Delay bound of youngest serve first (YSF) aggregated packet scheduling

L. Zhu, G. Cheng and N. Ansari

Abstract: A simple scalable aggregated traffic scheduling scheme is proposed, called the 'youngest serve first' (YSF) algorithm. It is shown analytically that YSF can provide bounded end-to-end delay time with high link utilisation that may not be possible for the first-in first-out (FIFO) scheme.

1 Introduction

For the past decade, how to provide end-to-end quality of service (QoS) guarantees has received much attention. Conventional solutions to this problem are rooted in per flow based scheduling schemes [1, 2]. The most significant drawback of these approaches is the lack of scalability, which could hamper the provision of end-to-end QoS guarantees in large internet service providers (ISPs) [3]. Recently, aggregated packet scheduling has been the focus of research as a possible alternative to provide a scalable approach to end-to-end QoS guarantees. In particular, the Differentiated Services Working group has proposed RFC 2598 [4], which defines expedited forwarding per hop behaviour (EF PHB). In this approach, EF traffics, which are regulated at the network edge, share a single FIFO buffer and are scheduled in an aggregated manner in the core network. FIFO packet scheduling is one of the most attractive approaches because of the implementation simplicity.

EF PHB aims to guarantee bandwidth at both large and small time scales, but whether it can guarantee end-to-end QoS still remains unclear. It was believed that end-to-end QoS in the network could be guaranteed if the link utilisation was kept small enough (i.e. less than 50%). Recent studies [3, 5] show that the worst-case end-to-end delay bound for EF traffic through the network is proportional to $1/\{1-(H-1)\alpha\}$ if the FIFO scheme is applied, where H is the number of hops along the longest path of all the flows in the network, and α, the so-called link utilization, is the ratio between the total amount of EF traffic on the link and the capacity of the corresponding link. It is clear that the worst-case delay is bounded only when $\alpha < 1/(H-1)$. Thus, the provisioning power of traffic aggregation is significantly weakened. The reason behind the difficulty of obtaining bounded end-to-end delay for an arbitrary network topology is that in aggregated scheduling, packet delay not only depends on the traffic behaviour of the flows sharing the same queue, but also on the traffic patterns in the whole network, even those that occurred a long time ago [3].

In this paper, a new simple aggregated packet scheduling algorithm is proposed: youngest serve first (YSF) for EF traffics. The main objective of YSF is to achieve a better end-to-end delay bound for EF traffics than a FIFO aggregate scheduling scheme. In YSF, EF traffics are shaped at the network edge. A label value is used to indicate the packet state and is encoded in a certain field in each packet header. As packets travel through the network, the encoded information is updated. All the packets are scheduled based on the information carried in the header. This approach not only has low computational complexity, but it also needs a very limited number of bits ($\log_2 H$) to carry the label in the packet headers. Most importantly, it can provide bounded end-to-end delay for any $\alpha < 1$ and any H.

2 Network model, terminology, and background

It is assumed that there are at least two classes of end-to-end flows [3] including the class of EF traffics, which are served with strict priority over other classes of traffics. In general, EF services can be realised by the guaranteed rate (GR) scheme instead of being limited to priority queueing [5]. The proposed YSF can be extended to the GR framework. It is also assumed that all network nodes perform the same packet-scheduling algorithm based on the limited information carried in the packet headers. Before entering the network, EF traffic flow k is shaped at the network edge to conform to a token bucket with parameters (r^k, β^k), which is the traffic arrival curve satisfying $A^k(t_0, t_0+t) \le r^k t + \beta^k$, where $A^k(t_0, t_0+t)$ is the total traffic from flow k released to the network during time interval $[t_0, t_0+t]$. Denote $F(I)$ as the set of flows traversing node I. It is assumed that for every $F(I)$ and I, the following conditions hold [3, 5]:

$$\sum_{k \in F(I)} r^k \le \alpha C_I \tag{1}$$

and

$$\sum_{k \in F(I)} \beta^k \le \beta C_I \tag{2}$$

where α (<1) is the link utilisation factor, β is a constraint on the burstiness of all flows through I, and C_I is the outgoing link capacity of I. According to [3], β is linearly dependent on α, and so we set $\beta = \tau_0 \alpha$, where τ_0 is a constant. In this paper, the fluid traffic model is adopted,

