Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Computer Networks 43 (2003) 669–694

COMPUTER NETWORKS

www.elsevier.com/locate/comnet

A generalized hierarchical fair service curve algorithm for high network utilization and link-sharing

Kihyun Pyun *, Junehwa Song, Heung-Kyu Lee

Department of Electrical Engineering and Computer Science, KAIST, 373-1 Kusong-dong, Yusong-gu, Taejon 305-701, South Korea

Received 27 September 2002; received in revised form 8 April 2003; accepted 27 June 2003

Responsible Editor: N. Fonseca

Abstract

The number of real-time applications, such as video-on-demand and video conferencing, is rapidly increasing. Real-time data now occupies a significant portion of network traffic. These applications require real-time service; as such, they need to bound end-to-end delays. Generally, real-time service is provided by reserving bandwidth in advance. Thus, to admit a large number of real-time applications, bandwidth needs to be reserved efficiently. In addition, to share limited bandwidth effectively among multiple agencies, a link-sharing service is required. A link-sharing service provides a guaranteed share of link bandwidth to each agency. To provide both real-time and link-sharing services simultaneously, several hierarchical link-sharing schemes have been proposed. However, existing hierarchical schemes still have problems in achieving a high level of network utilization, especially when variable-bit-rate (VBR) video sessions participate in a hierarchy. The problem is serious since a large portion of real-time applications transmit MPEG-coded VBR data. We propose a new scheduling algorithm that can achieve a high level of network utilization even with VBR video sessions. The proposed scheduling algorithm is a hierarchical link-sharing scheme; it is an extension of the H-FSC generalized with a new bandwidth reservation scheme. The new scheme provides tight bandwidth reservations for VBR video sessions. Even with the advantage in network utilization, the generalized H-FSC does not require high overhead for scheduling; the scheduling complexity remains the same as that of the H-FSC.

Keywords: Packet scheduling; Guaranteed service; Quality-of-service; Service curve

1. Introduction

The number of real-time applications, such as video-on-demand and video conferencing, is rapidly increasing. Real-time data now occupies a significant portion of network traffic. These applications require real-time service; as such, they need to bound end-to-end delays. Generally, real-time service is provided by reserving bandwidth in advance [19]. Thus, to admit a large number of real-time applications, bandwidth needs to be reserved efficiently. In addition, to share limited bandwidth effectively among multiple agencies, a link-sharing service is required [6]. A link-sharing

* Corresponding author.
E-mail addresses: khpyun@rtlab.kaist.ac.kr (K. Pyun), junesong@nclab.kaist.ac.kr (J. Song), hklee@rtlab.kaist.ac.kr (H.-K. Lee).

1389-1286/$ - see front matter
doi:10.1016/S1389-1286(03)00335-9

service provides a guaranteed share of link bandwidth to each agency. An agency can represent data traffic from an organization, those using a protocol such as IP or SNA, or a specific traffic type such as FTP traffic or telnet traffic.

To provide both real-time and link-sharing services simultaneously, several hierarchical link-sharing schemes have been proposed, including class-based queueing (CBQ) [6], hierarchical packet fair queueing (H-PFQ) [1] and hierarchical fair service curve (H-FSC) [22]. In hierarchical schemes, bandwidth sharing among different agencies is represented via a hierarchy. Fig. 1 provides an example of this. In the hierarchy, each node represents an agency. A hierarchy specifies the bandwidth share for each node, which should be guaranteed for each agency, and further represents how remaining bandwidth, if any, is shared among different agencies. If a share of bandwidth is specified for each node, hierarchical schemes guarantee the specified bandwidths if there is sufficient demand. In addition, if nodes do not fully use their share of bandwidth, the excess bandwidth is distributed to other nodes by the specified bandwidth ratios in the hierarchy.

Although existing hierarchical schemes satisfy both requirements of real-time and link-sharing services, they still have problems in achieving a high level of network utilization. (Here, network utilization indicates how many real-time sessions can be admitted.) This is particularly true when variable-bit-rate (VBR) video sessions participate in a hierarchy. This problem is serious since a large portion of real-time applications transmit MPEG-coded VBR data. In this paper, we propose a new scheduling algorithm that can achieve a high level of network utilization even with VBR video sessions. The proposed scheduling algorithm is a hierarchical link-sharing scheme; it is an extension of the H-FSC generalized with a new bandwidth reservation scheme. The new scheme provides tight bandwidth reservations for VBR video sessions. Even with this advantage in network utilization, the generalized H-FSC does not require high

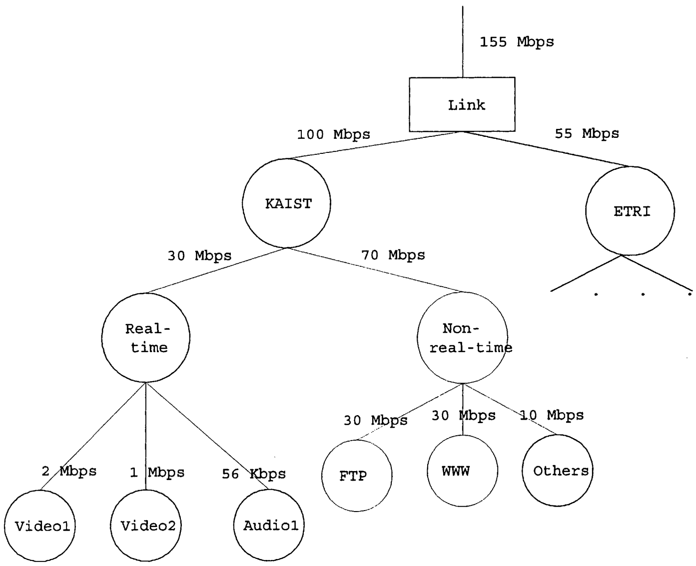

Fig. 1. An example of a link-sharing hierarchy: In the figure, two organizations, KAIST and ETRI, share a 155 Mbps link. All data traffic from applications in KAIST are guaranteed to receive 100 Mbps if there is sufficient demand. Real-time traffic from KAIST is guaranteed to consume bandwidth up to 30 Mbps. Non-real-time traffic from KAIST, FTP, WWW, and other traffic types are guaranteed to receive 30, 30, and 10 Mbps, respectively. If applications from ETRI do not fully use the specified bandwidth, the remaining bandwidth is distributed to real-time and non-real-time traffic from KAIST in a ratio of 3–7. Similarly, if there is no FTP traffic from KAIST, the remaining bandwidth is distributed to WWW and other traffic from KAIST in a ratio of 3–1.

overhead for scheduling; the scheduling complexity remains the same as that of the H-FSC.

Among the various hierarchical link-sharing schemes, H-FSC can achieve higher network utilization than other schemes such as H-PFQ and CBQ. Two reasons account for the higher network utilization. First, H-FSC can reserve bandwidth for each session independently of the delay bound for the session. Second, H-FSC gives preference to real-time applications. That is, whenever there is a potential danger to violate the delay bounds of some real-time packets, the real-time packets are transmitted with higher priority. Within a short interval, this may result in a violation of guaranteeing a specified bandwidth. However, for long intervals, the violation disappears. On the contrary, H-PFQ gives priority to guaranteeing the specified bandwidth. In this case, however, delay bounds for real-time applications are increased according to the depth of the link-sharing hierarchy [1,22]. This problem does not occur in H-FSC.

Unfortunately, H-FSC cannot provide tight reservations for VBR video sessions mainly due to inaccurate traffic characterization. H-FSC assumes that each session characterizes its traffic only with a burstiness and an average rate, and reserves bandwidth for each session with those two different rates, i.e., the service rate clearing the burstiness within the delay bound and the average rate. However, this characterization is not effective for VBR videos due to a high variability of frame sizes. If this scheme is applied for VBR video sessions, an excessive amount of bandwidth is unnecessarily reserved. This over-reservation results in a low level of network utilization for VBR video sessions.

Our scheme provides tight bandwidth reservation even for VBR video sessions. This is made possible by using accurate traffic characterization for VBR sessions. For this, a VBR video session is characterized by multiple traffic rates [16,23]. Each session is characterized by different rates on different time intervals. Then, our scheme reserves those multiple rates for each session considering a delay bound for the respective sessions. Note that by considering delay bounds in reservations, the service for each session can be maximally deferred in favor of other sessions. This scheme can reserve bandwidth tightly for VBR sessions, thus achieving a high level of network utilization.

Although the generalized H-FSC provides tight bandwidth reservation, it does not introduce any additional complexity in computation. That is, compared to H-FSC, the generalized H-FSC has the same scheduling complexity. The total complexity is $O(\log N)$ for transmitting a packet with the smallest priority where N is the number of sessions. We note that this complexity can be reduced to $O(1)$ with special hardware such as a sequencer [10] or a systolic array [11].

The generalized H-FSC has advantages over other algorithms that can utilize multiple traffic rates, e.g., the rate-controlled (RC) algorithm adopting the earliest deadline first (EDF) scheduler [8] and the multirate algorithm [16]. First, they are not hierarchical link-sharing schemes. Thus, they cannot guarantee a share of bandwidth according to organizations, protocols, and application types. In addition, the generalized H-FSC has advantages in implementation cost and network utilization. Compared to the RC algorithm, the generalized H-FSC does not need any regulators. A regulator in an EDF algorithm regulates session traffic entering a queue. Note that each session requires multiple regulators for multiple traffic rates to achieve a high level of network utilization [8]. Compared to the multirate algorithm, the generalized H-FSC is superior in network utilization. The multirate algorithm reserves bandwidth with multiple traffic rates. However, it does not consider delay bound for each session in reservations. Thus, the multirate algorithm results in a lower level of network utilization.

In Section 2, we present some background helpful to understand H-FSC. In Section 3, we review H-FSC scheme. In Section 4, we propose our new bandwidth reservation scheme and discuss the network utilization achievable by the scheme. In Section 5, we discuss the scheduling complexity of the new scheme. In Section 6, we present simulation experiments and show that our scheme can achieve a high level of network utilization compared with other algorithms. In Section 7, we describe related works. Lastly, the paper is concluded in Section 8.

2. Background

This section provides some background useful to understand how the proposed algorithm can guarantee delay bounds for real-time sessions. In Section 2.1, we present the network modeling and traffic characterizations used in this paper. In Section 2.2, we introduce the definition of a guaranteed service curve, which represents bandwidth reservation in H-FSC. We present three theorems, which are the basis of service curve allocation. Our proposing allocation scheme is described in Section 4.1.

2.1. Network modeling and traffic characterizations

We model a network by a series of routers. Transmission links are ignored for the convenience of discussion. We denote the amount of data received from session i to a router during a time interval $(s,t]$ by $A_i(s,t)$ and the transmitted amount by $W_i(s,t)$. We define $A_i(s,t) = W_i(s,t) = 0$ if $s \geqslant t$. For notational convenience, we denote $A_i(0,t)$ and $W_i(0,t)$ by $A_i(t)$ and $W_i(t)$, respectively. We divide $W_i(t)$ into two parts, $W_i^{\mathrm{RT}}(t)$ and $W_i^{\mathrm{ES}}(t)$. $W_i^{\mathrm{RT}}(\cdot)$ denotes the amount of data transmitted to meet the delay bound for session i. Similarly, $W_i^{\mathrm{ES}}(\cdot)$ denotes the amount transmitted using extra capacity which remains after meeting all the delay bounds. When we consider multiple routers, we appropriately superscribe each notation to distinguish different routers. For example, $W_i^{m,\mathrm{RT}}(t)$ denotes $W_i^{\mathrm{RT}}(t)$ at the mth router that a session i passes through. It is said that session i is in a backlogged period in a router if there exists a packet to serve from session i.

Each real-time session characterizes its traffic by a rate function $b_i(\cdot)$. Intuitively, $b_i(t)$ indicates the maximum amount of traffic that is allowed to be transmitted from session i during a time interval of length t. Thus, $b_i(\cdot)$ must be a non-decreasing function and we define that $b_i(u) = 0$ for all $u \leqslant 0$. We say that session i is *b_i-smooth* or session i's traffic envelope function is $b_i(\cdot)$ if the amount of data traffic incoming from session i to the network during the interval $(s,t]$ is not greater than $b_i(t-s)$. Also, session i is *expected* to be *b_i-smooth* if $b_i(\cdot)$ is specified in the traffic specification given to the network. As a special case of b_i, we say that session i is *(σ,ρ)-smooth* if $b_i(t) = \sigma + \rho t$. σ represents the burstiness allowed to session i. ρ is the average traffic rate.

We also say that session i is *$K(\sigma,\rho)$-smooth* or session i's traffic envelope function is $K(\sigma,\rho)$ if $b_i(t) = \min_{k=1\cdots K}\{\sigma_i^k + \rho_i^k t\}$. Without loss of generality, we assume that $\sigma_i^1 \leqslant \cdots \leqslant \sigma_i^K$ and $\rho_i^1 \geqslant \cdots \geqslant \rho_i^K$. By using a $K(\sigma,\rho)$ as the traffic envelope function, we can accurately characterize the traffic and its variability of a VBR video [16,23]. Fig. 2 illustrates how the traffic is characterized when a session transmits an MPEG-coded video. An MPEG encoder produces three types of encoded frames: I (intra-coded), P (predicted), and B (bidirectional). We can use the knowledge of the encoding pattern of frames, say IBBPBB, and the size of the largest frames of each type, i.e., I, B, and P to characterize the video traffic. In the figure, the traffic is characterized by either a (σ,ρ) function or a $3(\sigma,\rho)$ function. Using a $3(\sigma,\rho)$ function, the traffic can be characterized more accurately than using a (σ,ρ) function.

2.2. Service curves, delay bound, and backlog bound

Let $S_i(\cdot)$ be a non-decreasing function with $S_i(u) = 0$ for all $u \leqslant 0$. We say that a service curve $S_i(\cdot)$ is guaranteed [1] for session i by a router if, for

[1] Our definition is the same as in [22]. However, our definition is different from that in [18], which is applicable only to a fixed-sized packet environment. In [18], it is said that a service curve $S_i(\cdot)$ is guaranteed for session i by a router if, for *any* time t, there exists a time s, $s \leqslant t$, that satisfies Eq. (1). Although both definitions use the same equation, the respective scheduling algorithms have different implementation implications due to different conditions for the variables s and t. The scheduling algorithm which guarantees service curves using the definition in [18] should check Eq. (1) whenever a packet arrives. In a variable-sized packet environment such as the Internet, the number of packet arrival times can be extraordinarily large even within a short interval. (A packet may arrive at any time since we cannot put any restriction on packet arrival times.) On the contrary, the scheduling algorithm using our definition has only to check the equation whenever each session becomes newly backlogged, not every packet arrival time. Note that the number of backlogged times is significantly smaller than that of packet arrival times.

