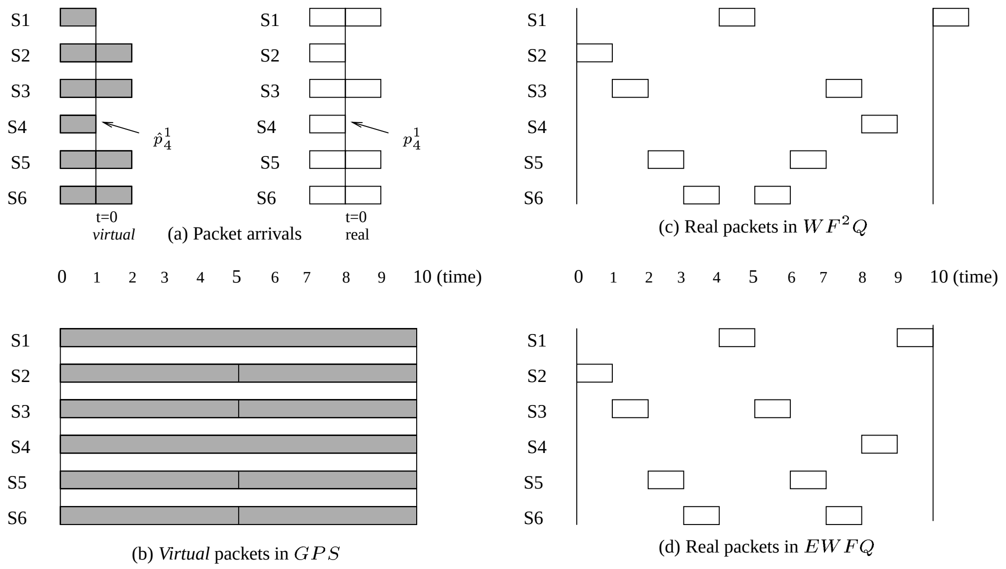
Contribution: The technology progress requires networks to serve the packets, belonging to an application whether unicast or multicast, with an assurance of QoS required. This QoS is not assured by reserving the sources statically to the application rather the application's throughput is throttled up 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 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. Additionally the work conserving property of WF^2Q scheme depends upon the status of backlogged[2] sessions (under/over-utilization of guaranteed resources). We move ahead in the context of feedback based congestion/traffic controlled networks and propose a packet scheduling scheme named as Enhanced Weighted Fair Queuing ($EWFQ$) which has the same complexity as that of WF^2Q scheme but an additional capacity of maintaining inter-packets spacing closer to that in GPS scheme with lesser number of operations. The work conserving property of $EWFQ$ is independent of session's status.

2 Enhanced Weighted Fair Queuing scheme

In Enhanced 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 n_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 *session order* $n_i = 1$ has the largest service share where as one with $n_i = N$ has the minimum service share.

Algorithm

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



S1
S2
S3
S4
S5
S6
$\hat{p}_4^1$
p_4^1
t=0
virtual
t=0
real
(a) Packet arrivals
(b) *Virtual* packets in GPS
(c) Real packets in WF^2Q
(d) Real packets in $EWFQ$
0 1 2 3 4 5 6 7 8 9 10 (time)

Figure 1: An example of *EWFQ*

1. Calculate $\hat{b}^k_{i,GPS}$ for all virtual packets as:

$$\hat{b}^k_{i,GPS} = \begin{cases} 0 & : \quad k = 0 \\ \hat{b}^{k-1}_{i,GPS} + \frac{\hat{L}^k_i}{r_i} & : \quad k > 0 \end{cases}$$

2. Creating *session order*: Arrange sessions in decreasing order of their service share r_i and attribute them n_i such that $\{r_i \parallel n_i = 1\} \geq \{r_i \parallel n_i = 2\} \geq \ldots \{r_i \parallel n_i = N\}$.

3. Stamp each *virtual* packet as:

$$stamp^k_i = \hat{b}^k_{i,GPS} + (n_i - 1) \tag{1}$$

4. Creating *service order*: Arrange packets are in increasing order of their $stamp$ values.

   - If two or more packets have the same $stamp$ value, arrange them in decreasing order of their respective session's n_i value.