IEE Proceedings online no. 20030239

doi:10.1049/ip-com:20030239

Paper first received 15th March and in revised form 19th August 2002

The authors are with the Advanced Networking Lab, Electrical and Computer Engineering Dept., New Jersey Institute of Technology, University Heights, Newark NJ 07102, USA

but this work can easily be extended to the packet traffic model. The effect of propagation delay is also assumed negligible.

The following notation is adopted: d_i represents the maximum delay (worst case) experienced by packet p at the ith node along its path from the source to the destination, and D_i represents the maximum total delay (worst case) experienced by packet p from the first node to the ith node (inclusive) along the path, i.e.

$$D_i = \sum_{j=1}^{i} d_j$$

Next, some basic conclusions from network calculus theory are reviewed [6, 7]. For simplicity, $A(t)$ replaces $A(t_0, t_0+t)$ as the total traffic arrival curve. Denote $S(t)$ as the traffic service curve, which, in this case, is $S(t)=Ct$. The number of packets stored at each node is at most B, which is given by

$$B = A \oslash S(0) \tag{3}$$

where $\oslash$, deconvolution, is defined by

$$A \oslash S(t) := \sup_{\tau \in R} \{A(t+\tau) - S(\tau)\} \tag{4}$$

If the total traffic arrive curve is

$$A(t) = \alpha Ct + \beta C \tag{5}$$

it can be verified that

$$B = \beta C \tag{6}$$

In Fig. 1, B is the maximum number of packets stored, which is βC as shown above. t_B, the maximum queueing delay, is β in our case if FIFO scheduling is adopted. t_d is the maximum burst length or the longest time for the system to clear the queue, as long as work-conserving scheduling algorithms are adopted. t_d can be obtained by solving the following equation:

$$S(t_d) = A(t_d) \tag{7}$$

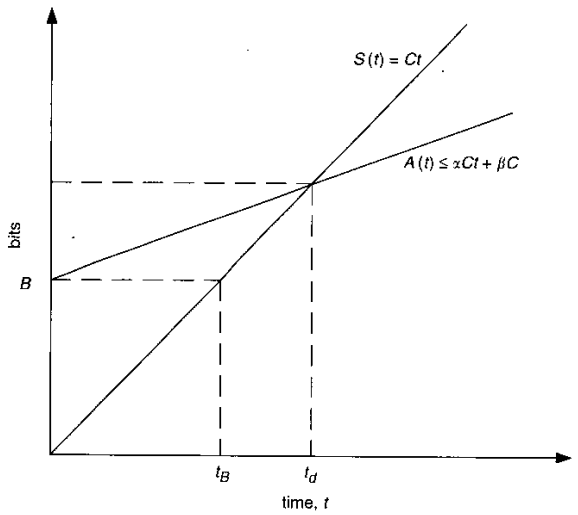

Fig. 1 *Illustration of basic concept of network calculus theory*

3 The youngest serve first (YSF) algorithm

It is assumed that the edge node is the first hop for all EF traffic. The proposed YSF algorithm works as follows: before entering the network, each packet is labelled with the number 1; at each node, the packet with the smallest label value is served first, and packets with the same label value are processed in a FIFO manner; the label on each packet is increased by one just before they are transmitted. The label value of a packet indicates the time it has spent in the network, or more precisely, the number of hops it has traversed through the network. This is why the scheme is called 'youngest serve first' [Note 1].

