Available online at www.sciencedirect.com

ScienceDirect

Computer Networks xxx (2008) xxx–xxx

Computer Networks

www.elsevier.com/locate/comnet

Enhanced bulk scheduling for supporting delay sensitive streaming applications

Yung-Cheng Tu [a,*], Meng Chang Chen [b], Yeali S. Sun [c], Wei-Kuan Shih [a]

[a] *Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan, ROC*
[b] *Institute of Information Science, Academia Sinica, Taipei, Taiwan, ROC*
[c] *Department of Information Management, National Taiwan University, Taipei, Taiwan, ROC*

Received 21 February 2007; received in revised form 4 November 2007; accepted 27 November 2007

Responsible Editor: Prof. P. Frossard

Abstract

Providing end-to-end delay guarantees for delay sensitive applications is an important packet scheduling issue with routers. In this paper, to support end-to-end delay requirements, we propose a novel network scheduling scheme, called the bulk scheduling scheme (BSS), which is built on top of existing schedulers of intermediate nodes without modifying transmission protocols on either the sender or receiver sides. By inserting special control packets, which called TED (Traffic Specification with End-to-end Deadline) packets, into packet flows at the ingress router periodically, the BSS schedulers of the intermediate nodes can dynamically allocate the necessary bandwidth to each flow to enforce the end-to-end delay, according to the information in the TED packets. The introduction of TED packets incurs less overhead than the per-packet marking approaches. Three flow bandwidth estimation methods are presented, and their performance properties are analyzed. BSS also provides a dropping policy for discarding late packets and a feedback mechanism for discovering and resolving bottlenecks. The simulation results show that BSS performs efficiently as expected.

Keywords: Bandwidth allocation; End-to-end delay; Bulk scheduling; Quality of service

1. Introduction

Many real-time applications rely on networks to support quality of service (QoS) guarantees, typically in the form of bounded end-to-end delays. The increasingly popular delay sensitive Internet streaming applications such as Internet TV and video streaming require a loose end-to-end delay bound. For the viewing pleasure, the streaming applications only allow a small percentage of late or dropped packets. However, those applications do not accept a long start latency [1], which means a large buffer on the user side. A common practice in supporting end-to-end delay requirements is to reserve a fixed bandwidth along the path of the packet flow. The bandwidth reservation method is suitable for constant-bit-rate (CBR) traffic, because

[*] Corresponding author. Tel.: +886 9 19526562.
E-mail address: albert@rtlab.cs.nthu.edu.tw (Y.-C. Tu).

1389-1286/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2007.11.020

the delay bound in each node can be accurately estimated, and consequently, the end-to-end delay bound can be obtained. However, for most applications, such as video, audio, and transaction processing, the traffic is often variable-bit-rate (VBR). Methods that combine the bandwidth reservation with buffer management have been proposed for VBR multimedia traffic [2–5]. These methods require users to give traffic specifications in advance, among which the most notable one is the T-spec in RSVP [6] and ATM [7]. However, it is difficult for users to provide such traffic specifications of real-time streaming applications. Another approach, called the effective bandwidth or equivalent capacity [8] method, uses a statistical technique to provide a statistical performance guarantee. It also has difficulty in capturing the actual traffic generation process of applications to derive proper effective bandwidth.

An alternative to fixed bandwidth allocation is to dynamically allocate bandwidth and/or reschedule packets in real-time during the course of transmission based on the actual offered load, the QoS requirements of the session, and the status of transmission. Per-packet deadline scheduling, proposed in [9], assumes that a flow has a QoS profile stored in each intermediate node along the flow path. The arriving packet of the flow is assigned a nodal delay bound for each intermediate node, and the nodal scheduler schedules the next packet for transmission according to the nodal delay bounds of packets. Compared with rate-based scheduling, this approach can achieve better performance for VBR traffic, because each packet is scheduled according to its own delay bound. However, the QoS profile for each flow must be given to each intermediate node in advance, and must generally be fixed during the course of the session. In addition, this method favors flows with QoS guarantees and may cause starvation of the best-effort flows.

There are other studies related to dynamic bandwidth allocation. In ATM, special cells, called Resource Management cells, are used in available bit rate (ABR) service to probe the bandwidth in the network [10]. In [11], an ATM block transfer (ABT) scheme was proposed, in which the traffic flow is divided into blocks and each block requests bandwidth individually. Another similar approach was proposed in [12], whereby the sender divides the data stream into bursts during the scheduling phase and the first packet of each burst specifies its traffic rate and burst size. In this scheme, the sender needs to be able to communicate the application-level semantics to the lower layer protocol entities so that they can segment the byte stream into bursts.