The *service order* is the order with which the sessions are served by $EWFQ$ server. In order to construct the *service order*, we consider a virtually assumed scenario of packets arrival (fig.**??**,a) in which and all the sessions, in competition, have enough packets, referred as *virtual* packets hereafter, in their respective queues (regardless of the number of packets actually present in the queue) so that the session gets its guaranteed share in one server cycle with all the *virtual* packets fully transmitted. All the *virtual* packets have unit size i.e. $\hat{L}^k_i$=1 which means that there will be as many packet slots in the *service order* as total number of *virtual* packets in all the sessions. For allocating the slots of the *service order* among the sessions, we start with the session i attributed with $n_i = 1$ (i.e. having the largest bandwidth share) and allocate its *virtual* packet $\hat{p}^k_i$ a slot in the *service order* bounded by:

$$\hat{b}^1_{i,GPS} \leq \hat{b}^1_{i,EWFQ} < \frac{\hat{L}^1_i}{r_i} + \frac{1}{r} \tag{2}$$

$$\hat{b}^{k-1}_{i,EWFQ} + \frac{\hat{L}^{k-1}_i}{r_i} - \frac{2}{r} < \hat{b}^k_{i,EWFQ} < \hat{b}^{k-1}_{i,EWFQ} + \frac{\hat{L}^{k-1}_i}{r_i} + \frac{2}{r} \quad \text{for } k > 1 \tag{3}$$

The above relations specify the bounds for $EWFQ$ server to look for the appropriate slot for $\hat{p}^k_i$ among the unallocated ones in the *service order*, thus reduce the sorting computations. The unallocated slot which is the most closest to $\hat{b}^k_{i,GPS}$ and falls within the bounds specified by (**??**, **??**) is finally allocated to $\hat{p}_i^k$. After having allocated slots to all the *virtual* packets of session i, the next session attributed with $n_i = 2$ is picked up and its *virtual* packets are allocated slots in the *service order* in similar fashion. The process is repeated till the session with $n_i = N$ is allocated slots in the *service order*. The *service order* has the following characteristics:

- It is independent of the fact that one or more sessions is silent at the given instant.
- The *service order* once calculated stays valid unless there is a change in any session's bandwidth share or the set of sessions sharing the bandwidth is altered.
- The *service order* is independent of the instant 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 $t = 0$, 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 $t = 0$. 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.

Example: Consider the example shown in figure **??**. There are six sessions sharing the bandwidth of link server and their respective share values are $r_i = 1, 2, 2, 1, 2, 2$. The server speed is $r = 10$. First the sessions are arranged in the decreasing order of their respective service share i.e. $S_2, S_3, S_5, S_6, S_1, S_4$, which makes their respective *session order* n_i values as: $1, 2, 3, 4, 5, 6$. Following the $EWFQ$ principles, we get the *service order* as shown in fig(**??**). The $EWFQ$ server consults

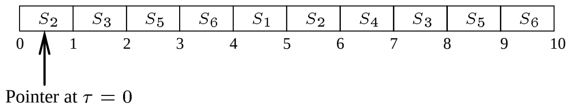


Figure 2: The *service order*.

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(**??**,d) for the packets progress in $EWFQ$ scheme. Note that the scheme WF^2Q (refer to fig(**??**,c)) is no more work conserving in this case.

3 Conclusion

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

- **Inter-packets spacing:** We define the inter-packets spacing for a session, served by server of scheme S, 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:

$$dist_{i,S}^k = b_{i,S}^k - b_{i,S}^{k-1} \tag{4}$$

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

$$dist_{i,GPS}^k = b_{i,GPS}^k - b_{i,GPS}^{k-1} = \frac{L_i^{k-1}}{r_i} \tag{5}$$

We define a parameter $Delta_{i,S}^k$, for a packet p_i^k, which measures that how much the inter-packet spacing in the scheme S differ from that in GPS system. It is calculated as:

$$Delta_{i,S}^k = dist_{i,S}^k - dist_{i,GPS}^k \tag{6}$$

$Delta_{i,S}^k$ can be positive or negative. Lesser is the absolute $Delta_{i,S}^k$ value, better the scheme S emulates the GPS scheme.

Inter-packet spacing in WFQ system: In a WFQ system, when the server chooses the next packet for transmission at time τ, it selects. among all the packets that are backlogged at τ, the first packet that would complete service in the corresponding GPS system. In other words packets are served in the increasing order of respective $d\,_{i,GPS}^{k}$ values. A packet can leave much earlier in a WFQ system than in a GPS system. The earliest possible service start time of a packet p_i^k in WFQ system is given by $b_{i,GPS}^k - ((1-\frac{r_i}{r})r - \frac{L_i^k}{r})$ where as a packet should get start of service no later than $b_{i,GPS}^k + \frac{L_i^k}{r_i} - \frac{L_i^k}{r}$. Thus we have:

$$\begin{aligned}
max_k(dist_{i,WFQ}^k) &= (b_{i,GPS}^k + \frac{L_i^k}{r_i} - \frac{L_i^k}{r}) - (b_{i,GPS}^{k-1} - ((1-\frac{r_i}{r})r - \frac{L_i^{k-1}}{r})) && (7)\\
&= \frac{L_i^{k-1}}{r_i} + \frac{L_i^k}{r_i} + \frac{1}{r}(L_i^{k-1} - L_i^k - (1-\frac{r_i}{r})r && (8)\\
max_k(Delta_{i,WFQ}^k) &= \frac{L_i^k}{r_i} + \frac{1}{r}(L_i^{k-1} - L_i^k - (1-\frac{r_i}{r})r && (9)
\end{aligned}$$

