

Frame-based priority scheduling in hybrid IP/ATM networks

Y.-C. Chen^{a,*}, S.-C. Hu^a, C.-T. Chan^b, C.-L. Lee^a

^aDepartment of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu 30050, Taiwan

^bTelecommunication Laboratories, Chunghwa Telecom Co. Ltd, Taipei, Taiwan

Received 23 December 1998; received in revised form 20 April 1999; accepted 20 April 1999

Abstract

The exploding growth of World Wide Web applications in recent years has overwhelmed the internet with multimedia traffic. A future performance concern is to keep pace with the quickly growing bandwidth requirements. IP over ATM is a way to relax such requirements. It provides a hybrid approach to support both layer 3 software forwarding and layer 2 ATM hardware switching. A switched virtual connection may be suitable for traffic involving large data transfers and QoS provision like Guarantee Service (GS) applications. In the hybrid IP/ATM networks, the packet-level instead of the cell-level behavior is the relevant measure of performance. In this paper, we propose a so-called Frame-based Priority Scheduling (FBPS) algorithm for GS applications. FBPS focuses on the packet-level performance while transmitting based on cells. The most attractive feature of FBPS is its simplicity. We provide a systematic analysis of FBPS scheduler and derive the bound on its fairness when the flow traffic is shaped by a leaky bucket. It shows that FBPS is competitive with certain complex scheduling disciplines such as PGPS. We also verify our analytical bounds by simulation. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Frame-based priority scheduling; Guarantee service; Real-time variable bit rate

1. Introduction

IP and ATM integration is a well-discussed topic for the provision of multimedia communication services. The Internet Draft [1] by IETF indicates the trend of IP over ATM towards a hybrid approach to support both layer 3 software forwarding and layer 2 ATM hardware switching. An example is Multiprotocol over ATM (MPOA) [2] which is built upon both LANE [3] and NHRP [4]. MPOA supports both default forwarding across subnetwork borders via route servers and cut-through switching to bypass route servers.

The flow discrimination mechanism is considered as a key component for MPOA. As an IP flow may not require a direct ATM SVC, the IETF introduced a general idea [1] in which short-lived flows composed of few packets (e.g. DNS and SMTP) are well suited for connectionless service, while long-lived flows containing a large number of packets would prefer connection-oriented service. A more flexible flow discriminator still needs further study. Generally, we can say that a dedicated SVC should be set up for the Guarantee Service (GS) flow. GS [5] provides an assured level of bandwidth, a firm end-to-end delay bound, and no queuing loss for conforming packets of a data flow, it can be mapped

to Real-time Variable Bit Rate (rtVBR) service in ATM networks [1]. Resources reservation for each SVC in hybrid IP/ATM networks can be accomplished through the integration of ATM signaling and RSVP [6].

Effective scheduling algorithms are required for providing the QoS guarantee. The important features of schedulers are not identical for different types of traffic. For traditional data flows without any QoS request, short end-to-end delay is not crucial while instantaneous fairness (two backlogged connections are served fairly in any time interval) is important. On the contrary, the real-time applications emphasize low end-to-end delay, and instantaneous fairness is unconvincing to worst-case QoS guarantees. Other desirable common features for all traffic types are isolation of flows, simplicity of implementation and high link utilization [7].

In general, packet scheduling can be characterized as either work-conserving or nonwork-conserving. It is generally known that work-conserving disciplines are more efficient than nonwork-conserving disciplines in resource utilization. In this work, we focus on the traffic control method for GS/rtVBR traffic in hybrid IP/ATM networks. We present a so-called Frame-based Priority Scheduling (FBPS) scheme, which focuses on the packet-level performance. FBPS is based on a concept of time frame and priority queuing, it is also preemptive and work-conserving. Comparing with PGPS [8,9], FBPS has a slightly longer maximum end-to-end packet delay theoretically, although,

* Corresponding author. Tel.: + 886-35-731864; fax: + 886-35-727842.

E-mail address: ycchen@csie.nctu.edu.tw (Y.-C. Chen)

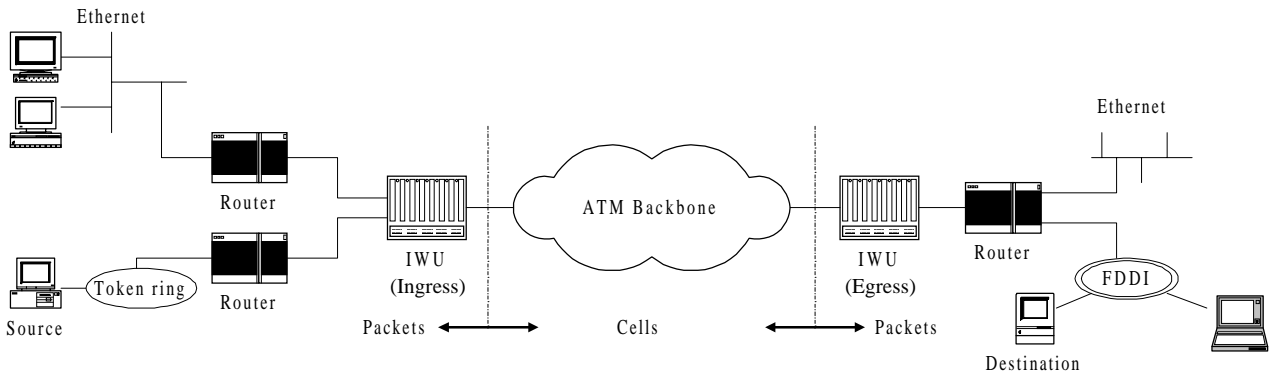


Fig. 1. The internetworking environment of packet transmission over ATM backbone.

FBPS saves the complexity of maintaining virtual time and sorting. Meanwhile, FBPS features both simplicity and efficiency for flow admission, therefore it is more feasible for implementation than PGPS in ATM networks.

The rest of this paper is organized as follows. In Section 2, we describe the motivation of our work. Section 3 addresses the traffic control framework, the call admission control algorithm and the FBPS scheduling algorithm. The detailed analysis of FBPS is discussed in Section 4. The system simulation results are presented in Section 5. Finally, Section 6 concludes the work.

2. Motivation

Consider a typical internetworking environment as shown in Fig. 1, LANs connect to an ATM backbone via Internetworking Unit (IWU). Consider a flow that, in the absence of topology change, takes the same route throughout the

network. The source host of a flow transmits multimedia data through IP packets. Upon entering the IWU (ingress), each IP packet is encapsulated in an AAL5 packet that is fragmented into 53-byte cells. Then these cells are transmitted through the backbone to the IWU at the destination side (egress). The egress needs to collect all cells, which can be reassembled, into a higher layer packet required by the destination. Therefore, cells will be queued at the egress until the last cell of the same packet is received. Obviously, the delay incurred to deliver a packet instead of a cell will be the relevant measure of performance.