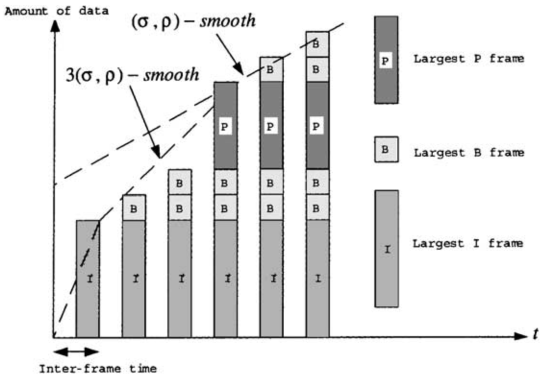

Fig. 2. Two ways to characterize the traffic of an MPEG-coded video. Using a $3(\sigma, \rho)$ function, the traffic can be characterized more accurately than using a (σ, ρ) function.

a packet departure time t of session i, there exists a time s, $s < t$, such that s is the beginning of one of the session's backlogged periods and

$$W_i(t) \geqslant A_i(s) + S_i(t - s). \quad (1)$$

Intuitively, $S_i(t)$ indicates the minimum amount of service, i.e., minimum amount of data transmitted, by the router for session i during an interval t starting from the beginning of a session i's backlogged period.

We further say that a service curve $S_i(\cdot)$ is *strictly* guaranteed for session i if session i is guaranteed $S_i(\cdot)$ without relying on extra capacity for its service. More formally, $S_i(\cdot)$ is strictly guaranteed for session i by a router if, for any packet departure time t, there exists a time s, $s < t$, such that s is the beginning of one of the session's backlogged periods and

$$W_i^{\mathrm{RT}}(t) \geqslant W_i^{\mathrm{RT}}(s) + S_i(t - s). \quad (2)$$

Eq. (2) can be equivalently expressed as follows:

$$W_i^{\mathrm{RT}}(t) \geqslant \min_{s \in B_i(t)} \{W_i^{\mathrm{RT}}(s) + S_i(t - s)\}, \quad (3)$$

where $B_i(t)$ is the set of the start time points of the session i's backlogged periods that is not greater than the packet departure time t.

We can see that a service curve $S_i(\cdot)$ is always guaranteed for a session if it is strictly guaranteed. This is so since, from Eq. (2), $W_i^{\mathrm{RT}}(t)$ is equal to $W_i(t) - W_i^{\mathrm{ES}}(t)$. However, in general, the reverse is not true.

Now, consider multiple routers which a session i passes through. Given service curves which are strictly guaranteed in each of those routers, we can derive a single service curve which is guaranteed by those multiple routers as a whole. In other words, the multiple routers can be reduced to a single composite router (or a network) which guarantees the derived service curve. The derivation is presented in Theorem 1. Intuitively, the theorem tells that *network service curve* $S_i^{\mathrm{net}}(\cdot)$ is constructed by concatenating, in an increasing order of slopes, the line segments of the strictly guaranteed service curves. Note, however, that the network service curve $S_i^{\mathrm{net}}(\cdot)$ constructed as such is *not strictly* guaranteed by the network. (Theorems similar to the following three can be found in [18], however, they are applicable only to a fixed-sized packet environment.)

Theorem 1 (Network service curve theorem). *Suppose that session i passes through M routers in tandem and the mth router,* $1 \leqslant m \leqslant M$, *strictly guarantees* $S_i^m(\cdot)$ *to session i. Then, the M routers guarantee* $S_i^{\mathrm{net}}(\cdot)$ *to session i where*

$$S_i^{\mathrm{net}}(t) \triangleq \min \left\{ \sum_{m=1}^{M} S_i^m(\Delta_m) : \Delta_m > 0 \text{ and } \sum_{m=1}^{M} \Delta_m = t \right\}. \quad (4)$$

Proof. See Appendix A. □

Fig. 3 illustrates the derivation when two routers are given. In the figure, $S_i^1(\cdot)$ and $S_i^2(\cdot)$ are the strictly guaranteed service curves at the first and second routers, respectively. $S_i^1(\cdot)$ has slope s_1, that is less than s_2, during the interval $(0, t_1]$. $S_i^2(\cdot)$ has slope zero until time t_2. The two curves have the same slope s_2 from the times t_1 and t_2, respectively. By Theorem 1, the network service curve $S_i^{\mathrm{net}}(\cdot)$ has slope zero until the time t_2, slope s_1 until the time $(t_1 + t_2)$, and slope s_2 afterwards.

When a router guarantees a service curve for session i, the delay and backlog bound for session i can be derived.

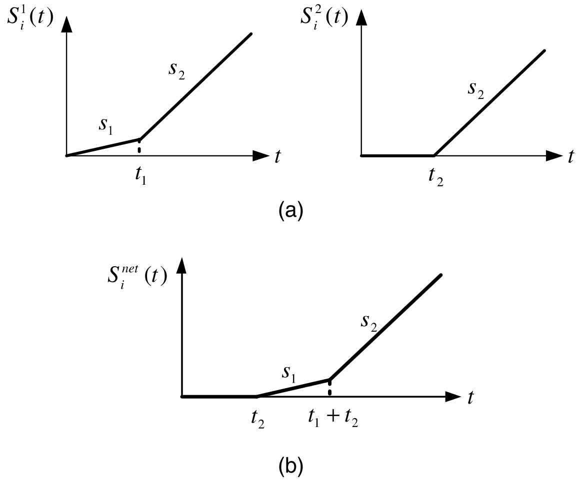

Fig. 3. Example—derivation of a network service curve given two routers: (a) service curves strictly guaranteed by each router and (b) derived network service curve.

Theorem 2 (Delay bound theorem). *Let $b_i(\cdot)$ be the traffic envelope function of session i. When a router guarantees $S_i(\cdot)$ to session i, the delay at the router is not greater than*

$$\max_{k:k>0} \min\{\Delta\colon \Delta > 0 \text{ and } b_i(k) \leqslant S_i(k+\Delta)\}. \tag{5}$$

Proof. See Appendix A. □

Theorem 3 (Backlog bound theorem). *Let $b_i(\cdot)$ the traffic envelope function of session i. If a router guarantees $S_i(\cdot)$ to session i, at any moment, the backlog is not greater than*

$$\max_{k:k>0} \{b_i(k) - S_i(k)\} + l^{\max}, \tag{6}$$

where $l^{\max}$ is the maximum packet size.

Proof. See Appendix A. □

Fig. 4 graphically illustrates the delay and backlog bound. In the figure, $b_i(t)$ is the maximum amount of traffic from session i for interval $(0, t]$ and $S_i(t)$, the minimum service amount. It can be easily understood that the maximum horizontal distance from $b_i(\cdot)$ to $S_i(\cdot)$, denoted by $D(b_i\|S_i)$, becomes the delay bound as described in Theorem 2. Similarly, the backlog bound described in Theorem 3 comes from the maximum vertical distance

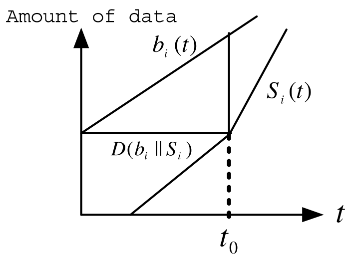

Fig. 4. The graphical interpretation of the delay and backlog bounds. The maximal horizontal distance denoted by $D(b_i\|S_i)$ becomes the delay bound. The vertical distance between $b_i(\cdot)$ and $S_i(\cdot)$ becomes the maximum at time t_0. The backlog bound is the maximal vertical distance plus the maximum packet size.

between $b_i(\cdot)$ and $S_i(\cdot)$. In the figure, the vertical distance becomes the maximum at time t_0. In the theorem, the backlog bound includes the term $l^{\max}$ since each packet is cleared in the backlog after the last bit of the packet is served.

3. Review of H-FSC

In this section, we review H-FSC proposed in [22]. H-FSC provides both real-time and hierarchical link-sharing services simultaneously. A real-time service is provided based on service curves described in the previous section. We first overview how H-FSC provides both services. Then, we present how each service is provided.

3.1. Overview of H-FSC

To provide real-time and hierarchical link-sharing services simultaneously, H-FSC maintains a single hierarchy in which each node represents a different entity such as a specific traffic type or data traffic from an organization. Nodes are constructed in advance to provide the hierarchical link-sharing service except those for real-time applications. In the case of real-time applications, one leaf node is constructed for each application when the application is admitted to provide the real-time service. Since one node is constructed for each real-time application, the service for the node is solely given to the application. The service for a node constructed in advance is shared by multiple non-real-time applications. The service amount for each node is determined by a service curve allocated to achieve either real-time or link-sharing service.

Fig. 5 illustrates an example hierarhcy. The hierarchy is constructed in advance except for the area within the box. The root node represents the link. In the example, the link capacity is 100 Mbps. Thus, the service curve for the root node, $S_{\text{Root}}(\cdot)$, becomes a straight line with a slope of 100 Mbps. In the next level, H-FSC allocates the service curves $S_{\text{RT}}(\cdot)$ and $S_{\text{nonRT}}(\cdot)$ with slopes of 30 and 70 Mbps to guarantee those bandwidths, respectively. For FTP, WWW, and other non-real-time applications, H-FSC guarantees 30, 30, and 10 Mbps, respectively, out of the 70 Mbps assigned for the non-real-time node. It allocates $S_{\text{FTP}}(\cdot)$, $S_{\text{WWW}}(\cdot)$, and $S_{\text{other}}(\cdot)$ as their service curves. The leaf nodes, video1 and video2, are constructed when each video application is admitted. The service curves, $S_{\text{video1}}(\cdot)$ and $S_{\text{video2}}(\cdot)$, are allocated to guarantee delay bounds of video1 and video2 real-time applications. (In Section 4.1, we describe how the service curves for such real-time applications are allocated.)

H-FSC provides real-time and hierarchical link-sharing services by integrating two separate algorithms sharing a single hierarchy. In the hierarchy, only leaf nodes can be used for real-time applications, whereas, for link-sharing services, both internal and leaf nodes are used. Since the real-time service and the link-sharing service have different goals, H-FSC maintains different data structures for leaf nodes and internal nodes. For a leaf node, H-FSC maintains three variables for each head packet: a deadline, an eligible time, and a virtual time. Deadlines are used to satisfy the real-time goal. A deadline is assigned in such a way that if the deadlines of all packets are met, delay bounds for all sessions are guaranteed. Virtual times are used to meet the hierarchical link-sharing goal. A virtual time indicates a normalized amount of service to the allocated service curve for the node. Eligible times are used to mediate both goals giving preference to the real-time goal. An eligible time is assigned in such a way that even though no service is provided to this node until the eligible time, there is no danger to violate a deadline. Thus, if all the eligible times are greater than the current time, there is no danger to violate *any* deadlines at all. For an internal node, H-FSC maintains only the virtual time since the node is not used for a real-time service. H-FSC transmits a packet with the earliest deadline if the smallest eligible time is not greater than the current time. Otherwise, a packet with the smallest virtual time is transmitted.

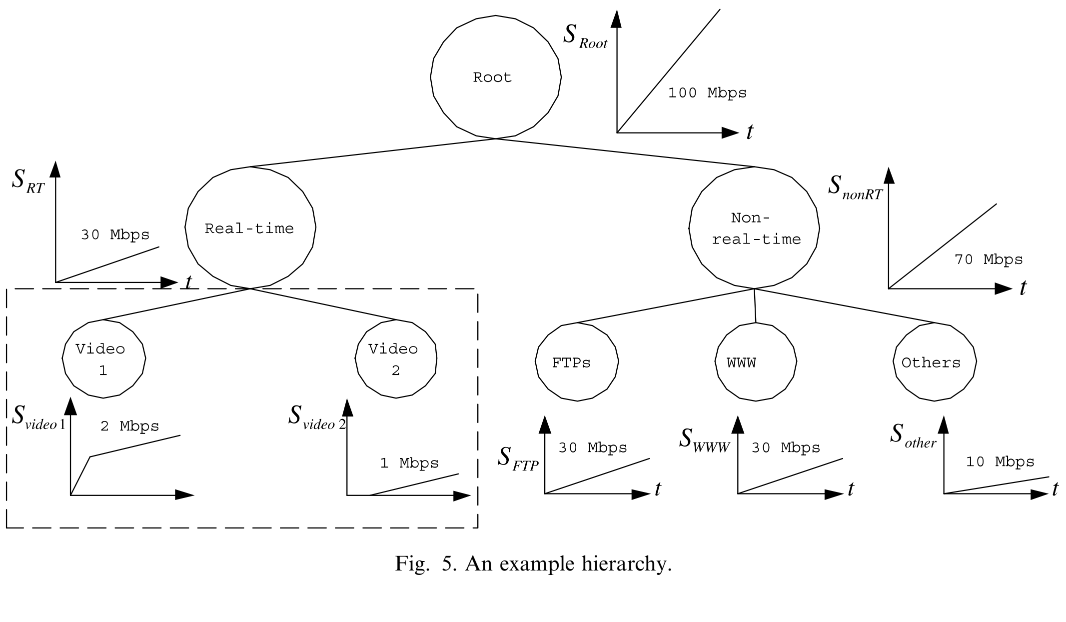

Fig. 5. An example hierarchy.

In the following subsections, we review the individual algorithms of H-FSC. We focus on explaining how to compute variables, namely, deadlines, eligible times, and virtual times. Throughout those subsections, we consider a leaf node i for a real-time session since the leaf node involves above mentioned three types of variables. We focus on the head packet of the node, denoted by p_i, which arrives during the mth backlogged period of the session, denoted by b_i^m.

3.2. Service curve algorithm for the real-time service

To satisfy the real-time service, H-FSC uses a service curve algorithm. The service curve algorithm transmits a packet with the earliest deadline non-pre-emptively. Ties are broken arbitrarily.