Naturally, the worst-case delay bound is experienced by packets with H hops from their source to the destination; H, as defined earlier, is the number of hops along the longest path of all the flows in the network. Consider a packet p. It can be seen that at different nodes along the path, p experiences different delay bounds. At the first hop, p has the highest priority due to the smallest label value carried in its header. After p is transmitted from the first hop node, the label value is increased to 2 at the second hop node. Thus, p cannot receive service as long as there are packets with label value 1 in that node. Intuitively, the longer the packet stays in the network, the larger the maximum delay it will experience at each node. In other words, $d_i \leq d_j$, for $1 \leq i \leq j \leq H$. In the rest of this paper, traffic is grouped based on labels carried in packet headers.

Definition 1: Denote $F_j(I)$ as the set of flows which assume their jth hop at node I. Therefore, $F(I) = F_1(I) \cup F_2(I) \cup \ldots \cup F_H(I)$, Define ρ_I^j and σ_I^j as

$$\rho_I^j = \sum_{i \in F_j(I)} r^i \tag{8}$$

$$\sigma_I^j = \sum_{i \in F_j(I)} \beta^i \tag{9}$$

In other words, ρ_I^j is the sum of rates of flows that assume their jth hop at node I, and σ_I^j is the sum of burstiness of flows that assume their jth hop at node I.

Thus, (1) and (2) can be rewritten as

$$\sum_{k \in F(I)} r^k = \sum_{j=1}^{H} \sum_{k \in F_j(I)} r^k = \sum_{j=1}^{H} \rho_I^j \leq \alpha C_I \tag{10}$$

$$\sum_{k \in F(I)} \beta^k = \sum_{j=1}^{H} \sum_{k \in F_j(I)} \beta^k = \sum_{j=1}^{H} \sigma_I^j \leq \beta C_I \tag{11}$$

Next, we derive the end-to-end delay bound with respect to the link utilisation α, the value of H, and the burstiness constraint β.

Lemma 1: The maximum delay experienced by packet p at its first hop in the network is $d_1 = \beta$ (also D_1).

Proof: Suppose the first hop node is I. Since packet p has the smallest label value 1, it has the highest priority in the system. Any other packets with a larger label value will not affect the service time of p. As a result, p experiences the worst-case delay when I is the first hop for all the flows traversing it. In this case, all packets have the same label value or priority. Thus, the scheduling algorithm is just FIFO. According to (10) and (11), the maximum overall traffic arrival curve at I is $A(t) = \alpha C_I t + \beta C_I$. According to Fig. 1, based on the network calculus theory introduced earlier and the fluid traffic model used throughout this paper, the delay bound is β. □

Note 1: Youngest serve first is with respect to the hop count.

Lemma 2: The maximum delay experienced by packet p from its first hop node to the second hop node (inclusive) along the path is bounded by

$$D_2 = \beta + \frac{\beta}{1-\alpha}$$

Proof: This lemma could be proved by the alternative approach used in [8]. Suppose the second hop node for p is I. According to the analysis in lemma 1, only packets with label value either 1 or 2 can affect the delay time of p at I. To obtain the worst-case delay bound for flow j at I, it is assumed that only packets with label value 1 and 2 traverse node I. In other words, only packets experiencing their either first or second hop at I are considered. Since packets labelled with 2 may have already experienced delay D_1, the traffic arrival curve for those flows is $\rho_I^2(t+D_1)+\sigma_I^2$ instead of $\rho_I^2 t+\sigma_I^2$ [3]. For flows with the first hop at I, the arrival traffic curve is still $\rho_I^1 t+\sigma_I^1$. Therefore, the total arrival traffic curve at I is

$$A(t) = \rho_I^1 t + \sigma_I^1 + \rho_I^2(t+D_1) + \sigma_I^2 \tag{12}$$

To obtain the maximum delay, the equalities in (10) and (11) are used:

$$\sum_{j=1}^{2} \rho_I^j = \alpha C_I \tag{13}$$

$$\sum_{j=1}^{2} \sigma_I^j = \beta C_I \tag{14}$$

Using (13) and (14), one can rewrite (12) as

$$\begin{aligned} A(t) &= \alpha(1-x)C_I t + x\alpha C_I(t+D_1) + \beta C_I \\ &= \alpha(1-x)C_I t + x\alpha C_I t + x\alpha C_I D_1 + \beta C_I \end{aligned} \tag{15}$$

where $x = \rho_I^2/\alpha C_I$. As can be seen, the burst size of the overall traffic is changed from βC_I to $x\alpha C_I D_1 + \beta C_I$. According to the explanation in Section 2, it is known that when packet p arrives at node I, the maximum buffer occupied at I is $x\alpha C_I D_1 + \beta C_I$. All the packets in the buffer before p arrives will be served before p, since their priorities are not lower than p; packets that are labelled with 2 and arrive after p will not affect its departure time, because they carry the same labels as p and will be served in FIFO order. However, those packets labelled with 1, of which the arrival rate is $(1-x)\alpha C_I$, will affect the departure time of p even though they enter the queue after p. Thus, the effective total arrival traffic curve that can determine the departure time of p is

$$A_{eff}(t) = (1-x)\alpha C_I t + (x\alpha C_I D_1 + \beta C_I) \tag{16}$$

instead of (15). If we replace $A(t)$ in Fig. 1 with $A_{eff}(t)$, the maximum delay experienced by p at its second hop node I is $d_2 = t_d$. Using (7), d_2 can be determined from the following equality:

$$C_I d_2 = A_{eff}(d_2) = (1-x)\alpha C_I d_2 + (x\alpha C_I D_1 + \beta C_I) \tag{17}$$

Thus,

$$d_2 = \frac{x\alpha D_1 + \beta}{1-(1-x)\alpha} = \frac{(x\alpha+1)\beta}{1-(1-x)\alpha} \tag{18}$$

Since

$$\frac{\partial d_2}{\partial x} = \frac{-\alpha^2}{\{1-(1-x)\alpha\}^2}\beta < 0 \tag{19}$$

when $x=0$, d_2 reaches its maximum:

$$d_2 = \frac{\beta}{1-\alpha} \tag{20}$$

Hence,

$$D_2 = D_1 + d_2 = \beta + \frac{\beta}{1-\alpha} \tag{21}$$

□

After packet p takes its second hop and moves to the jth (>2) node, there are possibly more packets with smaller label values or higher priorities at that node. Intuitively, p could experience a longer delay at those nodes. Next, in theorem 1, the delay bound is derived as p moves closer to its destination.

Theorem 1: The maximum delay experienced by packet p from its first hop node to the kth hop node (inclusive) along the path is bounded by

$$D_k = D_{k-1} + \frac{\beta + \alpha D_{k-2}}{1-\alpha} \tag{22}$$

for any $k \geq 3$.

Proof: This theorem could be proved by an approach applying the worst-case delay for a priority queue [8]. Here we use induction to complete the proof. One can reasonably define $D_0 = 0$. From lemma 1 and lemma 2 we have

$$D_2 = D_1 + d_2 = \beta + \frac{\beta}{1-\alpha} = D_1 + \frac{\beta + \alpha D_0}{1-\alpha} \tag{23}$$

Assume

$$D_k = D_{k-1} + \frac{\beta + \alpha D_{k-2}}{1-\alpha} \tag{24}$$

Next, it is shown that the above expression holds for $k+1$. Let node I be the $(k+1)$th hop of packet p, and thus only packets with label not larger than $k+1$ can affect the delay of p at I. In other words, only packets assuming their jth ($j \leq k+1$) hop at node I will be considered. The overall arrival traffic curve can be written as

$$A(t) = \sum_{j=1}^{k+1} \{\rho_I^j(t+D_{j-1}) + \sigma_I^j\} \tag{25}$$

Define $x_j = \rho_I^j/\alpha C_I$, and also note that $\sum_{j=1}^{k+1} x_j = 1$. Then (25) can be rewritten as