At SONET OC-3 rate the transmission time of a cell is less than 2.8 μ s, it is even smaller for higher speeds such as OC-12. The sort-based schedulers like Virtual Clock [10], PGPS [8,9], WF²Q [11] and SCFQ [12] require at least $O(\log N)$ time complexity, where N is the number of connections, to make a scheduling decision. Meanwhile, all these schedulers need to maintain a virtual finish time on a per cell basis. These factors complicate the cell

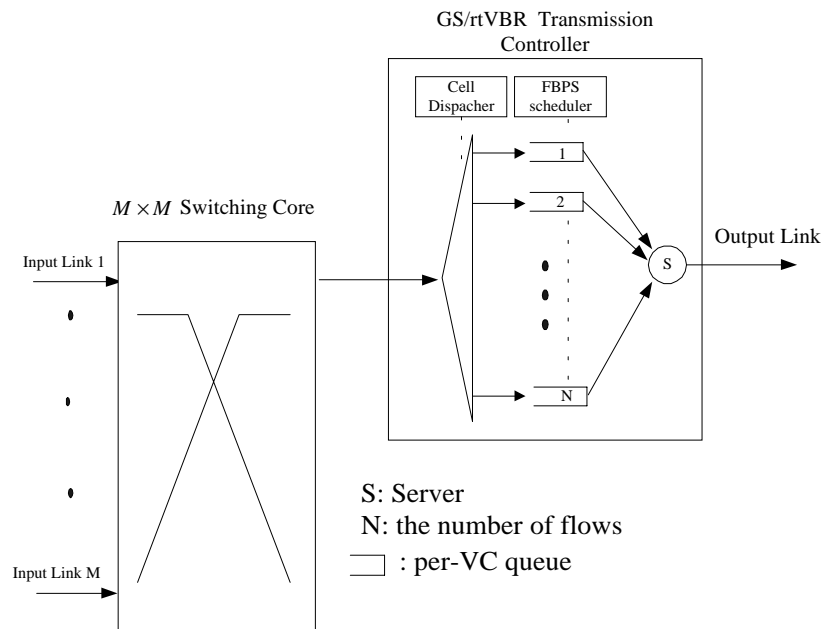


Fig. 2. Proposed GS/rtVBR transmission controller in an ATM switching node.

transmission process. Shreedhar et al. proposed a quite simple service discipline called DRR [13] that is designed to accommodate variable length packets of a flow. DRR and WRR are alike for network with small and fixed packet sizes, such as ATM. Although DRR and WRR have only $O(1)$ computational complexity for packet scheduling, their bandwidth allocation schemes are very restricted, which results in poor system flexibility.

In hybrid IP/ATM networks, the scheduling algorithm should be simple as well as flexible to meet various high-speed transmission requirements. Besides, the ATM cells should be served with the concern that packet-level QoS must be met.

3. Frame-based priority scheduling

In this section, we present a scheduling algorithm for hybrid IP/ATM networks. As it is frame-based with data being served according to their priorities, we name the mechanism Frame-based Priority Scheduling (FBPS).

3.1. The traffic control framework

Fig. 2 illustrates our GS/rtVBR traffic control architecture. The transmission controller consists of a cell dispatcher (CD), an FBPS scheduler and a buffer pool which can be logically partitioned into N per-VC queues, $Queue_i$, $1 \leq i \leq N$. Each per-VC queue uses FIFO discipline. The cell dispatcher distributes a cell to its designated FIFO queue. The FBPS scheduler fetches a Head-of-Line (HOL) cell from a per-VC queue for transmission.

Resource reservation is a must for GS flows to achieve deterministic QoS. Assume that the bandwidth r_i for flow i is reserved in each intermediate node along the end-to-end path, so that the maximal-sized packet can reach the destination in time. In FBPS, every flow i is characterized by (Q_i, T_i) where Q_i is the maximal IP packet size of flow i and T_i is the time-frame, $T_i = Q_i/r_i$. To ease the description, Q_i is measured in units of cells, r_i is in units of cells/cell-time-slot and T_i is in units of cell-time-slot. For each flow i , at most Q_i cells can be transmitted within a T_i interval. For any two flows i and j , we say that flow i has higher service priority than flow j if and only if $T_i < T_j$. In the cell level, a flow i with higher priority can preempt a lower priority flow j that is being served currently. A connection admission criterion is essential to assure that all flows are schedulable.

3.2. The call admission control

The purpose of call admission criterion here is to ensure that cells of a new flow i gets Q_i chances for services in any time interval T_i , and the QoS of existed flows should not be violated. Suppose that there are N flows with characteristics (Q_j, T_j) , $1 \leq j \leq N$. Without loss of generality, we can assume that $T_1 \leq T_2 \leq \dots \leq T_N$. The utilization of the server, $U = \sum_{i=1}^N (Q_i/T_i)$, must not be larger than 1. We

say that a flow i is schedulable if the following condition is satisfied:

$$\sum_{\forall j, T_j \leq T_i} \frac{Q_j}{T_i} \times \left\lceil \frac{T_i}{T_j} \right\rceil \leq 1. \quad (1)$$

The schedulability test is simple, although Eq. (1) is just a sufficient condition. To achieve better bandwidth utilization, we can use the schedulability test proposed in [14], which is for continuous model. In our case, it needs the following assumption; otherwise it would not work.

Assumption 1. The basic unit of service is a cell whose arrival can only happen at the boundary of a cell-time-slot. Each cell can be scheduled immediately after it arrives.

Lehoczky et al. [14] proposed that a flow i is schedulable if and only if the following condition holds:

$$\Omega_i = \min_{(k,l) \in H_i} \sum_{j=1}^i Q_j \times \left\lceil \frac{lT_k}{T_j} \right\rceil$$

where $H_i = \left\{ (k,l) \mid 1 \leq k \leq i, l = 1, \dots, \left\lfloor \frac{T_i}{T_k} \right\rfloor \right\}$

$$\times \frac{\Omega_i}{lT_k} \leq 1 \quad (2)$$

The schedulability test listed in Eq. (2) is more complex than that in Eq. (1). Nevertheless, we can relax the admission region to achieve a higher utilization. The algorithm for CAC is stated as follows:

Algorithm call admission control

Declaration:

N : the number of existed flows

(Q_i, T_i) : the flow characteristics of a new request

$\{f_{i+1}, f_{i+2}, \dots, f_N\}$: the subset of existed flows with larger time-frame size than T_i

$S_Test_1(Q_i, T_i)$: the schedulability test listed in Eq. (1), the output will be true if the inequality holds, otherwise the output is false.

$S_Test_2(Q_i, T_i)$: the schedulability test listed in Eq. (2), the output will be true if the inequality holds, otherwise the output is false.

R_{flag} : the recursive status flag, it is false if the recursive process terminates, and true otherwise

A_{flag} : the output status flag, it is true if the new connection is accepted, and false otherwise

Initialization: $A_{\text{flag}} = \text{false}; R_{\text{flag}} = \text{true}$

Procedure: CAC(i, Q_i, T_i)

1. Begin

2. If (R_{flag})

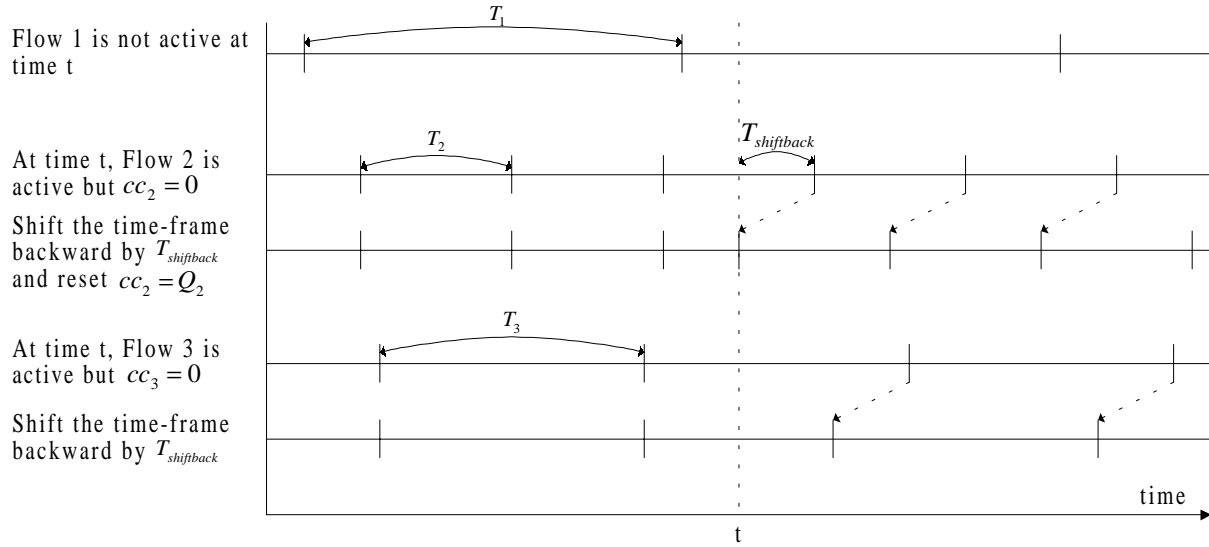


Fig. 3. An example of shifting the time-frame backward.