Although the above works tackle the dynamic bandwidth requirement of VBR traffic, they do not address the nodal and end-to-end delay requirements. Some researches on the delay bound guarantee have focused on the single-node case. Such approaches derive a worst-case delay bound at single-node and obtain the end-to-end delay bound by adding up the nodal delays [13,14]. These approaches do not consider the use of dynamic scheduling at intermediate nodes to expedite packets with longer delays in previous nodes, or to loosen bandwidth allocation to the packets ahead of their schedules. To overcome the above problems, we propose the bulk scheduling scheme to support end-to-end delay requirements for delay sensitive applications. The idea of the proposed scheme is to insert special control packets, called TED (Traffic Specification with End-to-end Deadline) packets, into the flow. The TED packets carry information that the schedulers in routers can use to dynamically allocate necessary bandwidth to flows in order to meet their end-to-end deadlines. The insertion (deletion) of TED packets is performed at the ingress (egress) routers by the lower layer protocol entities independent of existing systems and applications.

With the bulk scheduling scheme, each nodal scheduler in an intermediate node on the flow path schedules packets per bulk, according to the associated TED packet, and only the contents of TED packets are modified by the intermediate nodes. Our approach can be classified as a coordinated scheduler, which is fully discussed and shown to outperform traditional fixed rate/priority schedulers in [15,16]. However, those proposed coordinated schedulers (such as Zhu's [17] and PHBs [18]) all need to do per-packet marking which results in high system load in routers. For per-packet marking approaches, the routers must capture the end-to-end delay bound information, change the priority or service rate and store new information for each packet. But in our bulk scheduling scheme, these are only performed on TED packets. Therefore, our bulk scheduling scheme has lower computational overhead than per-packet markings. The bulk scheduling scheme does not change any information in the data packets, which makes end-to-end secure transmissions (such as IPsec [19]) possible. Furthermore, the proposed scheduling algorithms consider the delay conditions that TED packets have experienced, rather than just the fixed priority or deadline which was assigned when the packets were dispatched by the sender.

The remainder of this paper is organized as follows. In Section 2, we discuss the related works on dynamic scheduling with delay information. In Section 3, we present the proposed end-to-end bulk scheduling scheme in detail, together with BSS flow bandwidth estimation methods for scheduling packets in intermediate nodes. In addition, the bulk scheduling scheme is analyzed in this section. In Section 4, we propose two enhancements of the bulk scheduling scheme: a dropping policy and a feedback mechanism. In Section 5, we present simulation results, which show the characteristics and strengths of our proposed scheme. Finally, in Section 6, we give conclusions.

2. Related works

2.1. Per-packet scheduling

The method for recording information in real-time traffic was also used by Zhu et al. [17]. They proposed a per-node deadline-curve scheduling scheme to guarantee the end-to-end delay bound. The per-node deadlines of a packet are specified by the sender and the scheduler in each intermediate node using the Earliest Deadline First scheme. However, since the schedulers in the intermediate nodes have no means of knowing the source arrival time of a packet, each packet needs to carry time-stamp information in its header.

The method proposed in [17] transforms the deadline information of each packet into nodal delay bounds, which represents the time that the packet can be delayed in its intermediate nodes. If the actual delay of a packet in a node is less than its nodal delay bound, the saving delay time is added to the delay budget, which is recorded in the packet header and used to compute the nodal delay bound in the next node. If a packet experiences less delay than the nodal delay bound, it will have a looser nodal delay bound, i.e., it can have longer delay time, for scheduling in subsequent nodes. In [17], the authors also proved that the proposed per-node deadline-curve scheduling scheme can guarantee the end-to-end delay bound.

However, to obtain the delay budget for packets, the scheme must record the delay condition in each packet header. This will cause two problems: (1) there may not be enough space to record the information in the packet header; and (2) the per-packet computation and marking is a heavy overhead for an intermediate node. Another work of Claypool et al. in [20] also uses the per-packet marking to control the packet delay, but it still have the above problems.

2.2. Nodal delay assignment problem

With regard to the problem of how to assign the per-node delay, Vagish et al. [21] assumed that the traffic rate specification and the load of each node are known before a connection is established. They proposed several strategies for assigning specific delay values for all nodes along the routing path, so that the end-to-end delay requirement of the connection is satisfied. Their objective is to maximize network resource utilization, i.e., the number of connections accepted. The goal of the algorithm given in [21] is to achieve a balance among the amounts of resources used at each node. Suppose there are N_i nodes along the routing path of flow i and the end-to-end delay bound of flow i is $d^i_{\max}$. Then, the assignment of the delay bound at the kth node, denoted by $d_{i,k}$, is given by

$$d_{i,k} = d_i^{\max}/N_i \quad \text{where } k = 1 \text{ to } N_i. \tag{1}$$

As the optimal value may not be feasible because of insufficient network resources, Vagish et al. first compute the upper bound and lower bound of the feasible delay at each node. They then assign the feasible delay to each node between the upper and lower bounds. Since the computation of the feasible delay at each node is based on the information received before the connection is established, the accurate upper and lower bounds are difficult to obtain if the load at each node exhibits large variance. The traffic rate specification is also hard to formulate if the traffic of a specific connection has a real-time variable-bit-rate. Also, the assignment of per-node delay is pre-determined and cannot be changed after the connection is established.