Inter-packet spacing in WF^2Q system: In WF^2Q system, when the server is ready to transmit the next packet at time τ, the server only considers the set of packets that have started (and possibly finished) receiving service in the corresponding GPS system at time τ and picks up the packet among them that would complete service first in the corresponding GPS system. So a packet p_i^k may be served as late as at $b_{i,GPS}^k + \frac{L_i^k}{r_i} - \frac{L_i^k}{r}$ and as early as at $b_{i,GPS}^k$ and still respects the WF^2Q scheme principles. Thus we have:

$$\begin{aligned}
max_k(dist_{i,WF^2Q}^k) &= (b_{i,GPS}^k + \frac{L_i^k}{r_i} - \frac{L_i^k}{r}) - b_{i,GPS}^{k-1} && (10)\\
&= L_i^k(\frac{1}{r_i} - \frac{1}{r}) + \frac{L_i^{k-1}}{r_i} && (11)\\
max_k(Delta_{i,WF^2Q}^k) &= (L_i^k(\frac{1}{r_i} - \frac{1}{r}) + \frac{L_i^{k-1}}{r_i}) - \frac{L_i^{k-1}}{r_i} && (12)\\
&= L_i^k(\frac{1}{r_i} - \frac{1}{r}) && (13)
\end{aligned}$$

Whereas for $EWFQ$ scheme, the same value for packet p_i^k is bounded as:

$$max_k(Delta_{i,EWFQ}^k) = \frac{1}{r_i}(L_i^k - L_i^{k-1}) + \frac{1}{r}(L_{max} - L_i^k) \tag{14}$$

The results (**??**, **??**) shows that $EWFQ$ scheme maintains a better inter-packets spacing than the WF^2Q scheme.

- **Complexity:** In the $EWFQ$ scheme, we construct a *session order* attributing n_i to each session which represents its position in the decreasing order of the service shares of sessions. The inherited complexity of sorting a list of N sessions is O(log N). The calculations, required to generate the *service order*, do not require any list sorting and each slot in the *service order* is attributed to a session independently hence O(1) be the complexity of this operation. The $EWFQ$ scheme has, globally, a complexity of O(log N) which is the same as in WF^2Q. Since the probability of change in the *service order* at each packet arrival is very low, so most of the time the $EWFQ$ scheme does not need to construct the *service order* at each packet slot thus reducing the number of operations considerably. Note that in WF^2Q, similar operations are carried out at each packet slot.

- **Work conserving:** The work conserving property of $EWFQ$ scheme is independent of the status of backlogged sessions which is also manifested by the WFQ scheme [**?**]. The WF^2Q scheme does not hold this property when there is a backlogged session which is under-utilizing the resources guaranteed to it.

The $EWFQ$ scheme ensures an accurate inter-packets spacing (i.e. closer to that in GPS scheme) which helps the network to generate precise feedback information for source. Moreover, session's packets get distributed more accurately with no additional cost, rather the $EWFQ$ scheme is highly probable to perform lesser number of operations than other schemes (e.g. WFQ, WF^2Q) while ensuring better packet's *scheduling*.

References

[1] Jon C.R. Bennett, Hui Zhang. WF^2Q: Worst-case Fair Weighted Fair Queuing. *IEEE INFOCOM, Mar-96, pp. 120-128.*

[2] S. Jamaloddin Golestani. A Self-Clocked Fair Queuing Scheme for Broadband Applications. *IEEE INFOCOM'94, Toronto, CA, June 1994, pp. 636-646.*

[3] A. Parekh. A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks. *Ph.D. dissertation, Massachusetts Institute of Technology, February 1992.*

[4] S. Keshav. A control-theoretic approach to flow control. *Proceedings of ACM SIGCOMM'91, pages 3-15, Zurich, Switzerland, September 1991.*

[5] L. Kleinrock. Queuing Systems, *Vol 1, Computer Applications. Wiley, 1974.*

[6] S. Shenker. Making greed work in networks: A game theoretical analysis of switch service discipline. *Proceedings of ACM SIGCOMM'94. pg 47-57, London, UK, August 1994.*