3. If ($i == N$)
4. If ($S_Test_1(Q_N, T_N)$)
5. $A_{flag} = true; R_{flag} = false;$
6. Else if ($S_Test_2(Q_N, T_N)$)
7. $A_{flag} = true; R_{flag} = false;$
8. Else
9. $R_{flag} = false;$
10. Else
11. If ($S_Test_1(Q_i, T_i)$)
12. $CAC(i + 1, Q_{i+1}, T_{i+1});$
13. Else If ($S_Test_2(Q_i, T_i)$)
14. $CAC(i + 1, Q_{i+1}, T_{i+1});$
15. Else
16. $R_{flag} = false;$
17. End

3.3. The FBPS algorithm

Each flow i is assigned 2 counters, the time-frame counter (TFC _{i}) and the credit counter (CC _{i}). When the first cell of flow i arrives at a switch, TFC _{i} and CC _{i} are set to T_i and Q_i , respectively. TFC _{i} will decrement by 1 every cell-time-slot until it reaches 0. Whenever a cell of flow i is scheduled, CC _{i} decrements by 1. Once a time-frame boundary (i.e. TFC _{i} = 0) is encountered, TFC _{i} will be reset to T_i , and CC _{i} will be reset to Q_i if $Queue_i$ is still backlogged, otherwise CC _{i} is set to 0. Let a flow be *active* if $Queue_i$ is nonempty, a cell of flow i is defined as *eligible* if it is head-of-line (HOL) and CC _{i} > 0. The *eligible* cell with the highest priority will be selected to serve by FBPS scheduler.

Basically, Q_i cells can be transmitted within a T_i time interval if flow i keeps backlogged during the interval. In general, the total traffic load will be less than one hundred percent due to the restriction of CAC, and it is very unlikely

that all admitted flows are active simultaneously. It may happen that at least one active flow exists but no cell is eligible because of the unassigned link capacity. As FBPS is a work-conserving scheduler, the server will never be idle if the output buffer is not empty. The unassigned link capacity should be allocated to the active flows fairly. This can be achieved by shifting the time-frames of active flows backward.¹ That is, we perform a time-frame shift for every active flow i by updating TFC _{i} as follows:

$$TFC_i = TFC_i - T_{shiftback},$$

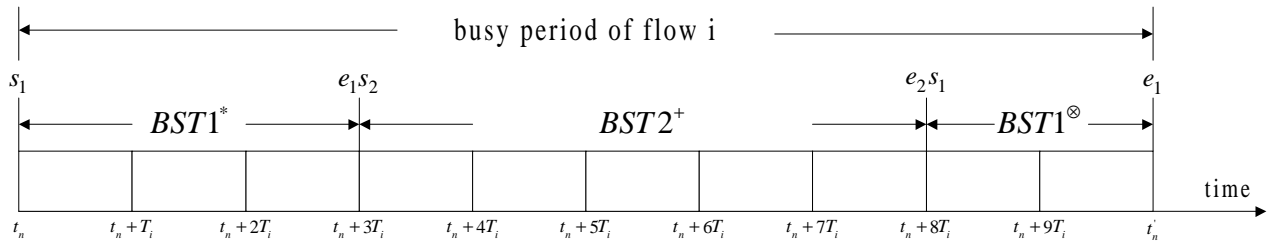
where

$$T_{shiftback} = \min_{\text{Active flow } j} (TFC_j).$$

As a sequel, CC _{i} of active flow i can be reset to Q_i . As illustrated in Fig. 3, flow 2 and flow 3 are active but no cell is eligible. Shifting the time-frames of both flows backward by $T_{shiftback}$ so that the time-frame boundary of flow 2 will start from t . CC₂ then can be reset to Q_2 and the HOL cell of $Queue_2$ will become eligible immediately.

When an active flow i encounters TFC _{i} = 0 with CC _{i} > 0, it means that flow i has missed its service chances because $Queue_i$ was not always backlogged during the past time-frame T_i . From the fairness aspect, the unused bandwidth is gone, therefore CC _{i} is reset to Q_i . For a newly backlogged flow j with CC _{j} = 0 and TFC _{j} = 0, its two counters are set to Q_j and T_j , respectively. The bandwidth accumulation that is the major drawback of Virtual Clock scheme, will not occur in our algorithm. Details of FBPS algorithm can be found in the Appendix A.

¹ The current time-frame of an active flow i is shorter than T_i because the next time-frame will start earlier.



$$\begin{aligned}
 &^* 0 \leq B_i(t_n + jT_i) \leq Q_i, j \in \{1,2,3\} \\
 &^+ B_i(t_n + jT_i) > Q_i, j \in \{4,5,6,7\} \\
 &^\otimes 0 \leq B_i(t_n + jT_i) \leq Q_i, j \in \{8,9,10\}
 \end{aligned}$$

Fig. 4. An example of busy period, BST1 and BST2.

4. Analysis of FBPS

In the hybrid IP/ATM networks, the IWU is responsible for the fragmentation of the encapsulated IP packet into cells at the source side and to reassemble cells into proper higher layer protocol data unit at the destination side. According to [5], the source traffic of a GS flow i obeys the behavior of a token bucket (σ_i, ρ_i) , where σ_i is the bucket depth and $\rho_i, \rho_i \leq r_i$, is the token rate. At an IWU, the traffic shaping mechanism is performed to reshape the incoming/outgoing traffic so that it conforms to the negotiated traffic contract. The overhead caused by LLC/SNAP [15] and AAL5 encapsulation is neglected in our analysis. The peak rate of the flow is defined as R_i . To ease the presentation, we assume that σ_i is in units of cells, while R_i and ρ_i are in units of cells/cell-time-slot. We also assume that q_i is equal to Q_i/C where C is the link capacity (i.e. C is 1 cells/cell-time-slot) and q_i is in units of cell-time-slot. The flow i travels through K Network Elements (NEs) in ATM backbone with NE_1 as the ingress and NE_K as the egress.

4.1. Delay and backlog analysis of a single network element

The scheduler of GS flows has to guarantee a bounded delay. In hybrid IP/ATM networks, packet delay is a more important concern than the individual cell delay. First, we address the packet delay and the backlog bound when FBPS is applied in a single network element (NE). Let $A_i(t, \tau)$ denote the traffic volume of flow i that arrives to the switch in the interval $[t, \tau)$, and $W_i(t, \tau)$ denote the amount of service received by flow i during the same interval. In ATM networks, we assume that $A_i(t, \tau)$ increases only when the last bit of a flow i cell enters $Queue_i$; likewise, $W_i(t, \tau)$ increases only when the last bit of a flow i cell leaves the server. As flow i conforms to the leaky bucket constraint, $A_i(t, \tau)$ should be bounded by

$$A_i(t, \tau) \leq \min\{Q_i + R_i(\tau - t), \sigma_i + \rho_i(\tau - t)\}, \quad \tau \geq t. \tag{3}$$

The following definitions are essential for the analysis of our work.

Definition 1. The n th busy period of flow i is a time interval $[t_n, t'_n]$ where t_n is the first cell arrival time after t'_{n-1} , and t'_n is the first time-frame boundary after t_n such that $Queue_i$ is empty at t'_n .

The busy period is defined recursively. The first busy period of flow i begins at the arrival time of its very first cell. Therefore, every busy period has independent time-framing. Note that the time-frame size is set to T_i , the upper limit, for flow i . As FBPS is work-conserving, the actual time-frame required may be less than T_i practically. However, the maximum delay would occur under the situation that every actual time-frame of flow i has the same size T_i .

Definition 2. A Busy Sub-period of Type 1 of flow i (BST1) is a maximum time interval $[s_1, e_1]$ such that the queue occupancy at every time-frame boundaries in $[s_1, e_1]$ is no greater than Q_i and no less than 0. Both s_1 and e_1 are time-frame boundaries.