Let us see how the deadline of each packet is computed. For this, given a leaf node i, the algorithm keeps a deadline curve $D_i(\cdot)$, which is defined as

$$D_i(t) = \min_{s \in B_i(t)} \{W_i^{\mathrm{RT}}(s) + S_i(t-s)\}. \tag{7}$$

This $D_i(t)$ is derived from Eq. (3) and indicates the minimum amount of service to receive for the service curve $S_i(\cdot)$ to be strictly guaranteed. Then, given a head packet p_i and the accumulated amount of data including the packet p_i, denoted by w_i, the deadline of the packet p_i becomes the smallest t such that $D_i(t) = w_i$.

$D_i(\cdot)$ is initialized to $S_i(\cdot)$ when the session becomes backlogged for the first time. Subsequently, it is sufficient to update $D_i(\cdot)$ only when the session becomes newly backlogged. We denote the updated deadline curve at each backlogged point b_i^m by $D_i(b_i^m;\cdot)$. Then, $D_i(b_i^m;t)$ can be recursively computed as follows:

$$D_i(b_i^m;t) = \begin{cases} \min\{D_i(b_i^{m-1};t), W_i^{\mathrm{RT}}(b_i^m) + S_i(t-b_i^m)\} \\ \quad \text{for } t \geqslant b_i^m, \\ D_i(b_i^{m-1};t), \quad \text{otherwise.} \end{cases} \tag{8}$$

Note that $D_i(b_i^m;t) = D_i(t)$ if b_i^m is the last backlogged time point.

Through an admission test, the service curve algorithm accepts real-time applications. All the packets are transmitted until their deadlines plus $l^{\max}/C$ where $l^{\max}$ is the maximum packet size and C is the link capacity if the following equation is satisfied [22]:

$$\sum_{i=1}^{L} S_i(t) \leqslant Ct, \tag{9}$$

where $S_i(t)$ is the allocated service curve for a leaf node i and L is the number of leaf nodes. A complete and lengthy proof for this result can be found in [21]. In addition, to satisfy the hierarchical link-sharing, we allocate service curves in such a way that the following equation is satisfied:

$$\sum_j S_j(t) \leqslant S_p(t), \tag{10}$$

where $S_p(\cdot)$ and $S_j(\cdot)$ are the service curves for the node p and j, respectively, and the node j is a child of the node p. That is, the sum of the service curves of children does not exceed the parent's service curve. The service curve for the root $S_{\mathrm{root}}(t)$ is Ct. From Eqs. (9) and (10), we can derive a sufficient admission condition for real-time applications. Suppose that H-FSC uses an internal node k with a service curve $S_k(\cdot)$ for real-time applications. (A node for each real-time application is constructed as a child node of the node k if the application is admitted.) Consider each leaf node i for $i = 1, \ldots, N$ to be constructed for the respective real-time applications. Let leaf node i require a service curve $S_i(\cdot)$ be allocated to meet the delay bound for the application. (How to allocate such service curves is deferred to Section 4.1.) Then, from Eqs. (9) and (10), those applications are admitted if

$$\sum_{i=1}^{N} S_i(t) \leqslant S_k(t). \tag{11}$$

Note that since the admission condition in Eq. (11) is given, how to allocate a service curve determines the level of network utilization.

It is worth mentioning that the service curve algorithm does not guarantee the service curves for internal nodes although the short-term discrepancy between the serviced traffic amount for an internal node and the amount specified by the service curve is minimized. In fact, as mentioned in [22], it is impossible to guarantee all the service curves for both leaf and internal nodes simultaneously. Thus, the H-FSC guarantees the service curves for leaf nodes since those nodes are allocated to real-time sessions.

3.3. Hierarchical fair algorithm for the hierarchical link-sharing service

To achieve the hierarchical link-sharing service, H-FSC uses a hierarchical fair algorithm. The fair algorithm maintains a virtual time curve $V_i(\cdot)$ for each node i. $V_i(v)$ represents the amount of service for the node i to be received by the virtual time v. In the special case that the node i is an internal node, $V_i(\cdot)$ indicates the aggregated amount of service for all the descendants of the node i to be received. In addition, the parent of the node i, denoted by $p(i)$, maintains a system virtual time curve $V^s_{p(i)}(\cdot)$ for the node i and its siblings. $V^s_{p(i)}(v)$ represents the expected amount of service for any child to be received by the virtual time v. When a child of the node $p(i)$ becomes newly backlogged, $V^s_{p(i)}(\cdot)$ is used to increase the virtual times of the child to catch up with other siblings without penalizing them. In the H-FSC, $V^s_{p(i)}(\cdot)$ is set to $(V_{i,\max}(\cdot) + V_{i,\min}(\cdot))/2$ where $V_{i,\max}(\cdot)$ and $V_{i,\min}(\cdot)$ indicate the maximum and the minimum virtual time curve among the node i and its siblings, respectively, for minimizing the discrepancy between the virtual times of any two siblings. Specifically, similar to deadline curves, the virtual time curve $V_i(\cdot)$ for the node i is defined as follows [22]:

$$V_i(b_i^m; v) = \min\{V_i(b_i^{m-1}; v), W_i(b_i^m) + S_i(v - V^s_{p(i)})\} \quad \text{for } v \geqslant V^s_{p(i)}(b_i^m). \quad (12)$$

Note that $V_i(b_i^m; v) = V_i(v)$ if b_i^m is the last backlogged time point. Given a head packet, its virtual time is computed from the inverse of $V_i(\cdot)$. The hierarchical fair algorithm distributes bandwidth fairly to each node according to the hierarchy by transmitting a packet with the smallest virtual time. (Sometimes a packet with no smallest virtual time may be transmitted by the service curve algorithm. However, this deviation disappears in long intervals if the service curve algorithm transmits a packet *only* when there is a danger to violate deadlines.)

3.4. Mediating real-time and link-sharing services

Since it is not possible to guarantee both real-time and link-sharing services all the time [22], the H-FSC gives "enough" bandwidth for the real-time service not to violate deadlines in the future. The excess bandwidth can be safely distributed for the link-sharing service. However, this preference to the real-time service may result in deviation from the ideal link-sharing service whenever a packet with no smallest virtual time is transmitted by the service curve algorithm. Thus, to minimize the deviation in just a short period, the H-FSC tries to transmit a packet by the service curve algorithm *only* when there is a danger to violate deadlines in the future.

Let us see how the H-FSC can determine whether a packet can be safely transmitted for the link-sharing service without the danger of violating deadlines. Consider the current time t. Let us denote by $\mathscr{A}(t)$ and $\mathscr{P}(t)$ the set of backlogged and non-backlogged sessions at the time t, respectively. In the worst-case, for a future time t', all the sessions in $\mathscr{A}(t)$ remain backlogged up to the time t'. In addition, all the sessions in $\mathscr{P}(t)$ also become immediately backlogged at the time t and remain backlogged during the time interval $(t, t']$. In this case, the maximum amount of service over the interval $(t, t']$ required by $\mathscr{A}(t)$ and $\mathscr{P}(t)$ becomes $\sum_{i\in\mathscr{A}(t)}(D_i(b_i^m; t') - D_i(b_i^m; t))$ and $\sum_{i\in\mathscr{P}(t)}(D_i(t; t') - W_i^{\mathrm{RT}}(t))$, respectively. Until the time t, $\mathscr{A}(t)$ has received at least $\sum_{i\in\mathscr{A}(t)} D_i(b_i^m; t)$ to guarantee their service curves. Similarly, both $\mathscr{A}(t)$ and $\mathscr{P}(t)$ can together receive at most $C \times (t' - t)$ during the interval $(t, t']$. Now let us denote the minimum service amount required by the service curve algorithm until the time t to guarantee all the service curves for leaf nodes in the future by the system eligible curve $E(t)$. Then, $E(t)$ is defined as follows [22]:

$$E(t) = \sum_{i\in\mathscr{A}(t)} D_i(b_i^m; t) + \left[\max_{t'>t}\left(\sum_{i\in\mathscr{A}(t)}(D_i(b_i^m; t') - D_i(b_i^m; t)) + \sum_{i\in\mathscr{P}(t)}(D_i(t; t') - W_i^{\mathrm{RT}}(t)) - C \times (t' - t)\right)\right]^+, \quad (13)$$

where b_i^m is the last backlogged time point of the session i and $[x]^+$ is defined as $\max(x, 0)$. If the total service amount transmitted by the service curve algorithm is no less than $E(t)$ at the time t, there is no danger to violate any deadline at all.

However, maintaining $E(t)$ is not practical since it depends on *all* sessions. Whenever a session becomes either backlogged or empty, $E(t)$ is updated using all the deadline curves. Thus, each updating requires $O(N)$ complexity where N is the number of sessions. In addition, each updaetd $E(t)$ can be arbitrarily many piecewise linear even if all deadline curves are two piecewise linear. Thus, it is difficult to maintain and implement $E(t)$ efficiently.

Due to the implementation difficulty in $E(\cdot)$, the H-FSC keeps an *eligible time curve* $E_i(\cdot)$ for each leaf node i instead of $E(\cdot)$. $E_i(t)$ represents the minimum amount of service for leaf node i to receive until time t to prevent any deadline violations. If the node receives less service until t, there exists potential danger to violate the deadlines of some packets in the node. On the contrary, it is superfluous to provide more service than $E_i(t)$ until t not to violate the deadlines. In [22], it is shown that $\sum_{i\in\mathscr{A}(t)} E_i(t) \geqslant E(t)$. That is, $\sum_i E_i(\cdot)$ overestimates $E(\cdot)$ slightly. However, $E_i(\cdot)$ has a big advantage on maintenance and implementation. $E_i(\cdot)$ does not depend on other sessions at all. As with the deadline curve $D_i(t)$, $E_i(\cdot)$ is updated whenever the session is newly backlogged. For this update, H-FSC maintains $E_i(b_i^m;\cdot)$ at the latest backlogged time point b_i^m as follows [22]:

$$E_i(b_i^m;t) = D_i(b_i^m;t) + \left[\max_{t'>t}\left(D_i(b_i^m;t') - D_i(b_i^m;t) - S_i(t'-t)\right)\right]^+ \quad \text{for } t \geqslant b_i^m. \tag{14}$$

Note that $E_i(\cdot) = E_i(b_i^m;\cdot)$ if b_i^m is the last backlogged time point.

Given a head packet p_i, its eligible time is computed from the inverse of $E_i(\cdot)$. Then, if all the eligible times are greater than the current time, a packet can be safely served for the link-sharing service. Thus, if the smallest eligible time is not greater than the current time, the service curve algorithm is selected. Otherwise, the hierarchical fair algorithm is selected.

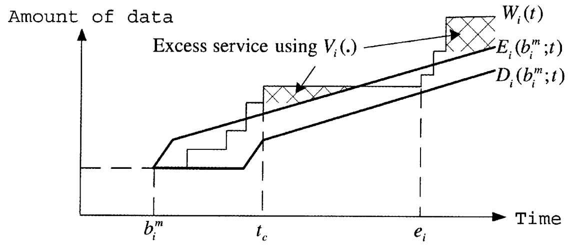

Fig. 6. An illustrative example of the deadline curve $D_i(\cdot)$, the eligible time curve $E_i(\cdot)$, and the work amount curve $W_i(\cdot)$.

3.5. An illustration of the deadline, the eligible time and the virtual time curve

The link bandwidth is commonly used to provide the real-time and the link-sharing service with different goals. In the H-FSC, both requirements are specified by service curves. That is, a session's real-time and bandwidth requirement is expressed by a single service curve, not a separate curve for either requirement. However, each curve uses the common service curve in different ways to satisfy its goal. Fig. 6 illustrates an example of the deadline curve $D_i(\cdot)$, the eligible time curve $E_i(\cdot)$, and the work amount curve $W_i(\cdot)$ of the session i when the last backlogged time point is b_i^m. In the figure, when the current time is t_c, the head packet of the session i is transmitted by the fair algorithm using the virtual time curve $V_i(\cdot)$ since its eligible time e_i is greater than t_c. The virtual time curve $V_i(\cdot)$ is not illustrated explicitly since the function is based on a different time domain, i.e., the virtual time. Although a virtual time curve is initially set by a curve based on the real time, both times are not synchronized. For example, if a session receives more service than its specified service curve, its virtual time goes faster than the real time. Note that for the fair service, only the relative time values of sessions are meaningful, not their absolute values.

4. Proposed service curve allocation scheme

In Section 4.1, we propose a scheme to allocate a service curve given a traffic characterization and a delay bound for each session. The service curve algorithm reviewed in the previous section can use any non-decreasing functions. The service curve allocated by our scheme is a specialization of a service curve which results in tight bandwidth reservation and achieves a high level of network utilization.

4.1. Service curve allocation

Given a guaranteed service curve for a real-time session, we can induce the delay bound for the session by Theorem 2. We use this theorem as the basis of our service curve allocation scheme.

When a service curve is allocated for a session, the curve cannot be guaranteed by the H-FSC. This is so since packets are transmitted non-pre-emptively. The following theorem tells that to guarantee a service curve $S_i(\cdot)$, the H-FSC needs to allocate $S_i(\cdot)$ shifted *left* by $l^{\max}/C$ to the session where $l^{\max}$ is the maximum packet size and C is the link capacity. (The following theorem was also touched in [22], but very lightly without a formal proof.)

Theorem 4. *Suppose that each leaf node i has been allocated a service curve $S_i(\cdot)$ for a real-time session and has passed the admission test. Then, H-FSC strictly guarantees the following service curve $\widehat{S}_i(\cdot)$ for the session*:

$$\widehat{S}_i(t) = \begin{cases} 0, & t \leqslant l^{\max}/C, \\ S_i(t - l^{\max}/C), & t > l^{\max}/C, \end{cases} \tag{15}$$

where $l^{\max}$ is the maximum packet size and C is the link capacity.