$$\begin{aligned} A(t) &= \sum_{j=1}^{k+1} \{\rho_I^j(t+D_{j-1}) + \sigma_I^j\} = \sum_{j=1}^{k+1} \rho_I^j(t+D_{j-1}) + \sum_{j=1}^{k+1} \sigma_I^j \\ &= \sum_{j=1}^{k+1} x_j \alpha C_I(t+D_j) + \beta C_I \leq x_{k+1}\alpha C_I t + x_{k+1}\alpha C_I D_k \\ &\quad + \sum_{j=1}^{k} x_j \alpha C_I t + \sum_{j=1}^{k} x_j \alpha C_I D_{k-1} + \beta C_I \\ &= x_{k+1}\alpha C_I t + x_{k+1}\alpha C_I D_k + (1-x_{k+1})\alpha C_I t \\ &\quad + \sum_{j=1}^{k} x_j \alpha C_I D_{k-1} + \beta C_I = x_{k+1}\alpha C_I t + (1-x_{k+1})\alpha C_I t \\ &\quad + [x_{k+1}\alpha C_I D_k + (1-x_{k+1})\alpha C_I D_{k-1} + \beta C_I] \end{aligned} \tag{26}$$

The inequality in (26) is based on the fact that $D_i < D_{k-1}$ if $i < k-1$. The third term in the last equality stands for the maximum traffic queued in the system when packet p joins

the queue. The first term can be viewed as traffic which assumes the $(k+1)$th hop at node I, and arrives after p, and thus cannot affect p's departure time. The second term represents traffic with smaller label values, which also arrives after p, but can affect the departure time of p. Using a similar argument to that the proof of lemma 2, the effective arrival traffic curve can be written as

$$A_{eff}(t) = (1 - x_{k+1})\alpha C_I t + x_{k+1}\alpha C_I D_k + (1 - x_{k+1})\alpha C_I D_{k-1} + \beta C_I \quad (27)$$

By solving for d_{k+1} in the following equation

$$C_I d_{k+1} = A_{eff}(d_{k+1}) = (1 - x_{k+1})\alpha C_I d_{k+1} + x_{k+1}\alpha C_I D_k + (1 - x_{k+1})\alpha C_I D_{k-1} + \beta C_I \quad (28)$$

we get

$$d_{k+1} = \frac{x_{k+1}\alpha D_k + (1 - x_{k+1})\alpha D_{k-1} + \beta}{1 - (1 - x_{k+1})\alpha} \quad (29)$$

Thus,

$$\frac{\partial d_{k+1}}{\partial x_{k+1}} = \frac{\alpha[(1-\alpha)D_k - D_{k-1} - \beta]}{\{1 - (1 - x_{k+1})\alpha\}^2} = \frac{\alpha\left[(1-\alpha)\left(D_{k-1} + \frac{\alpha D_{k-2} + \beta}{1-\alpha}\right) - D_{k-1} - \beta\right]}{\{1 - (1 - x_{k+1})\alpha\}^2} = \frac{\alpha(D_{k-2} - D_{k-1})}{\{1 - (1 - x_{k+1})\alpha\}^2} < 0 \quad (30)$$

Equation (24) has been used to reach the third equality in (30). So, when $x_{k+1} = 0$, d_{k+1} reaches its maximum value. From (29),

$$d_{k+1} = \frac{\alpha D_{k-1} + \beta}{1 - \alpha} \quad (31)$$

and

$$D_{k+1} = D_k + d_{k+1} = D_k + \frac{\alpha D_{k-1} + \beta}{1 - \alpha} \quad (32)$$