Definition 3. A Busy Sub-period of Type 2 of flow i (BST2) is a maximum time interval (s_2, e_2) such that the queue occupancy at every time-frame boundary in (s_2, e_2) is larger than Q_i . Both s_2 and e_2 are time-frame boundaries.

An example of busy period, BST1 and BST2 are illustrated in Fig. 4. A busy period may consist of only BST1 or both BST1 and BST2. Let D_i be the queuing delay of a packet in flow i and $B_i(t)$ is the queue occupancy of $Queue_i$ at time t^- . The following theorem sets the bounds of D_i and $B_i(t)$ in BST1.

Theorem 1. In a BST1, the following bounds must hold:

1. At any time instant t within a BST1, $B_i(t) \leq 2Q_i$.
2. If the last cell of a packet of flow i arrives at time τ , $\tau \in [s_1, e_1)$, and $\tau + d$ is the closest time-frame boundary after τ then $D_i \leq d + \Omega_i$, where $0 \leq d < T_i$ and

$$\Omega_i = \min_{(k,l) \in H_i} \sum_{j=1}^i Q_j \times \left\lceil \frac{lT_k}{T_j} \right\rceil \quad \text{where}$$

$$H_i = \left\{ (k, l) \mid 1 \leq k \leq i, l = 1, \dots, \left\lfloor \frac{T_i}{T_k} \right\rfloor \right\}.$$

Proof. (1) Assume it is not the case, that is, $B_i(t) > 2Q_i$ at some time instance t in $[s_1, e_1)$. As at most Q_i cells are served in any time-frame T_i , if t' is the closest time-frame boundary after t , then $B_i(t') > Q_i$. This contradicts to Definition 2. (2) By Definition 2 $B_i(\tau + d) \leq Q_i$. In every time-frame T_i , Q_i cells can be served during the first Ω_i time-slots.² Therefore, all cells that arrived before or at τ can be served no later than $\tau + d + \Omega_i$. \square

Definition 4. A *Busy Sub-period of Type 3* of flow i (BST3) is a maximum time interval $[s_3, e_3)$ such that for any time $t \in [s_3, e_3)$, $A_i(s_3, t) \geq r_i \times (t - s_3)$.

Under the situation that a BST3 is contained in a BST1, the maximum delay and backlog has been shown in the above theorem. Now we focus on the maximum delay and backlog when a BST3 contains a BST2. Let $S_i(t, \tau)$ denote the total service provided to the packets of flow i that arrived within the time interval $(t, \tau]$. Notice that $W_i(s_3, \tau)$ may actually be larger than $S_i(s_3, \tau)$ because some cells that arrived before s_3 may still be queued in the system after s_3 and these cells must also be served as well.

Lemma 1. For any time instant t within the interval from s_3 that a BST3 starts to the time by which all packets arrived during this period are served,

$$S_i(s_3, t) \geq \max(0, r_i \times (t - s_3 - \Theta_i)), \quad (4)$$

where

$$\Theta_i = \Omega_i + T_i - q_i.$$

Proof. In every time-frame T_i , flow i can have q_i time-slots to transmit its cells. From Eq. (2), these Q_i cells can be served during the first Ω_i time-slots in a T_i . Suppose that $s_3 + d$ is the closest time-frame boundary after s_3 , consider the following two cases:

Case 1: $s_3 \leq t \leq s_3 + d + \Omega_i$

$$0 \leq S_i(s_3, t).$$

When $t = s_3 + d + \Omega_i$, $S_i(s_3, t) \geq r_i \times (t - s_3 - \Omega_i) = r_i d$

Case 2: $t > s_3 + d + \Omega_i$. Suppose that $\tilde{t} = (t - s_3 - d - \Omega_i) \bmod T_i$.

$$\begin{aligned} S_i(s_3, t) &= S_i(s_3, s_3 + d + \Omega_i) + W_i(s_3 + d + \Omega_i, t - \tilde{t}) \\ &\quad + W_i(t - \tilde{t}, t) \\ &\geq r_i d + \left\lfloor \frac{t - (s_3 + d + \Omega_i)}{T_i} \right\rfloor \times Q_i + W_i(t - \tilde{t}, t) \\ &= r_i(t - s_3 - (\tilde{t} + \Omega_i)) + W_i(t - \tilde{t}, t) \\ &\geq r_i(t - s_3 - (\Omega_i + T_i - q_i)). \end{aligned}$$

The last inequality holds when the worst case, $\tilde{t} = T_i - q_i$ and $W_i(t - \tilde{t}, t) = 0$, occurs. \square

The restriction Lemma 1 imposed is that the service provided to flow i from the beginning of its BST3 is lower bounded. Next, we show that the arrived packets within a BST3 in a NE complete their services no later than Θ_i after the BST3 ends.

Lemma 2. Let $[s_3, e_3)$ be a BST3. All flow i packets that arrived during $[s_3, e_3)$ will be served by time $e_3 + \Theta_i$.

Proof. Assume that the last packet of the BST3 completes its service at time t . Then the amount of service offered to the flow until time t is equal to the amount of traffic that arrived from the flow until e_3 . As s_3 is the beginning of the BST3,

$$A_i(s_3, e_3) = S_i(s_3, t). \quad (5)$$

From the definition of BST3 that

$$A_i(s_3, e_3) = r_i(e_3 - s_3). \quad (6)$$

From Lemma 1 and Eqs. (5) and (6)

$$r_i(t - s_3 - \Theta_i) - r_i(e_3 - s_3) \leq 0 \quad (7)$$

or equivalently,

$$t \leq e_3 + \Theta_i. \quad \square \quad (8)$$

The following two theorems bind the queuing delays as well as the buffer requirements for flow i within an NE.

Theorem 2. In a network element, the maximum delay D_i among packets arrived during BST3 of flow i is

² It follows the schedulability test listed in Eq. (2).

bounded as

$$D_i \leq \frac{\sigma_i - Q_i}{r_i} \frac{R_i - r_i}{R_i - \rho_i} + T_i + \Theta_i. \quad (9)$$

Proof. Assume that a packet arrived at time t^* during the BST3 suffers the maximum delay D_i . This means that the packet was served at time $t^* + D_i$, $t^* + D_i \leq e_3 + \Theta_i$. Hence, the amount of service offered to the flow i until time $t^* + D_i$ is equal to its traffic amount arrived up to time t^* . As s_3 is the beginning of the BST3,

$$S_i(s_3, t^* + D_i) = A_i(s_3, t^*). \quad (10)$$

From Eq. (3), we can derive

$$S_i(s_3, t^* + D_i) \leq \min(\sigma_i + \rho_i(t^* - s_3), Q_i + R_i(t^* - s_3)). \quad (11)$$

From Eq. (4), we have $D_i \geq \Theta_i$ and

$$S_i(s_3, t^* + D_i) \geq r_i(t^* + D_i - s_3 - \Theta_i). \quad (12)$$

From Eqs. (11) and (12)

$$r_i(t^* + D_i - s_3 - \Theta_i) \leq \min(\sigma_i + \rho_i(t^* - s_3), Q_i + R_i(t^* - s_3)). \quad (13)$$

Case 1: When $Q_i + R_i(t^* - s_3) \leq \sigma_i + \rho_i(t^* - s_3)$

$$D_i \leq \frac{R_i - r_i}{r_i} (t^* - s_3) + \frac{Q_i}{r_i} + \Theta_i; \quad (14)$$

and

$$Q_i + R_i(t^* - s_3) \leq \sigma_i + \rho_i(t^* - s_3),$$

$$t^* - s_3 \leq \frac{\sigma_i - Q_i}{R_i - \rho_i}.$$

Substituting $(t^* - s_3)$ in Eq. (14), it becomes

$$D_i \leq \frac{\sigma_i - Q_i}{r_i} \frac{R_i - r_i}{R_i - \rho_i} + \frac{Q_i}{r_i} + \Theta_i. \quad (15)$$