Proof. See Appendix A. □

We first derive an ideal service curve allocation scheme using Theorems 2 and 4. Consider a session i which requires a $(d_i + l^{\max}/C)$ delay bound to be guaranteed, where C is the link capacity and $l^{\max}$ is the maximum packet size. Let the traffic envelope function $b_i(t)$ for session i be $\min_{k=1\cdots K}\{\sigma_i^k + \rho_i^k t\}$ where σ_i^k and ρ_i^k are greater than 0. In this case, we can see that H-FSC can guarantee the delay bound for session i if the following service curve $T_i(\cdot)$ is allocated to session i by the theorems:

$$T_i(t) = \begin{cases} 0, & 0 \leqslant t \leqslant d_i, \\ \min_{k=1\cdots K}\{\sigma_i^k + \rho_i^k(t - d_i)\}, & t > d_i. \end{cases} \tag{16}$$

However, if $T_i(\cdot)$ is allocated as a service curve, deadline computation becomes difficult. As an example, Fig. 7 illustrates an initial deadline curve for session i in the case that $T_i(\cdot)$ is allocated as the service curve for session i. In the figure, $D_i(\cdot)$ is the deadline curve initialized to $T_i(\cdot)$ when session i is backlogged at the first time. Consider a packet with size l which becomes head at a time t_c during the first backlogged period. The deadline of the packet is the first time point at which the accumulated input amount $(W_i^{\mathrm{RT}}(t_c) + l)$ becomes equal to $D(\cdot)$. In the context of mathematics, such a time point exists. However, we cannot express the *exact* value due to the discontinuity of $D_i(\cdot)$ at the time d_i. Thus, deadline computation using service curve in Eq. (16) is difficult.

To overcome the difficulty in deadline computation, we allocate a *concave* service curve [2] based on $T_i(\cdot)$. Fig. 8 graphically illustrates how to construct a service curve $S_i(\cdot)$ from $T_i(\cdot)$. In the figure, for $t > d_i$, $S_i(t)$ is identical to $T_i(t)$. However, to make $S_i(\cdot)$ concave, we allocate the first non-zero slope in $S_i(\cdot)$ to $\rho^{\max}$ which is the maximum rate we set up manually such that $\rho^{\max} \geqslant \rho_i^k$ for all i and k. Due to $\rho^{\max}$, the first non-zero line segment in $S_i(\cdot)$ starts from $(d_i - \sigma_i^1/\rho^{\max}, 0)$ and ends at (d_i, σ_i^1). The service curve $S_i(\cdot)$ constructed in the figure can be represented by an equation. For a simple representation, let σ_i^0 and ρ_i^0 be equal to σ_i^1 and $\rho^{\max}$, respectively. In addition, let us denote $(d_i - \sigma_i^1/\rho^{\max})$ by x_i. Then, the service curve $S_i(\cdot)$ becomes as follows:

$$S_i(t) = \begin{cases} 0, & 0 \leqslant t \leqslant x_i, \\ \min_{k=0\cdots K}\{\sigma_i^k + \rho_i^k(t - d_i)\}, & t > x_i. \end{cases} \tag{17}$$

Note that service curve in Eq. (17) is continuous. This continuity results in deadline curve without discontinuity. Thus, deadline computation be-

[2] A service curve $S_i(t)$ is said to be concave if for any two times, t_1 and t_2, and for any $\alpha \in (0, 1)$, we have $S_i(\alpha t_1 + (1-\alpha)t_2) \geqslant \alpha S_i(t_1) + (1-\alpha)S_i(t_2)$.

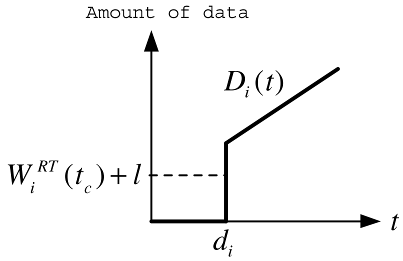

Fig. 7. An initial deadline curve for session i. Given a packet with size l which becomes head at a time t_c, it is difficult to determine the deadline of the packet since the deadline curve jumps at the time d_i.

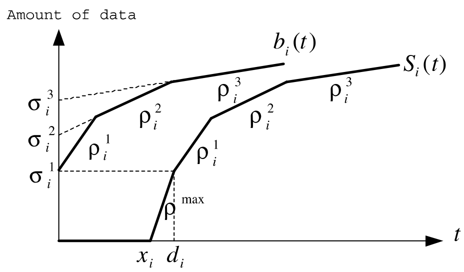

Fig. 8. The allocated service curve $S_i(\cdot)$ for session i in our scheme when session i is expected to have the traffic envelope function $b_i(\cdot)$ ($x_i = d_i - \sigma_i^1/\rho^{\max}$).

comes easier than that using service curve in Eq. (16).

Let us compare the service curve allocation scheme proposed in [22] with our scheme. While our scheme considers general $K(\sigma,\rho)$-*smooth* sessions, the scheme proposed in [22] considers only (σ,ρ)-*smooth* sessions. Fig. 9 graphically illustrates how to allocate a service curve in that scheme. In the figure, the allocation scheme is divided in two cases. Note that in both cases, a two piecewise linear service curve is allocated. On the contrary, if our scheme is used, a three piecewise linear curve will be allocated for a (σ,ρ)-*smooth* session. We will see in the next section that our service curve is tighter due to the first non-zero slope $\rho^{\max}$.

4.2. Network utilization

In [15], it is shown that if the ideal service curve allocation scheme explained in the previous section is used, H-FSC can achieve the same network utilization as the EDF with regulators. The EDF with regulators can achieve a very high level of network utilization [8,7,25].

The service curves allocated by our scheme are very close to those by the ideal scheme. While the ideal scheme allocates $T_i(\cdot)$ for session i, our scheme allocates $S_i(\cdot)$ for the session. Note that if $\rho^{\max} \to \infty$, $S_i(\cdot) \to T_i(\cdot)$. Thus, if we set up $\rho^{\max}$ to a sufficiently large value, our scheme can achieve the same level of network utilization as the EDF with regulators. Thus, our scheme can achieve a very high level of network utilization.

Our scheme can achieve higher level of network utilization than the scheme proposed in [22] for VBR sessions. This is because the service curves allocated by the scheme in [22] is not tight for VBR sessions compared to those by our scheme. Fig. 10 illustrates two different service curves allocated by both schemes to meet a delay bound for a VBR session. The delay bound is $(d_i + l^{\max}/C)$ where C

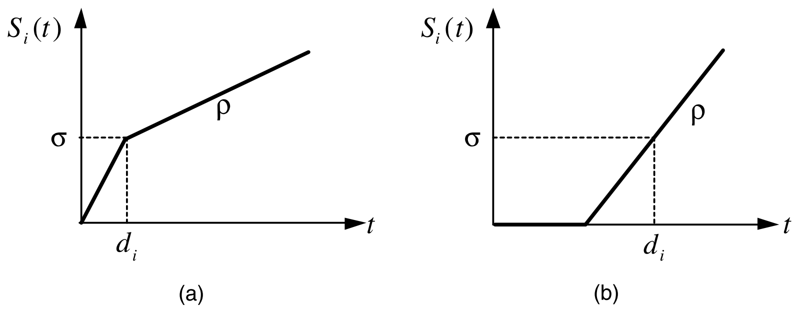

Fig. 9. The service curve allocation proposed in [22] for (σ,ρ)-*smooth* session i with $(d_i + l^{\max}/C)$ delay bound: (a) allocated service curve when $\sigma/d_i > \rho$ and (b) allocated service curve in other cases.

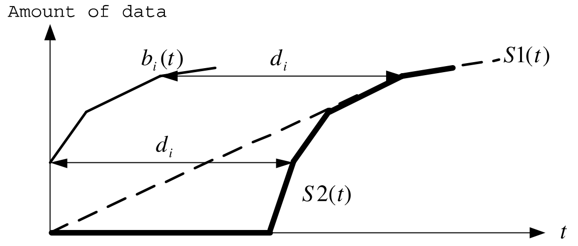

Fig. 10. Different service curves to guarantee the same delay bound for a VBR session. $S1(\cdot)$ is a two piecewise linear curve. $S2(\cdot)$ is a five piecewise linear curve. Note that $S2(\cdot)$ is much tighter.

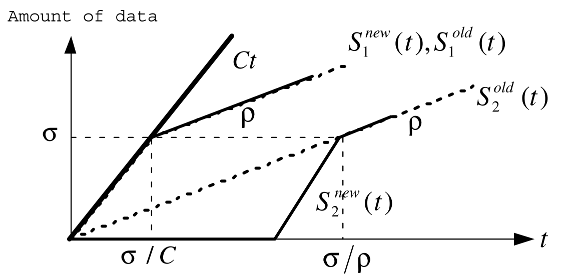

Fig. 11. Service curves for CBR sessions allocated by our scheme and the scheme proposed in [22].

is the link capacity and l^{max} is the maximum packet size. In the figure, $b_i(\cdot)$ is the expected traffic envelope function for the VBR session. The service curves, $S1(\cdot)$ and $S2(\cdot)$, are piecewise linear curves which have two and five line segments, and are allocated by the scheme in [22] and our scheme, respectively. Since $D(b_i\|S1)$ and $D(b_i\|S2)$ are all equal to d_i, the two service curves can guarantee the same delay bound d_i. However, the service curve $S1(\cdot)$ is less tight than $S2(\cdot)$.

Even for constant-bit-rate (CBR) sessions, our scheme can achieve a higher level of network utilization. We show this network utilization advantage of our scheme by an example. Consider two sessions, sessions 1 and 2. Let both sessions be expected to have the same traffic envelope function (σ, ρ), but with different delay bounds, $(\sigma/C + l^{max}/C)$ and $(\sigma/\rho + l^{max}/C)$, respectively, where l^{max} is the maximum packet size. Let us denote the service curves for sessions 1 and 2 allocated by our scheme by $S_1^{new}(\cdot)$ and $S_2^{new}(\cdot)$, respectively. We denote the curves for sessions 1 and 2 allocated by the scheme proposed in [22] by $S_1^{old}(\cdot)$ and $S_2^{old}(\cdot)$, respectively. Fig. 11 illustrates those service curves in the case that we set up ρ^{max} to C. In the figure, in the case of session 1, both schemes allocate the same service curve. However, for session 2, different service curves are allocated. Note that both sessions are admitted if H-FSC adopts our scheme. However, if H-FSC adopts the scheme proposed in [22], either sessions 1 or 2 is rejected. (If we add the service curves $S_1^{old}(\cdot)$ and $S_2^{old}(\cdot)$ in the figure, the added curve exceeds the link capacity curve during the interval $[0, \sigma/C]$.)

5. Scheduling complexity

The service curves allocated by the proposed scheme are piecewise linear. If piecewise curves are allocated without any restriction, computation of time-stamps for incoming packets, i.e., deadlines, eligible times and virtual times, requires high complexity. In [22], a special type of service curves, i.e., only two piecewise linear curves, are allocated to avoid such a high complexity. However, we show that the piecewise linear service curves allocated by our scheme require just a constant time for computing the time-stamps. That is, compared to the service curve allocation scheme proposed in [22], our scheme has better network utilization while having the same scheduling complexity.

The scheduling complexity of H-FSC consists of two parts, computing time-stamps for incoming packets and transmitting those packets. For packet transmission, three sorted priority queues are maintained, one for deadlines, another for virtual times, and the third for eligible times. The maintenance cost for a sorted priority queue is $O(\log N)$ where N is the number of leaf nodes. This cost can be further reduced to $O(1)$ with a special hardware such as a sequencer [10] or a systolic array [11]. Note that whatever service curves are used, this cost for packet transmission remains unchanged.

However, computing the time-stamps has different complexities depending on allocated service curves. To compute the time-stamps, each leaf node maintains three curves, i.e., a deadline curve, an eligible curve, and a virtual time curve, which depend on the allocated service curve. The deadline, the eligible time, and the virtual time of a head packet are calculated from the inverse of those curves.

In the subsequent subsections, we show that each time stamp can be computed in a constant time if a service curve is allocated by our scheme. In Section 5.1, we show that the maintenance cost for a deadline curve is constant. We also show that the deadline of a head packet can be calculated from the deadline curve in a constant time. Similarly, in Section 5.2, we show that the cost to maintain an eligible time curve and that to calculate an eligible time are constants. The same results apply to the maintenance of a virtual time curve and the calculation of a virtual time. However, we do not present the case of the virtual time since a virtual time curve is very similar to a deadline curve [22]. Detailed steps can be easily inferred from the case of a deadline curve.

5.1. The costs related to the deadline curve and the deadline

As mentioned in Section 3.2, given an allocated service curve $S_i(\cdot)$ and the last backlogged time point b_i^m, the deadline curve $D_i(b_i^m;\cdot)$ is updated as follows:

$$D_i(b_i^m;t) = \min\{D_i(b_i^{m-1};t), W_i^{\mathrm{RT}}(b_i^m) + S_i(t-b_i^m)\} \quad \text{for } t \geqslant b_i^m. \tag{18}$$

From Eq. (18), we can see that if the allocated service curve is complex, the update cost for the deadline curve becomes high. We show by an example that even with a simple service curve, which is two piecewise linear, updating the deadline curve incurs high complexity. However, we show that, at each update step, we can reduce the maintenance range of the deadline curve. Maintaining the deadline curve on this reduced range is sufficient to compute the subsequent deadlines. Further, the complexity of the computation is significantly reduced; the update cost for the deadline curve becomes constant since unnecessary time points are trimmed out at each update step. In addition, the deadline calculation can also be done in a constant time.

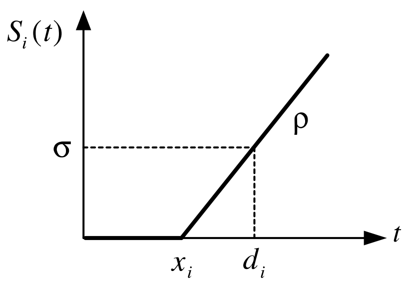

Fig. 12. The two piecewise service curve $S_i(\cdot)$ allocated by our scheme in the case that $\rho^{\max}$ is equal to ρ for simplicity of discussion.

Consider a leaf node i which is constructed for a real-time session. Suppose that the real-time session requires a delay bound of $(d_i + l^{\max}/C)$ to be guaranteed, where C is the link capacity and $l^{\max}$ is the maximum packet size. Assume that a two piecewise linear service curve $S_i(\cdot)$ has been allocated as shown in Fig. 12. This two piecewise curve has been allocated by setting up $\rho^{\max}$ to ρ for simplicity of discussion. Then, from Eq. (18), when $D_i(\cdot)$ is secondly updated, $D_i(\cdot)$ becomes as follows:

$$D_i(b_i^2;t) = \min\{S_i(t-b_i^1), S_i(t-b_i^2) + W_i^{\mathrm{RT}}(b_i^2)\} \quad \text{for } t \geqslant b_i^2. \tag{19}$$

Fig. 13 illustrates an example of $D_i(b_i^2;\cdot)$ given the two piecewise service curve $S_i(\cdot)$. In figure (a), the two curves used to update $D_i(\cdot)$, i.e., $S_i(t-b_i^1)$ and $S_i(t-b_i^2) + W_i^{\mathrm{RT}}(b_i^2)$, are illustrated. In this case, the two curves cross each other. In figure (b), the resulting deadline curve $D_i(\cdot)$ updated using the curves in figure (a) is illustrated. Note that, in the figure (b), the updated $D_i(t)$ is *not* linear for $t \geqslant b_i^2$. Thus, at the mth backlogged time point b_i^m, updating $D_i(b_i^m;t)$ for $t \geqslant b_i^m$ may require a very high complexity.