□

Using the recursive relationship in (32) with the initial conditions stated in lemma 1 and lemma 2, we have

$$D_j = \frac{\beta}{\alpha}\left\{\frac{r_2 - \alpha - 1}{r_2 - r_1} r_1^j + \frac{r_1 - \alpha - 1}{r_1 - r_2} r_2^j - 1\right\} \quad j \geq 3 \quad (33)$$

where

$$r_{1,2} = \frac{1 - \alpha \pm \sqrt{-3\alpha^2 + 2\alpha + 1}}{2(1 - \alpha)}$$

are the roots of the following quadratic equation

$$r^2 - r - \frac{\alpha}{1 - \alpha} = 0 \quad (34)$$

It can be shown using (33) with further algebraic manipulation that $D_H \propto H\beta$ when α is very small, and $D_H \propto (1-\alpha)^{-H/2}$ when $\alpha \approx 1$. Based on the above analysis, one can conclude that the link utilisation, which is independent of H, can approach 1, and the end-to-end delay is also bounded at the same time. The end-to-end delay bound for FIFO [3] is $H\beta/\{1-\alpha(H-1)\}$, if the fluid traffic model is applied. Note that β is denoted by τ in [3].

Fig. 2 shows the performance comparison between YSF and FIFO with $H = 10$. The vertical axis is the number of time units (in terms of τ_0). With a given link utilisation α, YSF performs much better than FIFO, especially when α is large.

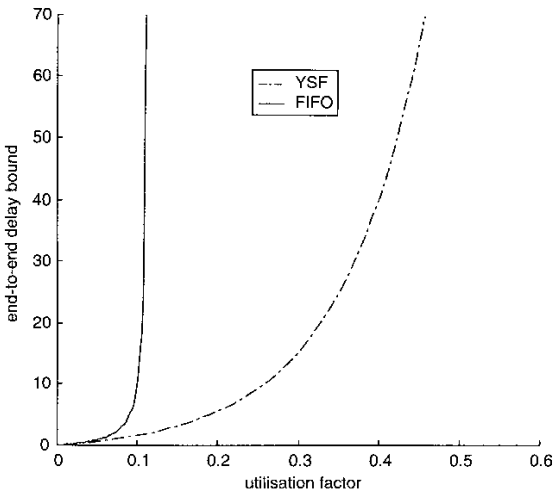

Fig. 2 *Performance comparison between YSF and FIFO*

It may seem very natural to adopt another scheme, namely oldest serve first (OSF), in which the packets with the largest label value are served first instead of being served last in YSF. Next, it is shown that YSF performs better than OSF.

Lemma 3: In OSF, for any packet p with H hops to its destination, the worst-case delay experienced upto the $(H-1)$th hop (inclusive) is

$$D_{H-1} = \frac{(H-1)\beta}{1 - \alpha H}$$

Proof: To derive the worst-case delay, it is assumed that the node I is the kth $(1 \leq k \leq H-1)$ hop node along p's path to its destination, and all packets from other flows assume their Hth hop at I. According to OSF, packet p has the lowest priority and can only be served when no other packets are in the corresponding node. Since other flows may all experience the maximum delay D_{H-1}, the total arrival traffic curve is

$$A(t) = \alpha C_I(t + D_{H-1}) + \beta C_I \quad (35)$$

Then, d_k, the delay bound of p experienced at the kth node can be obtained by solving the following equation:

$$S(d_k) = C_I d_k = A(d_k) = \alpha C_I(d_k + D_{H-1}) + \beta C_I \quad (36)$$

Thus, we get

$$d_k = \frac{\alpha D_{H-1} + \beta}{1 - \alpha} \quad (37)$$

On the other hand, we also have

$$D_{H-1} = \sum_{k=1}^{H-1} d_k = (H-1)\frac{\alpha D_{H-1} + \beta}{1 - \alpha} \quad (38)$$

Hence,

$$D_{H-1} = \frac{(H-1)\beta}{1-\alpha H} \tag{39}$$

□

From lemma 3, it can be seen that D_{H-1} is bounded only when $a < 1/H$. Since α can approach 1 in YSF, YSF achieves higher link utilisation. From (39), it is also seen that in OSF, when α approaches 1, the end-to-end delay bound tends to infinity. However, in YSF, the end-to-end delay bound always remains finite as long as $\alpha < 1$. Thus, YSF also performs better than OSF in terms of the end-to-end delay bound.