Case 2: When $Q_i + R_i(t^* - s_3) > \sigma_i + \rho_i(t^* - s_3)$,

$$D_i \leq \frac{\sigma_i}{r_i} - \frac{r_i - \rho_i}{r_i} (t^* - s_3) + \Theta_i \quad (16)$$

and

$$Q_i + R_i(t^* - s_3) > \sigma_i + \rho_i(t^* - s_3),$$

$$t^* - s_3 > \frac{\sigma_i - Q_i}{R_i - \rho_i}.$$

Substituting $(t^* - s_3)$ in Eq. (16), it becomes

$$\begin{aligned} D_i &\leq \frac{\sigma_i}{r_i} - \frac{r_i - \rho_i}{r_i} \frac{\sigma_i - Q_i}{R_i - \rho_i} + \Theta_i; \\ &= \frac{\sigma_i - Q_i}{r_i} \frac{R_i - r_i}{R_i - \rho_i} + \frac{Q_i}{r_i} + \Theta_i. \quad \square \end{aligned} \quad (17)$$

Theorem 3. If $B_i(t)$ is the backlog of flow i at time t , then $\forall t \in [s_3, e_3 + \Theta_i)$

$$\begin{aligned} B_i(t) &\leq \max(2Q_i + R_i(T_i - q_i), Q_i + \frac{\sigma_i - Q_i}{R_i - \rho_i}(R_i - r_i) \\ &\quad + r_i\Theta_i, Q_i + \sigma_i + \rho_i(T_i - q_i), \sigma_i + \rho_i\Theta_i). \end{aligned} \quad (18)$$

Proof. The backlog of flow i at time t in a BST3 is equal to the backlog at time s_3^- plus the traffic arrived during the interval $[s_3, t)$ then minus the amount of service received by flow i during the same time interval. That is

$$B_i(t) = B_i(s_3) + A_i(s_3, t) - W_i(s_3, t).$$

Suppose that s_3 falls in the k th time-frame of the n th busy period, that is, the time interval $[t_n^k, t_n^k + T_i)$. In the k th time-frame, there are q_i service chances during the first Ω_i time-slots. In order to derive the maximum backlog, let s_3 be a time instant before $t_n^k + \Omega_i - q_i$ and no cell be served before s_3 within the same time-frame. We can derive $B_i(s_3) = B_i(t_n^k) + A_i(t_n^k, s_3)$, hence $B_i(t) = B_i(t_n^k) + A_i(t_n^k, s_3) + A_i(s_3, t) - W_i(s_3, t)$. From Definition 4, the right-hand side of the equality reaches its maximum when $s_3 = t_n^k$. From Eq. (3), we consider the two following cases:

Case 1: When $Q_i + R_i(t - s_3) < \sigma_i + \rho_i(t - s_3)$, we get $A_i(s_3, t) \leq Q_i + R_i(t - s_3)$ and

$$\frac{\sigma_i - Q_i}{R_i - \rho_i} > t - s_3. \quad (19)$$

1. If $t < s_3 + \Omega_i$ then the maximum backlog happens when the cells are served during the last q_i time slots before $s_3 + \Omega_i$, hence, $W_i(s_3, s_3 + \Omega_i - q_i) = 0$.

$$\begin{aligned} B_i(t) &= B_i(s_3) + A_i(s_3, t) - W_i(s_3, t) \\ &\leq B_i(s_3) + A_i(s_3, s_3 + \Omega_i - q_i) \\ &\leq B_i(s_3) + Q_i + R_i(\Omega_i - q_i) \leq 2Q_i + R_i(T_i - q_i). \end{aligned} \quad (20)$$

The last inequality follows from the restrictions $B_i(s_3) \leq Q_i$ and $\Omega_i \leq T_i$.

2. If $s_3 + \Omega_i \leq t < s_3 + \Theta_i$ then all cells arrived before t_n^k will be served. Therefore, $B_i(t) = A_i(s_3, t) - S_i(s_3, t)$ and $S_i(s_3, t) \geq 0$ from Eq. (4).

$$\begin{aligned} B_i(t) &= A_i(s_3, t) - S_i(s_3, t) \leq Q_i + R_i(t - s_3) \\ &\leq Q_i + R_i\Theta_i. \end{aligned} \quad (21)$$

3. If $t \geq s_3 + \Theta_i$, we have $B_i(t) = A_i(s_3, t) - S_i(s_3, t)$ and

$$\begin{aligned} B_i(t) &= A_i(s_3, t) - S_i(s_3, t) \\ &\leq Q_i + R_i(t - s_3) - r_i(t - s_3 - \Theta_i) \\ &= Q_i + (R_i - r_i)(t - s_3) + r_i\Theta_i \\ &\leq Q_i + \frac{\sigma_i - Q_i}{R_i - \rho_i}(R_i - r_i) + r_i\Theta_i. \end{aligned} \quad (22)$$

As $Q_i + (R_i - r_i)(t - s_3) + r_i\Theta_i \geq Q_i + (R_i - r_i)\Theta_i + r_i\Theta_i = Q_i + R_i\Theta_i$, we have $B_i(t) \geq Q_i + R_i\Theta_i$.

Case 2: When $Q_i + R_i(t - s_3) \geq \sigma_i + \rho_i(t - s_3)$, we get $A_i(s_3, t) \leq \sigma_i + \rho_i(t - s_3)$ and

$$\frac{\sigma_i - Q_i}{R_i - \rho_i} \leq t - s_3. \quad (23)$$

Following the proof of case 1, we can show the following:

1. If $t < s_3 + \Omega_i$ then,

$$\begin{aligned} B_i(t) &\leq B_i(s_3) + A_i(s_3, s_3 + \Omega_i - q_i) \\ &\leq B_i(s_3) + \sigma_i + \rho_i(\Omega_i - q_i) \\ &\leq Q_i + \sigma_i + \rho_i(T_i - q_i). \end{aligned} \quad (24)$$

2. If $s_3 + \Omega_i \leq t < s_3 + \Theta_i$ then

$$B_i(t) \leq A_i(s_3, t) \leq \sigma_i + \rho_i(t - s_3) \leq \sigma_i + \rho_i\Theta_i. \quad (25)$$

3. If $t \geq s_3 + \Theta_i$ then

$$\begin{aligned} B_i(t) &= A_i(s_3, t) - S_i(s_3, t) \\ &\leq \sigma_i + \rho_i(t - s_3) - r_i(t - s_3 - \Theta_i) \leq \sigma_i + \rho_i\Theta_i. \end{aligned} \quad (26)$$

The last inequality follows from the restriction $r_i \geq \rho_i$.

□

4.2. Delay and backlog analysis of a network of FBPS servers

In the previous section we analyzed the delay behavior of

a flow when a single FBPS server is considered. We will now proceed to prove bounds on both backlog and delay over multiple nodes. Assume that flow i travels through K Network Elements (NEs) in ATM backbone. The k th NE is denoted by NE_k for $1 \leq k \leq K$. NE_1 is the ingress and NE_K is the egress. The FBPS scheduler is applied to all K NEs. The packet delay of flow i in the ATM backbone is counted from the time a packet enters the output queue in the ingress till the time it leaves the egress. Suppose that D_i denotes the maximum delay of a flow i in a network of K NEs. The propagation delay from NE_{l-1} to NE_l , $2 \leq l \leq K$, is denoted by $\beta_{l-1,l}$. $S_i^k(t, \tau)$ represents the amount of service provided by NE_k to flow i packets that arrived after time t till time τ .

FBPS does not need a global clock. At each NE, the time-framing is formed whenever the first cell of a busy period arrives. As FBPS is work-conserving, it is possible that time-frames have different sizes. The maximum time-frame size for flow i is T_i , and a time-frame size smaller than T_i means that flow i can share a bandwidth larger than r_i . The maximum end-to-end delay occurs when all time-frames have fixed size T_i in every NE along the path of flow i . According to the definition of busy period, certain cells are served during every time-frame in a busy period. Under such situation, we can prove the following lemma.