Fortunately, the following lemma tells that, at each mth backlogged time point b_i^m, to compute subsequent deadlines, it is sufficient to maintain $D_i(b_i^m;t)$ only for $t \geqslant b_i^m + x_i$. Here, x_i is the first time point at which the allocated service curve $S_i(\cdot)$ has a non-zero value. (An example of x_i is represented in Fig. 12.)

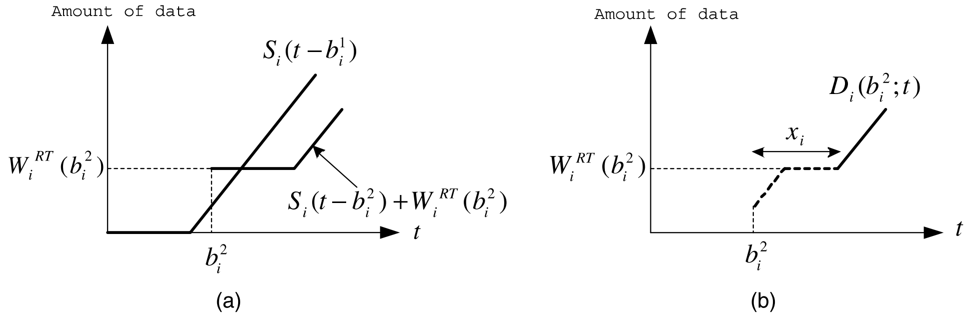

Fig. 13. An example of updating $D_i(\cdot)$ when the session is secondly backlogged given the two piecewise service curve $S_i(\cdot)$: (a) two curves for updating $D_i(\cdot)$ and (b) updated $D_i(\cdot)$.

Lemma 1. *Suppose that a service curve $S_i(\cdot)$ is allocated to leaf node i for a real-time session. Let x_i be the first time point at which $S_i(\cdot)$ has a non-zero value. Consider the packets from the session which arrive after the last backlogged time point b_i^m. Then, the deadlines of those packets are always greater than $b_i^m + x_i$.*

Proof. Consider a packet p which becomes the head packet after the last backlogged time point b_i^m. Let us denote the length and the deadline of the packet p by l and d, respectively. We also denote the time the packet p becomes head by t_c. In this case, the accumulated amount of data including the packet p to be transmitted until the deadline d becomes $(W_i^{\mathrm{RT}}(t_c) + l)$, which we denote by w for notational convenience. Then,

$$\begin{aligned} d &= \min\{t\colon D_i(b_i^m;t) = w\} && (20)\\ &= \min\{t\colon \min\{D_i(b_i^{m-1};t),\\ &\qquad W_i^{\mathrm{RT}}(b_i^m) + S_i(t-b_i^m)\} = w\} && (21)\\ &= \max\{\min\{t\colon D_i(b_i^{m-1};t) = w\},\\ &\qquad \min\{t\colon W_i^{\mathrm{RT}}(b_i^m) + S_i(t-b_i^m) = w\}\} && (22)\\ &\geqslant \min\{t\colon W_i^{\mathrm{RT}}(b_i^m) + S_i(t-b_i^m) = w\} && (23)\\ &\geqslant \min\{t\colon S_i(t-b_i^m) = l\}\\ &\qquad (\text{since } W_i^{\mathrm{RT}}(t_c) \geqslant W_i^{\mathrm{RT}}(b_i^m)) && (24)\\ &\geqslant b_i^m + x_i. \qquad \square && (25) \end{aligned}$$

For the reduced maintenance range, the deadline curve in Fig. 13(b) can be maintained in a constant time. In the figure, the updated $D_i(\cdot)$ remains linear after the time $(b_i^2 + x_i)$. To maintain a linear curve, it is sufficient to keep the starting time point and the slope.

Now we formally show in the following lemma that if we allocate a service curve using our scheme, at each backlogged time point, the updated deadline curve is reduced to a simple curve which is *K piecewise linear* for the reduced maintenance range.

Lemma 2. *Consider a leaf node i which is constructed for a real-time session. Suppose that the real-time session is expected to have the traffic envelope function $\min_{k=1\cdots K}\{\sigma_i^k + \rho_i^k t\}$ and requires the delay bound of $(d_i + l^{\max}/C)$ to be guaranteed, where C is the link capacity and $l^{\max}$ is the maximum packet size. Further assume that a service curve to the leaf node i is allocated using Eq.* (17). *Then, when the session becomes backlogged at the mth time, the updated $D_i(b_i^m;\cdot)$ becomes a K piecewise linear curve as follows*:

$$D_i(b_i^m;t) = \min_{k=0\cdots K}\{C_i^{m,k} + \rho_i^k t\} \quad \textit{for } t \geqslant b_i^m + x_i, \tag{26}$$

where $\rho_i^0 = \rho^{\max}$, $\sigma_i^0 = \sigma_i^1$, $x_i = d_i - \sigma_i^1/\rho^{\max}$, and

$$C_i^{m,k} = \begin{cases} \sigma_i^k - \rho_i^k(b_i^1 + d_i), & m = 1,\\ \min\{C_i^{m-1,k}, \sigma_i^k - \rho_i^k(b_i^m + d_i) + W_i^{\mathrm{RT}}(b_i^m)\},\\ \quad m \geqslant 2. \end{cases} \tag{27}$$

Proof. Using Eq. (17), the leaf node i is allocated the following service curve $S_i(\cdot)$ for the session:

$$S_i(t) = \begin{cases} 0, & 0 \leqslant t \leqslant x_i, \\ \min_{k=0\cdots K} \{\sigma_i^k + \rho_i^k (t - d_i)\}, & t > x_i. \end{cases} \quad (28)$$

Given the service curve $S_i(\cdot)$, the deadline curve $D_i(\cdot)$ is updated at each backlogged time point b_i^m as follows:

$$D_i(b_i^m; t) = \min\{D_i(b_i^{m-1}; t), W_i^{\mathrm{RT}}(b_i^m) + S_i(t - b_i^m)\} \quad \text{for } t \geqslant b_i^m + x_i. \quad (29)$$

Note that in Eq. (29), by Lemma 1, $D_i(b_i^m; t)$ is maintained only for $t \geqslant b_i^m + x_i$, not for $t \geqslant b_i^m$. Then, we prove the lemma by structural induction on the number of backlogged times. As the base step, for $t \geqslant b_i^1 + x_i$,

$$D_i(b_i^1; t) = S_i(t - b_i^1) = \min_{k=0\cdots K} \{\sigma_i^k + \rho_i^k (t - b_i^1 - d_i)\} = \min_{k=0\cdots K} \{C_i^{1,k} + \rho_i^k t\}.$$

By the induction hypothesis, for $t \geqslant b_i^m + x_i$,

$$D_i(b_i^m; t) = \min_{k=0\cdots K} \{C_i^{m,k} + \rho_i^k t\}. \quad (30)$$

Then, as the induction step, for $t \geqslant b_i^{m+1} + x_i$,

$$\begin{aligned} D_i(b_i^{m+1}; t) &= \min\{D_i(b_i^m; t), S_i(t - b_i^{m+1}) + W_i^{\mathrm{RT}}(b_i^{m+1})\} \\ &= \min\{\min_{k=0\cdots K} \{C_i^{m,k} + \rho_i^k t\}, \\ &\quad \min_{k=0\cdots K} \{\sigma_i^k + \rho_i^k (t - b_i^{m+1} - d_i)\} \\ &\quad + W_i^{\mathrm{RT}}(b_i^{m+1})\} \\ &= \min_{k=0\cdots K} \{\min\{C_i^{m,k}, \sigma_i^k - \rho_i^k (b_i^{m+1} + d_i) \\ &\quad + W_i^{\mathrm{RT}}(b_i^{m+1})\} + \rho_i^k t\} \\ &= \min_{k=0\cdots K} \{C_i^{m+1,k} + \rho_i^k t\}. \qquad \square \end{aligned}$$

Finally, using Lemma 2, we show in the following theorem that, at each update step, the cost to update a deadline curve as well as that to calculate deadlines become all constant.

Theorem 5. *Given a service curve allocated by Eq. (17) for a real-time session, at each backlogged time point, the deadline curve can be updated in a constant time. In addition, subsequent deadline calculations from the inverse of the deadline curve can also be done in a constant time.*

Proof. First, we prove that updating each deadline curve can be done in a constant time. It is sufficient to show that the deadline curve $D_i(b_i^m; \cdot)$ presented in Lemma 2 can be updated in a constant time. By Lemma 2, at the mth backlogged time point b_i^m, $D_i(\cdot)$ is updated as follows:

$$D_i(b_i^m; t) = \min_{k=0\cdots K} \{C_i^{m,k} + \rho_i^k t\} \quad \text{for } t \geqslant b_i^m + x_i. \quad (31)$$

From Eq. (31), to update $D_i(b_i^m; \cdot)$, we need to calculate $C_i^{m,k}$ for $k = 0, \ldots, K$. If we save $C_i^{m-1,k}$ for $k = 0, \ldots, K$ at the previous update, $\min\{C_i^{m-1,k}, \sigma_i^k - \rho_i^k (b_i^m + d_i) + W_i^{\mathrm{RT}}(b_i^m)\}$ can be done in O(1) since both $\sigma_i^k - \rho_i^k (b_i^m + d_i)$ and $W_i^{\mathrm{RT}}(b_i^m)$ are fixed at the time b_i^m. Thus, each $C_i^{m,k}$ can be calculated in O(1). Since each $\{C_i^{m,k} + \rho_i^k t\}$ is a linear curve, $D_i(b_i^m; \cdot)$ becomes a concave piecewise linear for the range. For maintaining this concave curve, we need to keep $K + 1$ number of starting time points and slopes. This operation can be done in $O(K)$. In most cases, K is restricted to a small number [16]. Thus, the updating cost is constant.

Next, we show that deadline calculations can be done in a constant time. The deadline of each packet is calculated from the inverse of $D_i(b_i^m; \cdot)$ if b_i^m is the last backlogged time point. Recall that $D_i(b_i^m; \cdot)$ is a *concave* piecewise linear for the range. Thus, the deadline of each packet can be calculated from the inverse of the deadline curve in $O(K)$ where K is the number of line pieces of the deadline curve. As mentioned, usually K is a small number. So, the deadline calculation cost is constant. $\square$

5.2. The costs related to the eligible time curve and the eligible time

In this subsection, we show that given a service curve allocated by our scheme, the maintenance cost for the eligible time curve is constant, and the eligible time of the head packet can be calculated in a constant time.

As mentioned in Section 3.4, if a service curve $S_i(\cdot)$ is allocated, at the last backlogged time point b_i^m, the eligible curve $E_i(\cdot)$ is updated as follows:

$$E_i(b_i^m;t) = D_i(b_i^m;t) + [\max_{t'>t} (D_i(b_i^m;t') - D_i(b_i^m;t) - S_i(t'-t))]^+ \quad \text{for } t \geqslant b_i^m. \tag{32}$$

$E_i(\cdot)$ in Eq. (32) is complicated. Moreover, considering the complexity of the curve $D_i(b_i^m;\cdot)$ in Eq. (18), the update cost for the eligible time curve can be extremely high. However, we show that given a service curve allocated by our scheme, the eligible time curve can be reduced to a simple one which is concave piecewise linear. In reality, the update of the eligible curve can be done using the updated deadline curve $D_i(b_i^m;\cdot)$ by a simple shifting. Thus, the complexity of maintaining the eligible curve becomes that of a deadline curve. By the Lemmas 3–5, we first show that the eligible curve can be reduced to a simple curve, which is a shifted deadline curve. Then, in Theorem 6, we show that the updating cost for the simple curve and the cost for calculating the eligible time become constants.

Lemma 3. *Consider a concave piecewise linear function $f(\cdot)$. Let t_1 and t_2 be x-coordinates such that $t_1 \leqslant t_2$. Then, given an interval of length I,*

$$f(t_2+I) - f(t_2) \leqslant f(t_1+I) - f(t_1). \tag{33}$$

Proof. Trivial. □

Lemma 4. *Assume that a service curve $S_i(\cdot)$ is allocated using Eq. (17) for a $K(\sigma,\rho)$ session. Let x_i be the first time point at which the service curve $S_i(\cdot)$ has a non-zero value. Then, given the service curve $S_i(\cdot)$, at the mth backlogged time point b_i^m, the updated deadline curve satisfy the following inequality:*

$$D_i(b_i^m;t+\Delta) - D_i(b_i^m;t) \leqslant S_i(x_i+\Delta), \tag{34}$$

where $t \geqslant b_i^m + x_i$ and $\Delta > 0$.