4 Discussion and conclusions

A new aggregated traffic scheduling scheme, youngest serve first (YSF) has been proposed, and its end-to-end delay bound derived. YSF has been proven to have the following merits. First, link utilisation α in YSF can approach 1, regardless of the network topology and the value of H; second, the end-to-end delay bound in YSF is much smaller than that in FIFO, and thus better end-to-end delay bound can be guaranteed. Even with the additional complexity required, which is rather low, the advantages described make YSF preferable to FIFO. At each node, there are H different label values. Thus, we need at most H queues, corresponding to the different label values. Packets are placed into different queues based on their label values and the backlogged queue with the smallest label value is served first. Therefore, YSF is scalable because we need to manage only H queues, no matter how many flows traverse each node. Note also that only $\log_2 H$ bits are required to encode the label. One may use either the TTL field or the TOS field in the IP header to realise YSF, but this issue is beyond the scope of this paper. Ideas from timestamp-based scheduling algorithms such as WFQ [1] and WF^2Q [2] may be incorporated in designing aggregated traffic scheduling schemes. However, YSF possesses the following advantage not shared by the timestamp-based approaches: the nodes in the network need not be synchronised in time, and timestamps need not be computed and updated.

YSF can be incorporated in Diffserv, which is the emerging service architecture for the Internet. YSF can be an alternative to the FIFO scheme, which is currently employed in EF traffic scheduling. EF is usually assumed to support delay-sensitive applications, e.g. audio streaming. As addressed in [3] and [5], the FIFO scheme can provide a strict end-to-end delay bound only for small link utilisation and limited hop count. This shortcoming of FIFO can severely limit the QoS provisioning required by many delay-sensitive applications using EF. Retaining the simplicity of aggregate scheduling, YSF provides strict and low end-to-end delay bound for any hop count and link utilisation (less than 1).

5 Acknowledgments

This work has been supported in part by the New Jersey Commission on Higher Education via the NJI-TOWER project, and the New Jersey Commission on Science and Technology via the NT Center for Wireless Telecommunications.

6 References

1 PAREKH, A.K., and GALLAGAR, G.: 'A generalized processor sharing approach to flow control in integrated services networks - The single node case'. Proceedings of IEEE INFOCOM'92, Florence, Italy, 1992, Vol. 2, pp. 915–924
2 BENNETT, J.C.R., and ZHANG, H.: 'WF^2Q: worst-case fair weighted fair queuing'. Proceedings of IEEE INFOCOM'96, 1996, pp. 120–128
3 CHARNY, A., and LE BOUDEC, J.-Y.: 'Delay bounds in a network with aggregated scheduling'. Proceedings of QoFIS, Berlin, Germany, October 2000, pp. 1–13
4 JACOBSON, V., NICHOLAS, K., PODURI, K.: 'An expedited forwarding PHB', RFC 2598, June 1999
5 BENNETT, J., BENSON, K., CHARNY, A., COURTNEY, W., and LE BOUDEC, J.Y.: 'Delay jitter bounds and packet scale rate guarantee for expedited forwarding'. Proceedings of IEEE INFOCOM, Anchorage, AL, USA, April 2001, pp. 1502–1509
6 AGRAWAL, R., CRUZ, R.L., OKINO, C., and RAJAN, R.: 'Performance bounds for flow control protocols', *IEEE/ACM Trans. Netw.*, June 1999, **3**, (7), pp. 310–323
7 LE BOUDEC, J.-Y., and THIRAN, P.: 'Network calculus, a theory of deterministic queuing systems for the Internet' (Springer Verlag-LNCS 2050, June 2000)
8 CHANG, C.S.: 'Performance guarantees in communication networks' (Springer, 2000), Chap. 2.3