Lemma 3. *For cells which leave NE_1 during the h th time-frame of the first busy period of flow i will be completely transmitted by NE_k before time $t_1 + hT_i + (k-1)T_i + \sum_{j=2}^k \beta_{j-1,j}$.*

Proof. We will prove the lemma through induction on k .

Basic step: At first, we have to prove that flow i cells which leave NE_1 during the h th time-frame will be served by NE_2 before time $t_1 + (h+1)T_i + \beta_{1,2}$.

Cells which are served during the first time-frame of the first busy period will depart NE_1 before $t_1 + T_i$, and these cells will arrive at NE_2 by time $t_1 + T_i + \beta_{1,2}$. At least Q_i cells can depart NE_2 before time $t_1 + 2T_i + \beta_{1,2}$. The amount of cells that come from NE_1 till time $t_1 + T_i + \beta_{1,2}$ is no more than Q_i . Hence, these cells can leave NE_2 before time $t_1 + 2T_i + \beta_{1,2}$.

Assume that cells served by NE_1 during the $(h-1)$ th time-frame will be served by NE_2 before time $t_1 + hT_i + \beta_{1,2}$. These cells served by NE_1 during h th time-frame will arrive at NE_2 before time $t_1 + hT_i + \beta_{1,2}$. At that time, all cells from NE_1 before h th time-frame have already left NE_2 . Therefore, cells, which are served by NE_1 during the h th time-frame, will leave NE_2 before time $t_1 + (h+1)T_i + \beta_{1,2}$.

Inductive step: We assume that the lemma holds at NE_{k-1} . That is, the cells, which depart from NE_1 during the h th time-frame, will be served by NE_{k-1} before time $t_1 + hT_i + (k-2)T_i + \sum_{j=2}^{k-1} \beta_{j-1,j}$. By the similar argument, it

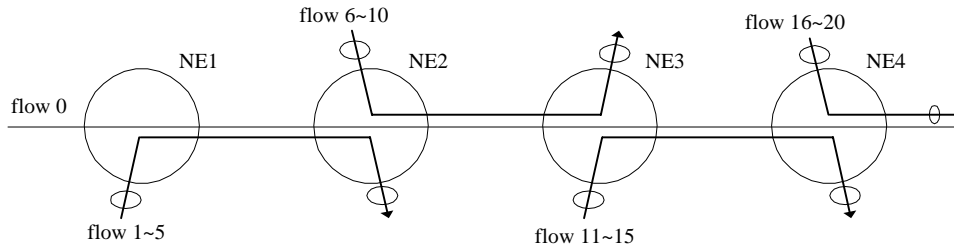


Fig. 5. Network configuration used for simulation.

is easy to see that cells which depart from NE₁ during the *h*th time-frame will be served by NE_{*k*} before time $t_1 + hT_i + (k - 1)T_i + \sum_{j=2}^k \beta_{j-1,j}$. □

Now we can extend the above lemma to every busy period of flow *i*.

Theorem 4. For flow *i* cells which leave NE₁ during the *h*th time-frame of the *n*th busy period will be completely transmitted by NE_{*k*} before time $t_n + hT_i + (k - 1)T_i + \sum_{j=2}^k \beta_{j-1,j}$.

Proof. From the above lemma, cells served by NE₁ during the first busy period of flow *i* will leave NE₂ before time $t'_1 + T_i + \beta_{1,2}$, and these cells served during the first time-frame of the second busy period of flow *i* will arrive at NE₂ before time $t_2 + T_i + \beta_{1,2}$. As $t_2 > t'_1$, these cells are able to leave NE₂ before time $t_2 + 2T_i + \beta_{1,2}$. Then by the similar proof of above lemma, it can be shown that cells which leave NE₁ during the *h*th time-frame of the second busy period of flow *i* will be completely transmitted by NE_{*k*} before time $t_2 + hT_i + (k - 1)T_i + \sum_{j=2}^k \beta_{j-1,j}$.

We omit the rest of the proof because it can be proved easily by induction on *n*. □

Using the above theorem, the end-to-end delay of flow *i* can be bounded if every time-frame has equal size *T_i* in every NE. The following theorem can be derived from Theorems 2 and 4.

Theorem 5. The maximum end-to-end delay *D_i* of a flow *i*

in a network of *K* NEs with FBPS is bounded as

$$D_i \leq \frac{(\sigma_i - Q_i)}{r_i} \frac{R_i - r_i}{R_i - \rho_i} + KT_i + \Theta_i + \sum_{k=2}^K \beta_{k-1,k}. \quad (27)$$

Stiliadis and Varma proposed a general model, called Latency-Rate Servers (LR-servers) [16], for the analysis of traffic scheduling algorithms in broadband packet networks. The behavior of an LR scheduler is determined by two parameters—the latency and the allocated rate. Several well-known scheduling algorithms, such as PGPS, SCFQ, VC, DRR and WRR belong to the class of LR-servers. We have already proved that FBPS also belongs to the class of LR-servers in Lemma 1. Let Θ_i^k denote the latency of the *k*th NE along the path of a flow, then according to Theorem 2 in Ref. [16] and our Theorem 5, we can derive $\Theta_i^1 = \Theta_i$ and $\Theta_i^k = T_i + \beta_{k-1,k}$ for $k \in 2, \dots, K$. From Corollary 1 in Ref. [16]

$$S_i^k(s_2, t) \geq \max \left(0, r_i \left(s_2 - t - \sum_{j=1}^k \Theta_i^j \right) \right). \quad (28)$$

We can treat the network from NE₁ to NE_{*k*} as equivalent to a single LR-server with latency equal to the sum of their latencies. We can now state the following theorem that bounds the backlog of flow *i* in each node of the network.

Theorem 6. The maximum backlog $B_i^k(t)$ in the *k*th node of

Table 1
Simulation parameters

Flow	Mean burst length (cells)	Mean silence period (cell-time-slot)	σ_i (cells)	ρ_i	r_i
0,5,10,15,20	50	800	100	1/16	1/16
1,2,6,7,11,12,16,17	70	490	210	1/7	1/7
3,4,8,9,13,14,18,19	100	400	300	1/4	1/4

Table 2
Flow characteristics

Flow	r_i	Q_i (cells)	T_i (cell-time-slot)
3,4,8,9,13,14,18,19	1/4	32	128
1,2,6,7,11,12,16,17	1/7	32	224
0,5,10,15,20	1/16	32	512

a flow i at time t is bounded as

$$B_i^k(t) \leq \max \left(2Q_i + R_i(T_i - q_i), Q_i + \frac{\sigma_i - Q_i}{R_i - \rho_i} (R_i - r_i) \right. \\ \left. + r_i \sum_{j=1}^k \Theta_i^j, \right. \\ \left. Q_i + \sigma_i + \rho_i(T_i - q_i), \sigma_i + \rho_i \sum_{j=1}^k \Theta_i^j \right). \quad (29)$$

Proof. From Eq. (28), we can treat NE_k as the first NE of the network with latency equal to $\sum_{j=1}^k \Theta_i^j$, then by Theorem 3, this theorem can be proved. \square

4.3. Fairness of a FBPS scheduler

There are different methods to estimate the fairness of a scheduling algorithm. The fairness parameter we used is based on Golestani's work [6] for the analysis of self-clock fair queuing. We call the fraction $W_i(t, \tau)/r_i$ the *normalized service* offered to flow i in the interval $[t, \tau)$. Golestani [6] suggested that the difference in normalized service offered to any two flows is used as the measure of fairness for the scheduling algorithm. More precisely, for any two flows i and j that are continuously backlogged in an interval of time $[\tau_1, \tau_2)$, the fairness of the scheduler is \mathcal{F} which satisfies $|(W_i(\tau_1, \tau_2)/r_i) - (W_j(\tau_1, \tau_2)/r_j)| \leq \mathcal{F}$.