Proof. Consider a $K(\sigma,\rho)$ session which requires a delay bound of $(d_i + l^{\max}/C)$ to be guaranteed, where C is the link capacity and $l^{\max}$ is the maximum packet size. Using Eq. (17), for the $K(\sigma,\rho)$ session, the following service curve $S_i(\cdot)$ is allocated:

$$S_i(t) = \begin{cases} 0, & 0 \leqslant t \leqslant x_i, \\ \min_{k=0\cdots K}\{\sigma_i^k + \rho_i^k(t-d_i)\}, & t > x_i, \end{cases} \tag{35}$$

where $\sigma_i^0 = \sigma_i^1$, $\rho_i^0 = \rho^{\max}$, and $x_i = d_i - \sigma_i^1/\rho^{\max}$. From Eq. (35),

$$S_i(x_i+\Delta) = \min_{k=0\cdots K}\left\{\sigma_i^k + \rho_i^k\left(-\frac{\sigma_i^1}{\rho^{\max}} + \Delta\right)\right\}. \tag{36}$$

Given the service curve $S_i(\cdot)$, at the mth backlogged time point, the deadline curve is updated, as shown in Lemma 2, as follows:

$$D_i(b_i^m;t) = \min_{k=0\cdots K}\{C_i^{m,k} + \rho_i^k t\} \quad \text{for } t \geqslant b_i^m + x_i, \tag{37}$$

where

$$C_i^{m,k} = \begin{cases} \sigma_i^k - \rho_i^k(b_i^1 + d_i), & m = 1, \\ \min\{C_i^{m-1,k}, \sigma_i^k - \rho_i^k(b_i^m + d_i) + W_i^{\mathrm{RT}}(b_i^m)\}, \\ \quad m \geqslant 2. \end{cases} \tag{38}$$

Now we prove the lemma by structural induction on the number of backlogged times. As the base step, consider the first backlogged time point b_i^1. In this case, $t \geqslant b_i^1 + x_i$. Let t be equal to $b_i^1 + x_i + \theta$ where θ is a non-negative number. Then,

$$D_i(b_i^1;t+\Delta) - D_i(b_i^1;t) = D_i(b_i^1;b_i^1+x_i+\theta+\Delta) - D_i(b_i^1;b_i^1+x_i+\theta). \tag{39}$$

Since the deadline curve $D_i(\cdot)$ is a concave piecewise linear curve, if we apply Lemma 3 to Eq. (39),

$$D_i(b_i^1;b_i^1+x_i+\theta+\Delta) - D_i(b_i^1;b_i^1+x_i+\theta) \leqslant D_i(b_i^1;b_i^1+x_i+\Delta) - D_i(b_i^1;b_i^1+x_i). \tag{40}$$

In Eq. (37), if we replace t by $(b_i^1 + x_i + \Delta)$ and $(b_i^1 + x_i)$, $D_i(b_i^1;b_i^1+x_i+\Delta) - D_i(b_i^1;b_i^1+x_i)$ becomes as follows:

$$\min_{k=0\cdots K}\{C_i^{1,k}+\rho_i^k(b_i^1+x_i+\Delta)\} - \min_{k=0\cdots K}\{C_i^{m,k}+\rho_i^k(b_i^1+x_i)\}$$
$$= \min_{k=0\cdots K}\{\sigma_i^k-\rho_i^k(b_i^1+d_i)+\rho_i^k(b_i^1+x_i+\Delta)\} - \min_{k=0\cdots K}\{\sigma_i^k-\rho_i^k(b_i^1+d_i)+\rho_i^k(b_i^1+x_i)\}$$
$$= \min_{k=0\cdots K}\left\{\sigma_i^k+\rho_i^k\left(-\frac{\sigma_i^1}{\rho^{\max}}+\Delta\right)\right\} - \min_{k=0\cdots K}\left\{\sigma_i^k-\frac{\rho_i^k}{\rho^{\max}}\sigma_i^1\right\}$$
$$(\text{since } x_i=d_i-\sigma_i^1/\rho^{\max})$$
$$= S_i(x_i+\Delta)-\min_{k=0\cdots K}\left\{\sigma_i^k-\frac{\rho_i^k}{\rho^{\max}}\sigma_i^1\right\} \quad (\text{from Eq. (36)}). \tag{41}$$

In Eq. (41), since $\sigma_i^1 \leqslant \sigma_i^k$ and $\rho_i^k/\rho^{\max}\leqslant 1$ for $k=0,\ldots,K$,
$$\min_{k=0\cdots K}\left\{\sigma_i^k-\frac{\rho_i^k}{\rho^{\max}}\sigma_i^1\right\}\geqslant 0. \tag{42}$$
From Eqs. (41) and (42),
$$S_i(x_i+\Delta)-\min_{k=0\cdots K}\left\{\sigma_i^k-\frac{\rho_i^k}{\rho^{\max}}\sigma_i^1\right\}\leqslant S_i(x_i+\Delta). \tag{43}$$

Thus,
$$D_i(b_i^1;t+\Delta)-D_i(b_i^1;t)\leqslant S_i(x_i+\Delta). \tag{44}$$
By the induction hypothesis, for $t\geqslant b_i^m+x_i$,
$$D_i(b_i^m;t+\Delta)-D_i(b_i^m;t) = \min_{k=0\cdots K}\{C_i^{m,k}+\rho_i^k(b_i^m+x_i+\Delta)\} - \min_{k=0\cdots K}\{C_i^{m,k}+\rho_i^k(b_i^m+x_i)\} \leqslant S_i(x_i+\Delta). \tag{45}$$

As the induction step, consider the $(m+1)$th backlogged time point b_i^{m+1}. In this case, $t\geqslant b_i^{m+1}+x_i$. Let t be equal to $b_i^{m+1}+x_i+\theta$ where θ is a non-negative number. Then,
$$D_i(b_i^{m+1};t+\Delta)-D_i(b_i^{m+1};t)$$
$$= D_i(b_i^{m+1};b_i^{m+1}+x_i+\theta+\Delta) - D_i(b_i^{m+1};b_i^{m+1}+x_i+\theta)$$
$$\leqslant D_i(b_i^{m+1};b_i^{m+1}+x_i+\Delta) - D_i(b_i^{m+1};b_i^{m+1}+x_i) \quad (\text{by Lemma 3}).$$

In Eq. (37), if we replace t by $(b_i^{m+1}+x_i+\Delta)$ and $(b_i^{m+1}+x_i)$, $D_i(b_i^{m+1};b_i^{m+1}+x_i+\Delta)-D_i(b_i^{m+1};b_i^{m+1}+x_i)$ becomes as follows:
$$\min_{k=0\cdots K}\{C_i^{m+1,k}+\rho_i^k(b_i^{m+1}+x_i+\Delta)\} - \min_{k=0\cdots K}\{C_i^{m+1,k}+\rho_i^k(b_i^{m+1}+x_i)\}. \tag{46}$$

Since
$$C_i^{m+1,k}=\min\{C_i^{m,k},\sigma_i^k-\rho_i^k(b_i^{m+1}+d_i)+W_i^{\mathrm{RT}}(b_i^{m+1})\},$$
we can rewrite Eq. (46) as follows:
$$\min_{k=0\cdots K}\{\min\{C_i^{m,k}+\rho_i^k(b_i^{m+1}+x_i+\Delta),\sigma_i^k+\rho_i^k(x_i-d_i+\Delta)+W_i^{\mathrm{RT}}(b_i^{m+1})\}\} - \min_{k=0\cdots K}\{\min\{C_i^{m,k}+\rho_i^k(b_i^{m+1}+x_i),\sigma_i^k+\rho_i^k(x_i-d_i)+W_i^{\mathrm{RT}}(b_i^{m+1})\}\}. \tag{47}$$

For notational convenience, let us denote, for each k, $C_i^{m,k}+\rho_i^k(b_i^{m+1}+x_i)$ and $\sigma_i^k+\rho_i^k(x_i-d_i)+W_i^{\mathrm{RT}}(b_i^{m+1})$ by V^k and W^k, respectively. Then, Eq. (47) becomes as follows:
$$\min_{k=0\cdots K}\{\min\{V^k+\rho_i^k\Delta,W^k+\rho_i^k\Delta\}\} - \min_{k=0\cdots K}\{\min\{V^k,W^k\}\}$$
$$= \min\{\min_{k=0\cdots K}\{V^k+\rho_i^k\Delta\},\min_{k=0\cdots K}\{W^k+\rho_i^k\Delta\}\} - \min\{\min_{k=0\cdots K}\{V^k\},\min_{k=0\cdots K}\{W^k\}\}. \tag{48}$$

First, consider the case that $\min_{k=0\cdots K}\{V^k\}\geqslant \min_{k=0\cdots K}\{W^k\}$. In this case, Eq. (48) becomes as follows:
$$\min\{\min_{k=0\cdots K}\{V^k+\rho_i^k\Delta\},\min_{k=0\cdots K}\{W^k+\rho_i^k\Delta\}\} - \min_{k=0\cdots K}\{W^k\}\leqslant \min_{k=0\cdots K}\{W^k+\rho_i^k\Delta\}-\min_{k=0\cdots K}\{W^k\}$$
$$= \min_{k=0\cdots K}\left\{\sigma_i^k+\rho_i^k\left(\Delta-\frac{\sigma_i^1}{\rho^{\max}}\right)+W_i^{\mathrm{RT}}(b_i^{m+1})\right\} - \min_{k=0\cdots K}\left\{\sigma_i^k-\frac{\rho_i^k}{\rho^{\max}}\sigma_i^1+W_i^{\mathrm{RT}}(b_i^{m+1})\right\}$$
$$\leqslant \min_{k=0\cdots K}\left\{\sigma_i^k+\rho_i^k\left(\Delta-\frac{\sigma_i^1}{\rho^{\max}}\right)+W_i^{\mathrm{RT}}(b_i^{m+1})\right\} - W_i^{\mathrm{RT}}(b_i^{m+1}) \quad (\text{from Eq. (42)})$$

$$= \min_{k=0\cdots K} \left\{ \sigma_i^k + \rho_i^k \left(\Delta - \frac{\sigma_i^1}{\rho^{\max}} \right) \right\}$$
$$= S_i(\Delta + x_i) \text{ (from Eq. (36))}.$$

Next, consider the other case that $\min_{k=0\cdots K} \{V^k\} < \min_{k=0\cdots K} \{W^k\}$. In this case, Eq. (48) becomes as follows:

$$\begin{aligned}&\min\{\min_{k=0\cdots K} \{V^k + \rho_i^k \Delta\}, \min_{k=0\cdots K} \{W^k + \rho_i^k \Delta\}\} \\ &\quad - \min_{k=0\cdots K} \{V^k\} \leqslant \min_{k=0\cdots K} \{V^k + \rho_i^k \Delta\} - \min_{k=0\cdots K} \{V^k\} \\ &= \min_{k=0\cdots K} \{C_i^{m,k} + \rho_i^k (b_i^{m+1} + x_i + \Delta)\} \\ &\quad - \min_{k=0\cdots K} \{C_i^{m,k} + \rho_i^k (b_i^{m+1} + x_i)\} \\ &\leqslant \min_{k=0\cdots K} \{C_i^{m,k} + \rho_i^k (b_i^{m} + x_i + \Delta)\} \\ &\quad - \min_{k=0\cdots K} \{C_i^{m,k} + \rho_i^k (b_i^{m} + x_i)\} \text{ (by Lemma 3)} \\ &\leqslant S_i(\Delta + x_i) \text{ (from Eq. (45))}. \qquad \square\end{aligned}$$

Lemma 5. *Assume that a service curve $S_i(\cdot)$ is allocated by Eq.* (17) *for a $K(\sigma, \rho)$ session. Let x_i be the first time point at which the service curve $S_i(\cdot)$ has a non-zero value. At the mth backlogged time point b_i^m, consider an updated deadline curve $D_i(b_i^m; \cdot)$. Then, $E_i(b_i^m; \cdot)$ is reduced as follows*:

$$E_i(b_i^m; t) = D_i(b_i^m; t + x_i) \quad \textit{for } t \geqslant b_i^m. \tag{49}$$

Proof. Given the service curve $S_i(\cdot)$, at the mth backlogged time point b_i^m, the eligible time curve is updated as follows:

$$\begin{aligned}E_i(b_i^m; t) = D_i(b_i^m; t) &\\ + [\max_{t'>t} (D_i(b_i^m; t') - D_i(b_i^m; t) &\\ - S_i(t' - t))]^+ \quad \text{for } t \geqslant b_i^m.&\end{aligned} \tag{50}$$

For notational convenience, let us define $F_i(b_i^m; \cdot)$ as follows:

$$\begin{aligned}F_i(b_i^m; t) = [\max_{t'>t} (D_i(b_i^m; t') - D_i(b_i^m; t) &\\ - S_i(t' - t))]^+ \quad \text{for } t \geqslant b_i^m.&\end{aligned} \tag{51}$$

To prove the lemma, we show that $F_i(b_i^m; t)$ becomes equal to $D_i(b_i^m; t + x_i) - D_i(b_i^m; t)$. First, consider the case that $t' \in (t, t + x_i]$. (Skip this case if x_i is zero.) Since $D_i(\cdot)$ is a non-decreasing function, $D_i(b_i^m; t') - D_i(b_i^m; t)$ is less than or equal to $D_i(b_i^m; t + x_i) - D_i(b_i^m; t)$. In addition, $S_i(t' - t) = 0$ since $t' - t < x_i$ and for $u < x_i$, $S_i(u) = 0$. Thus, for $t' \in (t, t + x_i]$,

$$\begin{aligned}D_i(b_i^m; t') - D_i(b_i^m; t) - S_i(t' - t) &\\ \leqslant D_i(b_i^m; t + x_i) - D_i(b_i^m; t).&\end{aligned} \tag{52}$$

Next, consider the other case that t' is greater than $t + x_i$. Let t' be equal to $(t + x_i + \theta)$ where θ is a positive number. Then,

$$\begin{aligned}D_i(b_i^m; t') - D_i(b_i^m; t) - S_i(t' - t) &\\ = D_i(b_i^m; t + x_i + \theta) - D_i(b_i^m; t) - S_i(x_i + \theta).&\end{aligned} \tag{53}$$

Since $t \geqslant b_i^m$ and $\theta > 0$, by Lemma 4,

$$D_i(b_i^m; t + x_i + \theta) - D_i(b_i^m; t + x_i) \leqslant S_i(x_i + \theta). \tag{54}$$

From Eqs. (53) and (54),

$$\begin{aligned}D_i(b_i^m; t + x_i + \theta) - D_i(b_i^m; t) - S_i(x_i + \theta) &\\ \leqslant D_i(b_i^m; t + x_i) - D_i(b_i^m; t).&\end{aligned} \tag{55}$$

Thus, for $t' > t + x_i$,

$$\begin{aligned}D_i(b_i^m; t') - D_i(b_i^m; t) - S_i(t' - t) &\\ \leqslant D_i(b_i^m; t + x_i) - D_i(b_i^m; t). \qquad \square&\end{aligned} \tag{56}$$

Finally, using Lemma 5, we show in the following theorem that the cost to update each eligible time curve and that to calculate an eligible time become all constant.