Theorem 7. For an FBPS scheduler,

$$\mathcal{F} = 2(T_i + T_j - q_i - q_j).$$

Proof. Suppose that the amount of service offered to flow i

and flow j during the time interval $[\tau_1, \tau_2)$ is $m_i \times Q_i$ and $m_j \times Q_j$, respectively. First, we will prove that $|m_i T_i - m_j T_j| \leq 2(T_i + T_j - q_i - q_j)$. As FBPS is work-conserving, let T_c be the amount of time that the server is idle while $Queue_i$ and $Queue_j$ are backlogged in $[\tau_1, \tau_2)$. With m_i and m_j , the following two inequalities must hold:

$$(m_i - 2)T_i + 2q_i - T_c \leq (\tau_2 - \tau_1) \leq (m_i + 2)T_i - 2q_i - T_c,$$

and

$$(m_j - 2)T_j + 2q_j - T_c \leq (\tau_2 - \tau_1) \leq (m_j + 2)T_j - 2q_j - T_c.$$

Therefore, we can obtain

$$(\tau_2 - \tau_1) - 2T_i + 2q_i + T_c \leq m_i T_i$$

$$\leq (\tau_2 - \tau_1) + 2T_i - 2q_i + T_c,$$

and

$$(\tau_2 - \tau_1) - 2T_j + 2q_j + T_c \leq m_j T_j$$

$$\leq (\tau_2 - \tau_1) + 2T_j - 2q_j + T_c.$$

From above inequalities, we can derive

$$-2(T_i + T_j - q_i - q_j) \leq m_i T_i - m_j T_j$$

$$\leq 2(T_i + T_j - q_i - q_j),$$

thus

$$|m_i T_i - m_j T_j| \leq 2(T_i + T_j - q_i - q_j). \quad (30)$$

The difference in normalized service offered to these two flows is

$$\frac{W_i(\tau_1, \tau_2)}{r_i} - \frac{W_j(\tau_1, \tau_2)}{r_j} = \frac{m_i Q_i}{r_i} - \frac{m_j Q_j}{r_j} = m_i T_i - m_j T_j.$$

From Eq. (30)

$$\left| \frac{W_i(\tau_1, \tau_2)}{r_i} - \frac{W_j(\tau_1, \tau_2)}{r_j} \right| \leq 2(T_i + T_j - q_i - q_j). \quad \square$$

5. Simulation and numerical results

In this section we present some simulation results to verify our analytical bound. The performance metrics we

Table 3
The maximum delays of flow 0 to flow 5 in node 1. Only flow 1 is misbehaving

Flow	Reserved bandwidth	Arrival rate	Maximum delay (cell-time-slot)	Delay bound (cell-time-slot)
0	0.0625 (1/16)	0.06243	1783	2464
1	0.14286 (1/7)	0.24524	N/A	N/A
2	0.14286 (1/7)	0.13694	1263	1854
3	0.25 (1/4)	0.24772	1111	1360
4	0.25 (1/4)	0.24527	1099	1360
5	0.0625 (1/16)	0.06246	1761	2464

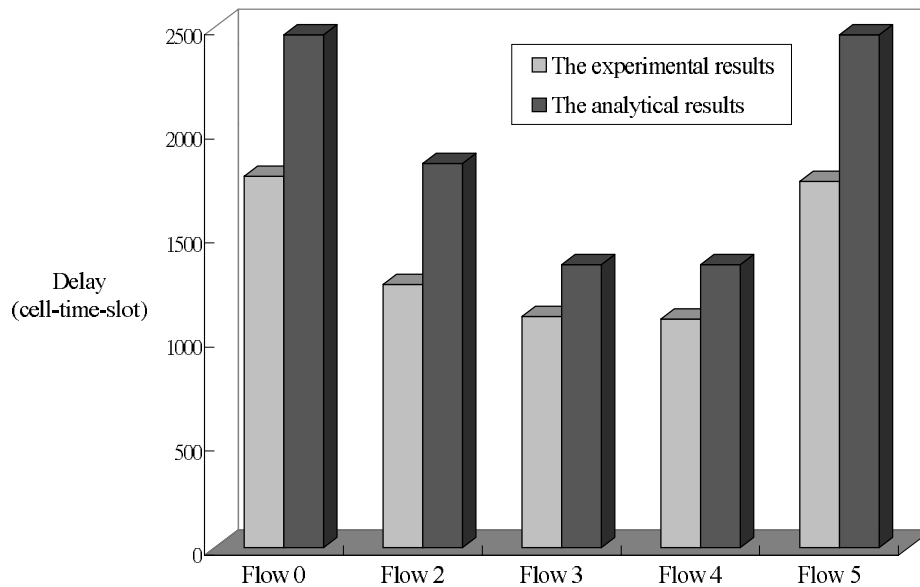


Fig. 6. Comparison of experimental maximum packet delays and analytical upper-bounds.

focus on are the average and maximum packet delays experienced by the traffic flows during the simulation. We present results from simulating FBPS algorithm in a multi-hop network configuration.

We have simulated FBPS algorithm in a 4-hop network model as shown in Fig. 5. It consists of four network elements (NEs), where NE1 is the ingress, NE4 is the egress and the others are ATM switches. The network model chosen is a “parking lot” configuration where a flow passes through four NEs in series and shares the outgoing link at each hop with local cross-traffic transmitted from one NE to the next. The 4-hop flow shares the outgoing link at each node with five other cross traffic flows. One of these flows at each node was set to misbehave which means the flow remains backlogged throughout the simulation. An ON-OFF traffic model was used to generate traffic within each flow. Both the burst length and the silence period of the traffic model were drawn from a geometric distribution. Traffic is shaped through a leaky bucket before entering the output buffer of NE1. The traffic parameters, leaky bucket parameters and reserved bandwidth of each flows are listed in Table 1. Both the leaky bucket rate (ρ_i) and the reserved bandwidth (r_i) are described in fraction of link bandwidth.

In our simulation, the maximum packet size is chosen as 1500 bytes, or 32 cells correspondingly, which represents

the MTU size of Ethernet. The burst of cells are partitioned into packets with size no greater than 32 cells. The packet delay is calculated from the arrival time of the last cell of a packet to the departure time of that cell. The characteristics (T_i, Q_i) of all flows are listed in Table 2.

Among these 21 flows, flow 1, flow 6, flow 11 and flow 16 were set to misbehave. At first, we examine the delay behavior of flow 0 to flow 5 in node 1. A summary of our results is presented in Table 3 and the corresponding illustration is shown in Fig. 6. The upper-bounds for delay for each flow in node 1 are computed using Theorem 2. Delays are shown in the tables in terms of cell-time-slots. It is clear that experimental maximum delays are bounded by our analytical upper-bound.

Table 4 provides the maximum delays seen by the 4-hop flow (i.e. Flow 0) at each NE. The maximum end-to-end delay of flow 0 is 2874 that is smaller than the analytical delay bound, 4000, which can be computed using Theorem 5. Fig. 7 demonstrates both the experimental delays and the analytical upper-bounds at each node. It is clear that the experimental results are bounded by the analytical upper-bounds. The differences between the experiment results and the analytical upper-bounds are understandable because the performance analyses are done under worst case assumption.