Theorem 6. *Given a service curve allocated by Eq.* (17) *for a real-time session, at each backlogged time point, the eligible time curve can be updated in a constant time. In addition, subsequent eligible time calculations from the inverse of the eligible time curve can also be done in a constant time.*

Proof. First, we prove that updating each eligible time curve can be done in a constant time. It is sufficient to show that the eligible time curve $E_i(b_i^m; \cdot)$ presented in Lemma 5 can be updated in a constant time. By Lemma 5, at the mth backlogged time point b_i^m, $E_i(\cdot)$ is updated as follows:

$$E_i(b_i^m; t) = D_i(b_i^m; t + x_i) \quad \text{for } t \geqslant b_i^m. \tag{57}$$

To update $E_i(b_i^m;\cdot)$ in Eq. (57), we have only to shift $D_i(b_i^m;t)$ for $t \geqslant b_i^m + x_i$ to the left by x_i. As mentioned earlier, $D_i(b_i^m;t)$ for $t \geqslant b_i^m + x_i$ is a concave piecewise linear curve, which requires to maintain a fixed number of changing time points and slopes. Shifting such a curve can be done in a constant time.

Next, we show that eligible time calculations can be done in a constant time. The eligible time of each packet is calculated from the inverse of $E_i(b_i^m;\cdot)$ if b_i^m is the last backlogged time point. Note that since $E_i(b_i^m;\cdot)$ is obtained by shifting a concave piecewise linear curve $D_i(b_i^m;\cdot)$, $E_i(b_i^m;\cdot)$ is also concave piecewise linear. So, the cost for computing eligible times is constant. □

6. Simulation experiments

In this section, we show, through a simulation study, that if H-FSC adopts our service curve allocation scheme, it can achieve a high level of network utilization. We consider three scheduling algorithms, i.e., the generalized H-FSC which adopts our allocation scheme, the H-FSC with the scheme proposed in [22], and the multirate algorithm proposed in [16] which uses multiple service rates like our scheme. To compare the level of network utilization, we measure and compare the number of sessions admitted by the three algorithms.

We consider various MPEG-coded video traces [3] for the experiments. Table 1 presents the characterized traffic envelope functions for the MPEG videos. In the table, the first three videos are movie, the next three are sports, and the last two are news and talk videos. Each frame is of 384×288 pixels. The frame rate is 25 frames per second. Each video consists of 40,000 frames which is equivalent to approximately half an hour. We have measured the maximum sizes of I, B, and P frames. The encoding pattern of all the MPEG videos is IBBPBBPBBPBB. We have characterized all the videos with $3(\sigma, \rho)$ envelope functions using the maximum sizes of each frame type.

We have measured the number of videos each algorithm can admit with various delay bound requirements. The link bandwidth is fixed to 100 Mbps. The maximum packet size is assumed to be 1500 bytes. [4] Fig. 14 illustrates the number of the Jurassic Park videos each algorithm can admit with 11, 22, 45, and 90 ms delay bound requirements. We do not present the results with other videos since they show similar results.

In Fig. 14, the generalized H-FSC outperforms both the H-FSC and the multirate algorithm. In the figure, the shorter the delay bound requirement becomes, the more videos the generalized H-FSC can admit compared to the H-FSC. For example, when the delay bound is 11 ms, the generalized H-FSC can admit about three times that the H-FSC can. Thus, the advantage of the generalized H-FSC becomes very clear for on-line real-time applications, e.g., video phone and video conferencing. Such online applications usually require a very short end-to-end delay bound. Stored applications such as video-on-demand may tolerate a longer end-to-end delay. The generalized H-FSC is much beneficial even for stored applications. Even though a long delay can be tolerated for such applications, the delay which should be met in a router becomes quite tight since there are multiple routers in a wide-area network and the given end-to-end delay bound should be distributed over multiple routers. Comparing the H-FSC and the multirate algorithm, one is better than the other depending on the different delay bound requirements. If the delay bound requirement is short, the multirate algorithm outperforms the H-FSC, while with a long delay bound requirement, the H-FSC is better than the multirate algorithm.

[3] These traces can be found in ftp-info3.informatik.uni-wuerzburg.de/pub/MPEG.

[4] The multirate algorithm was studied in the ATM network in which all packets have the same size of 53 bytes. Thus, in the case of the multirate algorithm, all packets are assumed to have the size of 53 bytes.

Table 1
Characterized traffic envelope functions for the MPEG-coded video data

Video title	Characterized traffic envelope function $b_i(t)$
Jurassic Park	$\min\{365Kt, 5924 + 220Kt, 9461 + 211Kt\}$
Star Wars	$\min\{380Kt, 8449 + 174Kt, 14959 + 158Kt\}$
The Silence of the Lambs	$\min\{409Kt, 8505 + 201Kt, 22515 + 167Kt\}$
ATP tennis	$\min\{582Kt, 8588 + 372Kt, 34168 + 310Kt\}$
Formula race	$\min\{567Kt, 1034 + 542Kt, 16438 + 504Kt\}$
Soccer	$\min\{580Kt, 8802 + 365Kt, 31989 + 309Kt\}$
News	$\min\{593Kt, 11576 + 310Kt, 59540 + 193Kt\}$
Talk	$\min\{325Kt, 7448 + 143Kt, 21428 + 109Kt\}$

The unit is in byte, 1 K = 1024.

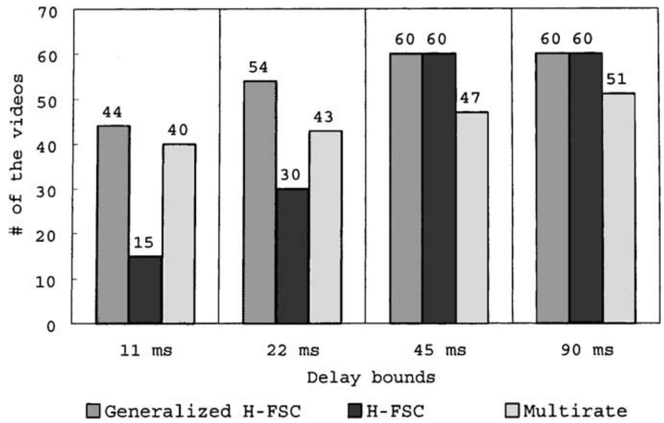

Fig. 14. The number of the Jurassic Park video admitted in each algorithm according to various delay bound requirements.

7. Related work

In the literature, many scheduling algorithms have been studied to provide a real-time service, a link-sharing service, or both. Most of those algorithms can be categorized into three major trends, namely, generalized processor sharing (GPS) algorithms [1,2,9,13,14,16,20], rate-controlled (RC) algorithms [5,8,25], and service curve (SC) algorithms [4,17,18,22].

7.1. GPS algorithms

GPS scheduling algorithms, e.g., GPS [13,14], worst-case fair weighted fair queueing (WF^2Q) [2], H-PFQ [1], self-clocked fair queueing (SCFQ) [9], rate-proportional servers [20], and multirate algorithm [16], are rate-based algorithms to provide real-time services. Usually these rate-based algorithms provide for each session a delay bound depending on its traffic envelope function. (It is theoretically possible to provide for a session a delay bound independent of its traffic envelope function. However, such cases are not practical [14,22].) Thus, compared to the generalized H-FSC, the achievable network utilization by those algorithms are low. In addition, they do not provide hierarchical link-sharing services.

To provide both the real-time and the hierarchical link-sharing services, a hierarchical version of GPS called H-GPS is studied in [1]. However, as mentioned in Section 1, H-GPS has a lower level of network utilization than the generalized H-FSC.

7.2. RC algorithms

RC scheduling algorithms [5,8,24,25] consist of regulators and a packet scheduler. In these algorithms, the roles for allocating a service rate and a delay bound for a session are separated to regulators and a packet scheduler, respectively. One or more regulators are allocated to each session according to the traffic envelope function. Regulators for each session control the output rate of the session. A different packet scheduler such as a static-priority (SP) scheduler or an EDF scheduler can be adopted. The RC algorithm adopting the SP/EDF scheduler is called an RC-SP/RC-EDF algorithm.

Note that all RC algorithms have a severe difficulty in providing a link-sharing service due to regulators. To give excess service to a session with regulators, a complex queue management mechanism is needed to make packets bypass the regulators [8,26].

In providing a real-time service, the level of network utilization as well as implementation cost and scheduling complexity are affected by the adopted scheduler. First, consider the case of the RC-EDF algorithm [5,8]. As mentioned, it has the same level of network utilization, but higher implementation cost than the generalized H-FSC due to the regulators. It has to implement at least N [5] regulators, where N is the number of sessions. In [24], a method using a calendar queue has been proposed to avoid implementing separate regulators for each session. However, although a calendar queue is used, more than N packets may have to be moved from the calendar queue to the EDF scheduler within a short duration. Note that the generalized H-FSC requires no regulators at all since it computes deadlines for packets smartly; both the traffic envelope function and the delay bound are considered to compute deadlines.

Next, let us consider the RC-SP algorithm. The SP scheduler maintains a certain number of priority groups in advance. A session is allocated to a priority group during admission test. The SP scheduler transmits a packet in the highest priority group that has packets in the queue. Compared to the generalized H-FSC, the RC-SP suffers from low level of network utilization [12]. However, the RC-SP has an advantage in scheduling complexity if a calendar queue method is used [24,25]. In this case, the number of priority groups is usually a small constant. Thus, moving packets from the calendar queue to the SP scheduler can be done in a constant time.

7.3. SC algorithms

Service curve earliest deadline (SCED) algorithm [4,18] is a general service curve algorithm which provides a real-time service. First of all, compared to the generalized H-FSC, SCED does not provide the hierarchical link-sharing service. Next, although SCED is similar to the service curve algorithm used in H-FSC, it considers only a fixed-sized packet environment that is applicable to ATM networks. In a variable-sized environment such as the Internet, packets arrive and depart asynchronously. Recall that we have defined a different guaranteed service curve to handle these differences. (Although the authors in [3] have studied how to allocate a service curve in SCED under a variable-sized environment, they do not address the issue of heavy implementation cost imposed by asynchronous packet arrivals as discussed in Section 2.)

8. Conclusions

In this paper, we have proposed a scheduling algorithm which achieves a high level of network utilization and link-sharing simultaneously. The proposed algorithm is an extension of the H-FSC generalized by a new service curve allocation scheme. By the new scheme, the generalized H-FSC can achieve a high level of network utilization even for VBR video data. Table 2 compares the H-FSC with the generalized H-FSC in summary. In the table, K is a small constant and N is the number of sessions. We have shown by simulation

[5] When each session is characterized by a $K(\sigma, \rho)$ traffic envelope function, KN number of regulators are required in total.

Table 2
Comparison of H-FSC and generalized H-FSC

	H-FSC	Generalized H-FSC
Service curve	Two piecewise linear	K piecewise linear
Network utilization	Low for VBR	High for both CBR and VBR
Total scheduling complexity	$O(\log N)$	$O(\log N)$
Deadline computation	$O(1)$	$O(K)$
Eligible time computation	$O(1)$	$O(K)$
Virtual time computation	$O(1)$	$O(K)$
Packet transmission	$O(\log N)$	$O(\log N)$

experiments that the generalized H-FSC can admit a significantly higher number of VBR video sessions than H-FSC and the multirate algorithm. Even with high network utilization, the generalized H-FSC requires the same scheduling complexity as the H-FSC.

Acknowledgements

This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the Advanced Information Technology Research Center (AITrc). The authors appreciate the anonymous referees for their helpful comments to improve the quality of this paper significantly.

Appendix A

Proof of Theorem 1. We denote the received amount of data from session i at the mth router until a time t by $A_i^m(t)$. We also denote $W_i^{\mathrm{RT}}(t)$ and $W_i^{\mathrm{ES}}(t)$ at the mth router that session i passes through by $W_i^{m,\mathrm{RT}}(t)$ and $W_i^{m,\mathrm{ES}}(t)$, respectively. For notational convenience, we denote the source of session i by the 0th router such that $W_i^0(\cdot)$ becomes equal to $A_i^1(\cdot)$.

Consider a time t when a packet from session i departs the final Mth router. Starting from the time t, we define $(M+1)$ time points, i.e., t_m, $0 \leqslant m \leqslant M$, as following. First, we define $t_M = t$. Given the fixed t_M, t_{M-1} is defined as a time such that $t_{M-1} < t_M$ and

$$W_i^{M,\mathrm{RT}}(t_M) \geqslant W_i^{M,\mathrm{RT}}(t_{M-1}) + S_i^M(t_M - t_{M-1}). \tag{A.1}$$

Since the Mth router strictly guarantees $S_i^M(\cdot)$ for session i, the above t_{M-1} exists and t_{M-1} is the beginning of one of session i's backlogged periods. From Eq. (A.1), since $W_i^{M,\mathrm{RT}}(\cdot)$ is equal to $W_i(\cdot) - W_i^{M,\mathrm{ES}}(\cdot)$,

$$W_i^M(t_M) - W_i^{M,\mathrm{ES}}(t_M) \geqslant W_i^M(t_{M-1}) - W_i^{M,\mathrm{ES}}(t_{M-1}) + S_i^M(t_M - t_{M-1}).$$

By arranging the above terms,

$$W_i^M(t_M) \geqslant W_i^M(t_{M-1}) + W_i^{M,\mathrm{ES}}(t_{M-1}, t_M) + S_i^M(t_M - t_{M-1}). \tag{A.2}$$

From Eq. (A.2), $W_i^M(t_{M-1})$ is equal to $A_i^M(t_{M-1})$ since the queue for session i is empty at the time t_{M-1}. Thus,

$$W_i^M \geqslant A_i^M(t_{M-1}) + W_i^{M,\mathrm{ES}}(t_{M-1}, t_M) + S_i^M(t_M - t_{M-1}) \tag{A.3}$$

$$= W_i^{M-1}(t_{M-1}) + W_i^{M,\mathrm{ES}}(t_{M-1}, t_M) + S_i^M(t_M - t_{M-1}). \tag{A.4}$$

By following the above steps recursively, we can define the times $t_{M-2}, t_{M-3}, \ldots, t_0$ that satisfy an equation similar to Eq. (A.4). Therefore, for the time t, given the $(M+1)$ time points,

$$\begin{aligned}W_i^M(t) &= W_i^M(t_M)\\ &\geqslant W_i^{M-1}(t_{M-1}) + W_i^{M,\mathrm{ES}}(t_{M-1},t_M)\\ &\quad + S_i^M(t_M - t_{M-1})\\ &\geqslant W_i^{M-2}(t_{M-2}) + \sum_{m=M-1}^{M}[W_i^{m,\mathrm{ES}}(t_{m-1},t_m)\\ &\quad + S_i^m(t_m - t_{m-1})]\\ &\cdots\\ &\geqslant W_i^0(t_0) + \sum_{m=1}^{M}[W_i^{m,\mathrm{ES}}(t_{m-1},t_m)\\ &\quad + S_i^m(t_m - t_{m-1})]\\ &= A_i^1(t_0) + \sum_{m=1}^{M}[W_i^{m,\mathrm{ES}}(t_{m-1},t_m)\\ &\quad + S_i^m(t_m - t_{m-1})]\\ &\geqslant A_i^1(t_0) + \sum_{m=1}^{M}[S_i^m(t_m - t_{m-1})]\\ &\geqslant A_i^1(t_0) + S_i^{\mathrm{net}}(t_M - t_0). \qquad \square\end{aligned}$$

Proof of Theorem 2. Consider a time t that a packet p from session i departs the router. Let us denote the arrival time and the delay of the packet p by a and d, respectively $(t = a + d)$. We derive the worst-case delay of the packet p. Since $S_i(\cdot)$ is guaranteed for session i, there exists a time s, $s < t$, which is the beginning of one of session i's backlogged periods such that

$$W_i(t) \geqslant A_i(s) + S_i(t - s). \tag{A.5}$$

The packets which arrived before the packet p plus the packet p are transmitted until the packet departure time t. However, if multiple packets arrive at the packet arrival time a, those packets are not transmitted until the time t. Thus,

$$A_i(a) \geqslant W_i(t). \tag{A.6}$$

From Eq. (A.6) and Eq. (A.5),

$$A_i(a) - A_i(s) \geqslant S_i(t - s). \tag{A.7}$$

Since the time s is not greater than the last backlogged time point, the packet arrival time a is greater than the time s, i.e., $a > s$. In addition, since session i has the traffic envelope function $b_i(\cdot)$,

$$b_i(a - s) \geqslant A_i(a) - A_i(s). \tag{A.8}$$

From Eqs. (A.7) and (A.8),

$$b_i(a - s) \geqslant S_i(t - s) = S_i(a - s + d). \tag{A.9}$$

To satisfy Eq. (A.9),

$$\begin{aligned}d &\leqslant \min\{\Delta\colon \Delta > 0 \text{ and } b_i(a - s) \leqslant S_i(a - s + \Delta)\}\\ &\leqslant \max_{k:k>0} \min\{\Delta\colon \Delta > 0 \text{ and } b_i(k) \leqslant S_i(k + \Delta)\}.\end{aligned} \tag{A.10}$$

From Eq. (A.10), for the packet p, the worst-case delay becomes $\max_{k:k>0} \min\{\Delta : \Delta > 0 \text{ and } b_i(k) \leqslant S_i(k + \Delta)\}$. Note that, however, all the packets including the packet p have this delay bound since Eq. (A.10) is independent of the packet departure time t, which is a special packet departure time for the packet p. $\square$

Proof of Theorem 3. Consider a time t that a packet p from session i departs the router. Since $S_i(\cdot)$ is guaranteed for session i, there exists a time s, $s < t$, which is the beginning of one of session i's backlog periods such that

$$W_i(t) \geqslant A_i(s) + S_i(t - s). \tag{A.11}$$

Let us denote the backlog of session i at the packet departure time t by $B_i(t)$. First of all, we derive the worst-case backlog at the time t. (Later, we consider all packet departure times and also the times that are not packet departure times.) The backlog amount at the time t is the arrived amount until the time t minus the transmitted amount until the time t. Thus,

$$\begin{aligned}B_i(t) &= A_i(t) - W_i(t)\\ &\leqslant A_i(t) - A_i(s) - S_i(t - s)\\ &\quad \text{(from Eq. (A.11))}\\ &= A_i(s, t) - S_i(t - s)\\ &\leqslant b_i(t - s) - S_i(t - s)\\ &\leqslant \max_{k:k>0}\{b_i(k) - S_i(k)\}.\end{aligned} \tag{A.12}$$

From Eq. (A.12), the worst-case backlog at the packet departure time t becomes $\max_{k:k>0}\{b_i(k) - S_i(k)\}$. Note that, however, this backlog bound holds for all packet departure times since Eq. (A.12) is independent of the time t, which is a special packet departure time. Now, we consider all times including those when no packets depart the router. Only one packet departs the router between packet departure times. Thus, for all times, the worst-case backlog amount cannot exceed $\max_{k:k>0}\{b_i(k) - S_i(k)\} + l^{\max}$ to satisfy Eq. (A.12) at all packet departure times. □

Proof of Theorem 4. We focus on a packet p of a leaf node i. The packet p becomes head at the time t_c during the mth backlogged period of the session, denoted by b_i^m. The departure time, the length, and the deadline of the packet p are denoted by d, l, and D, respectively. At the time t_c, the accumulated amount of data including the packet p becomes $W_i^{\mathrm{RT}}(t_c) + l$, which is equal to $W_i^{\mathrm{RT}}(d)$. Given the packet p and the accumulated amount, the deadline D becomes the smallest t such that $D_i(b_i^m; t) = W_i^{\mathrm{RT}}(d)$. Thus, for a time t' which is less than or equal to D,

$$D_i(b_i^m; t') \leqslant W_i^{\mathrm{RT}}(d). \tag{A.13}$$

Since all leaf nodes have passed the admission test, all the packets are transmitted until their deadlines plus $l^{\max}/C$. Thus, the packet p departs until D plus $l^{\max}/C$, i.e.,

$$d - l^{\max}/C \leqslant D. \tag{A.14}$$

From Eqs. (A.13) and (A.14),

$$W_i^{\mathrm{RT}}(d) \geqslant D_i(b_i^m; d - l^{\max}/C). \tag{A.15}$$

From Eq. (A.15), to prove H-FSC strictly guarantees $\widehat{S}_i(t)$ for leaf node i, it is sufficient to show that

$$D_i(b_i^m; d - l^{\max}/C) \geqslant \min_{s\in B_i(d)} \{W_i^{\mathrm{RT}}(s) + \widehat{S}_i(d-s)\}. \tag{A.16}$$

First, consider the case that $d - l^{\max}/C \geqslant b_i^m$. In this case, from Eq. (7),

$$D_i(b_i^m; d - l^{\max}/C) = \min_{s\in B_i(b_i^m)} \{W_i^{\mathrm{RT}}(s) + S_i(d - l^{\max}/C - s)\}. \tag{A.17}$$

Since there exists no new backlogged period of the session during the interval $(b_i^m, d]$,

$$\min_{s\in B_i(b_i^m)} \{W_i^{\mathrm{RT}}(s) + S_i(d - l^{\max}/C - s)\} = \min_{s\in B_i(d)} \{W_i^{\mathrm{RT}}(s) + S_i(d - l^{\max}/C - s)\}. \tag{A.18}$$

Next, consider the other case that $d - l^{\max}/C < b_i^m$. In this case, from Eq. (7),

$$\begin{aligned} D_i(b_i^m; d - l^{\max}/C) &= \min_{s\in B_i(b_i^{m-1})} \{W_i^{\mathrm{RT}}(s) + S_i(d - l^{\max}/C - s)\} \\ &\geqslant \min_{s\in B_i(b_i^m)} \{W_i^{\mathrm{RT}}(s) + S_i(d - l^{\max}/C - s)\} \\ &= \min_{s\in B_i(d)} \{W_i^{\mathrm{RT}}(s) + S_i(d - l^{\max}/C - s)\}. \end{aligned} \tag{A.19}$$

From Eqs. (A.18) and (A.19),

$$D_i(b_i^m; d - l^{\max}/C) \geqslant \min_{s\in B_i(d)} \{W_i^{\mathrm{RT}}(s) + S_i(d - l^{\max}/C - s)\} \tag{A.20}$$

$$= \min_{s\in B_i(d)} \{W_i^{\mathrm{RT}}(s) + \widehat{S}_i(d-s)\}. \quad \square \tag{A.21}$$

References

[1] J.C.R. Bennett, H. Zhang, Hierarchical packet fair queueing algorithms, IEEE/ACM Trans. Network. 5 (5) (1997) 675–689.

[2] J.C.R. Bennett, H. Zhang, WF^{2}Q: worst-case fair weighted fair queueing, in: INFOCOM, 1996, pp. 120–128.

[3] J.-Y. Le Boudec, P. Thiran, Network Calculus. A Theory of Deterministic Queueing Systems for the Internet, Lecture Notes in Computer Science, vol. 2050, Springer, Berlin.

[4] R.L. Cruz, Quality of service quarantees in virtual circuit switched networks, IEEE J. Select. Areas Commun. 13 (6) (1995) 1048–1056.

[5] D. Ferrari, D. Verma, A scheme for real-time channel establishment in wide-area networks, IEEE J. Select. Areas Commun. 8 (4) (1990) 368–379.

[6] S. Floyd, V. Jacobson, Link-sharing and resource management models for packet networks, IEEE/ACM Trans. Network. 3 (4) (1995) 365–386.

[7] L. Georgiadis, R. Guérin, A. Parekh, Optimal multiplexing on a single link: Delay and buffer requirements, IEEE Trans. Inform. Theory 43 (5) (1997) 1518–1535.

[8] L. Georgiadis, R. Guérin, V. Peris, K.N. Sivarajan, Efficient network QoS provisioning based on per node traffic shaping, IEEE/ACM Trans. Network. 4 (4) (1996) 482–501.

[9] S. Golestani, A self-clocked fair queueing scheme for broadband applications, in: INFOCOM, 1994, pp. 636–646.
[10] M.R. Hashemi, A. Leon-Garcia, The single-queue switch: A building block for switches with programmable scheduling, IEEE J. Select. Areas Commun. 15 (5) (1997) 785–794.
[11] P. Lavoie, Y. Savaria, A systolic architecture for fast stack sequential decoders, IEEE Trans. Commun. 42 (2/3/4) (1994) 324–335.
[12] J. Liebeherr, D.E. Wrege, D. Ferrari, Exact admission control for networks with a bounded delay service, IEEE/ACM Trans. Network. 4 (6) (1996) 885–901.
[13] A.K.J. Parekh, R.G. Gallager, A generalized processor sharing approach to flow control in integrated service networks: The single node case, IEEE/ACM Trans. Network. 1 (3) (1993) 344–357.
[14] A.K.J. Parekh, R.G. Gallager, A generalized processor sharing approach to flow control in integrated service networks: The multiple node case, IEEE/ACM Trans. Network. 2 (2) (1994) 137–150.
[15] K. Pyun, J. Song, H.-K. Lee, The service curve service discipline with the service curve discipline for the rate-controlled EDF service discipline in variable-sized packet networks, in: IEEE International Conference on Communications, 2002, pp. 1135–1141.
[16] D. Saha, S. Mukherjee, S.K. Tripathi, Multirate scheduling of VBR video traffic in ATM networks, IEEE J. Select. Areas Commun. 15 (6) (1997) 1132–1147.
[17] H. Sariowan, A service-curve approach to performance guarantees in integrated service networks, Ph.D. thesis, University of California, San Diego, 1996.
[18] H. Sariowan, R.L. Cruz, G.C. Polyzos, SCED: A generalized scheduling policy for guaranteeing quality-of-service, IEEE/ACM Trans. Network. 7 (5) (1999) 669–684.
[19] S. Shenker, C. Partridge, R. Guérin, Specification of Guaranteed Quality of Service, September 1997, RFC 2212.
[20] D. Stiliadis, A. Varma, Rate-proportional servers: A design methodology for fair queueing algorithms, IEEE/ACM Trans. Network. 6 (2) (1998) 164–174.
[21] I. Stoica, H. Zhang, T.S. Eugene Ng, A hierarchical fair service curve algorithm for link-sharing, real-time and priority services, Technical Report CMU-CS-99-168, Carnegie Mellon Univ., Pittsburgh, PA, September 1999.
[22] I. Stoica, H. Zhang, T.S. Eugene Ng, A hierarchical fair service curve algorithm for link-sharing, real-time and priority services, IEEE/ACM Trans. Network. 8 (2) (2000) 185–199.
[23] D.E. Wrege, E.W. Knightly, H. Zhang, J. Liebeherr, Deterministic delay bounds for VBR video in packet-switching networks: Fundamental limits and practical trade-offs, IEEE/ACM Trans. Network. 4 (3) (1996) 352–362.
[24] H. Zhang, Service disciplines for guaranteed performance service in packet-switching networks, Proc. IEEE 3 (4) (1995) 391–430.
[25] H. Zhang, D. Ferrari, Rate-controlled service disciplines, J. High Speed Networks 3 (4) (1994) 389–412.
[26] H. Zhang, S. Keshav, Rate-controlled static-priority queueing, in: INFOCOM, 1993, pp. 227–236.

Kihyun Pyun received the B.S. degree in Computer Science Engineering from the Inha University, Inchon, Republic of Korea, in 1995, and both M.S. and Ph.D. degree in Computer Science from the Korea Advanced Institute of Science and Technology (KAIST) in 1997 and 2003, respectively. Currently, he is a research staff in Electrical Engineering of KAIST as a post doctor. His major interests are resource control, quality-of-service, packet scheduling, and service systems for wired and wireless networks.

Junehwa Song is an assistant professor, Department of EECS, Korea Advanced Institute of Science and Technology (KAIST), Korea. Before joining KAIST, he worked at IBM T. J. Watson Research Center, Yorktown Heights, NY as a Research Staff Member from 1997 to September 2000. He received his Ph.D. in Computer Science from University of Maryland at College Park in 1997. His research interest lies in Internet Technologies, such as intermediary devices, high performance Web serving, electronic commerce, etc., and distributed multimedia systems.

Heung-Kyu Lee received the B.S. degree in Electronics Engineering from the Seoul National University, Seoul, Korea, in 1978, and M.S., Ph.D. degrees in Computer Science from the Korea Advanced Institute of Science and Technology (KAIST), in 1981, and 1984, respectively. From 1984 to 1985 he served as a post-doc at the University of Michigan, Ann Arbor. Since 1986 he has been a professor in the Department of Computer Science, the Korea Advanced Institute of Science and Technology, Taejon, Korea 305-701 (hklee@casaturn.kaist.ac.kr). He is now director of the URI-forum in the Korea. His major interests are digital watermarking, real-time processing, and image processing.