Table 4
The maximum delays seen by flow 0 at each NE

NE	Total traffic load	Maximum delay
1	1.00006	1783
2	1.00004	2287
3	1.00009	2505
4	1.00006	2874

6. Conclusion

In this paper, we introduced and analyzed a frame-based priority scheduling algorithm with application in hybrid IP/ATM networks. The data transmission of our algorithm is cell based with packet-level QoS guarantee. As presented in Section 4, the algorithm provided a reasonable end-to-end packet delay bound if the input traffic is leaky-bucket

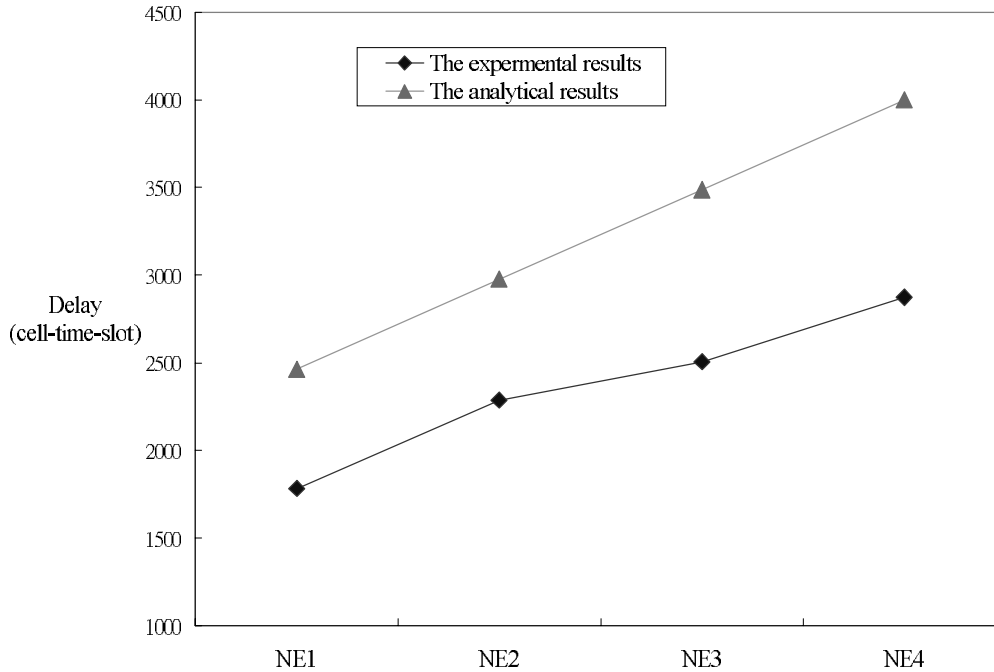


Fig. 7. Comparison of maximum delays seen by Flow 0 and analytical upper-bounds at each NE.

shaped. We also analyzed the fairness property of the algorithm, and showed that the difference in normalized service offered to any two connections that are continuously backlogged is always bounded. The main advantages of the algorithm is its simplicity, which allows the algorithm to be implemented in a simple and efficient manner, also the delay behavior is insensitivity to traffic patterns of other flows and is independent of the number of flows sharing the same output link. Besides, all information needed for the algorithm can be extracted from the scheduler itself.

The design of a traffic scheduling algorithm involves an inevitable tradeoff among its delay, implementation complexity, and fairness. Among the three, the delay and implementation complexity are clearly the most important criteria for the effectiveness of an algorithm in a real system. PGPS is considered to be the most efficient scheduling scheme but it is too complex to be realized in high speed networks. SCFQ simplifies the time-stamp computation of PGPS considerably, however, it provides delay bounds that depend on the number of flows sharing the output link, which causes serious degradation of its delay behavior. Virtual Clock (VC) has both simpler implementation and identical delay bounds compared with PGPS, but it has unbounded fairness. Our frame-based priority scheduling features satisfactory packet delay and simple implementation, as well as provides good fairness to individual flows.

Appendix A

The general behaviors of FBPS scheduler are described in the following algorithm. The processing performed by the

algorithm can be divided into three parts: (1) the part that is performed at the boundaries of cell time slots to update time-frame counters; (2) the part that is performed when a new cell arrives; and (3) the part that is executed when the transmission of a cell has been completed. The process of counter updating and selecting the eligible cell with highest priority can be implemented in hardware easily. $T_{\text{shiftback}}$ needs to be calculated only when there are active flows but no eligible cell.

FBPS algorithm

Declaration:

- N : the number of flows
- (Q_i, T_i) : the characteristics of flow i , $1 \leq i \leq N$
- TFC_i : time-frame counter of flow i
- CC_i : credit counter of flow i
- $Queue_i$: the per-VC queue associated with flow f_i
- $Active_i$: the active flag, it is true if flow i is active, and false otherwise

Initialization:

- For $i = 1$ to N
- $TFC_i = 0$; $CC_i = 0$; $Active_i = \text{False}$;

Procedure Adjust: at the boundaries of cell time slots

1. Begin
2. For ($i = 1$; $i \leq N$; $i++$)
3. If ($TFC_i > 0$)
4. TFC_i-- ;
5. If ($TFC_i == 0$ and $Active_i$)

6. $TFC_i = T_i;$
7. $CC_i = Q_i;$
8. Else If ($TFC_i == 0$ and not *Active_i*)
9. $CC_i = 0;$
10. End

Procedure Enqueue(i): on arrival of a cell

1. Begin
2. If (not *Active_i*)
3. If ($TFC_i == 0$)
4. $TFC_i = T_i;$
5. $CC_i = Q_i;$
6. *Active_i* = True;
7. add the arrival cell in the tail of *Queue_i*;
8. End

Procedure Schedule:

1. Begin
2. For ($i = 1; i \leq N; i++$)
3. If (*Active_i* and $CC_i > 0$)
4. retrieve cell from head of *Queue_i* and transmit;
5. $CC_i--;$
6. If (*Queue_i* is empty)
7. *Active_i* = False;
8. quit the procedure;
9. $T_{shiftback} = \min_{\text{active flow } j} (TFC_j);$
10. For ($i = 1; i \leq N; i++$)
11. If (*Active_i* and $TFC_i = TFC_i - T_{shiftback} == 0$)
12. $TFC_i = T_i;$
13. $CC_i = Q_i;$
14. GoTo 2;
15. End

References

- [1] M.W. Garrett, M. Borden, Interoperation of controlled-load service and guaranteed service with ATM, Internet-Draft, July 1997.
- [2] Multi-Protocol Over ATM Version 1.0, The ATM Forum Technical Committee, May 1997.
- [3] LAN Emulation Over ATM Specification—Version 1, ATM Forum Specification, February 1995.
- [4] D. Katz, D. Piscitello, NBMA next hop resolution protocol (NHRP), Internet-Draft, May 1995.
- [5] S. Shenker, C. Partridge, R. Gurein, Specification of guaranteed quality of service, RFC 2212, September 1997.
- [6] E. Guarene, P. Fasano, V. Vercellone, IP and ATM integration perspectives, IEEE Communication Magazine January (1998) 74–80.
- [7] H. Zhang, Service disciplines for guaranteed performance service in packet-switching networks, Proc IEEE 83 (10) (1995) 1374–1396.
- [8] A.K. Parekh, R.G. Gallager, Generalized processor sharing approach to flow control in integrated services networks: the single-node case, IEEE/ACM Trans. Networking 1 (3) (1993) 344–357.
- [9] A.K. Parekh, R.G. Gallager, Generalized processor sharing approach to flow control in integrated services networks: the multiple node case, IEEE/ACM Trans. Networking 2 (2) (1994) 137–150.
- [10] L. Zhang, VirtualClock: a new traffic control algorithm for packet switching networks, ACM SIGCOM (1990) 19–29.
- [11] J.C. Bennett, H. Zhang, Worse-case fair weighted fair queueing, IEEE INFOCOM (1996) 120–127.
- [12] S.J. Golestani, A self-clocked fair queueing scheme for broadband applications, IEEE INFOCOM (1994) 636–645.
- [13] M. Shreedhar, G. Varghese, Efficient fair queueing using deficit round-robin, IEEE/ACM Trans. Networking 4 (3) (1996) 375–385.
- [14] J. Lehoczy, L. Sha, The rate monotonic scheduling algorithm: exact characterization and average case behavior, Proceedings of the 10th IEEE Real-Time System Symposium, 1989, pp. 166–1719.
- [15] D. Stiliadis, A. Varma, Latency-rate serves: a general model for analysis of traffic scheduling algorithms, IEEE INFOCOM (1994) 111–119.
- [16] J. Heinanen, Multiprotocol encapsulation over ATM adaptation layer 5, RFC 1483, July 1993.