3. Bulk scheduling scheme (BSS)

Rather than adopting fixed rate allocation or per-packet scheduling, we divide the flow load into segments, called “bulk” in this paper, and dynamically allocate bandwidth for each bulk. In the proposed bulk scheduling scheme, Traffic Specification with

End-to-end Deadline (TED) packets are introduced and inserted into flow traffic periodically. With the information contained in TED packets, the BSS scheduler in the intermediate nodes can determine how to schedule the flows, so as to enforce their end-to-end delay bounds without having much impact on other flows. We will prove that the end-to-end delay bounds of all packets can be guaranteed if the end-to-end delay of TED packets is bounded. This means the scheduler only needs to dynamically adjust the bandwidths by means of TED packets, instead of by adjusting each packet.

Assuming that a path of flow i consists of N_{i+1} hops, the ingress node is responsible for inserting TED packets into the flow periodically with the period p_i, and the egress node removes TED packets when they arrive. Meanwhile, the intermediate nodes, which contain ingress and egress nodes, schedule and service packets based on the information in the TED packets and modify the TED contents. Packets in a flow are served in a first-in-first-out (FIFO) manner. Let h_n ($1 \leqslant n \leqslant N_i$) denote the nth intermediate nodes in the path of flow i. Fig. 1 shows an overview of the bulk scheduling scheme, and the notations used in this paper are summarized in Table 1. We define the bulk as follows.

Definition of bulk – Let $\mathrm{TED}_{i,k}$ denote the kth TED packet of flow i. We define the bulk as the packets between two adjacent TED packets and the corresponding TED packet. For example, the kth bulk represents the packets between $\mathrm{TED}_{i,k-1}$ and $\mathrm{TED}_{i,k}$ plus the packet $\mathrm{TED}_{i,k}$.

3.1. Nodal delay assignment problem

The end-to-end delay of a packet is computed here as the period from the time a packet arrives at the ingress node until the time it leaves the egress node because the delays from the sender to the ingress node and from the egress node to the receiver are not controllable by the schedulers in intermediate nodes and are usually constant so they can be ignored in scheduling. Here, we will prove that if the delays of TED packets are guaranteed, then the delay bounds of the packets in the bulks are guaranteed as well.

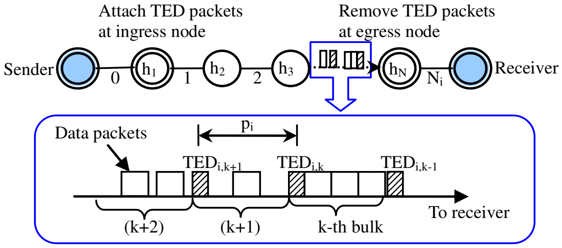

Fig. 1. TED-based bulk scheduling scheme overview.

Table 1
Summary of notations

Symbol	Definition
$A^n_{i,k}(j)$	Arrival time of the jth packet in the kth bulk of flow i at h_n
$A^n_{i,k}(\mathrm{TED}), A^n_{i,k}$	Arrival time of $\mathrm{TED}_{i,k}$ at h_n
$B_{i,k}$	Total data amount of the kth bulk of flow i
$\overline{B}^n_{i,k}$	Total data amount of the kth bulk of flow i when $\mathrm{TED}_{i,k}$ becomes the first TED packet at h_n
$D^n_{i,k}(j)$	Departure time of the jth packet in the kth bulk of flow i at h_n
$D^n_{i,k}(\mathrm{TED}), D^n_{i,k}$	Departure time of $\mathrm{TED}_{i,k}$ at h_n
N_i	Hops count of flow i
$S^n_{i,k}$	Scheduling time of $\mathrm{TED}_{i,k}$ at h_n
$\mathrm{TED}_{i,k}$	kth TED packet of flow i
$d_i^{\max}$	End-to-end delay bound of flow i
$d^n_{i,k}$	Residual end-to-end delay bound of $\mathrm{TED}_{i,k}$ at h_n
h_n	The nth intermediate node
p_i	Period of inserting TED packets into flow i
$r^n_i(t)$	Demand rate of flow i at h_n at time t
$w^n_{i,k}$	Waiting time of $\mathrm{TED}_{i,k}$ at h_n
$\Delta^n_{i,k}$	Processing index of $\mathrm{TED}_{i,k}$ at h_n
$\delta^n_{i,k}$	Processing time of $\mathrm{TED}_{i,k}$ at h_n

Theorem 1. *At the receiver, for the* kth *bulk of flow* i, *if the end-to-end delay of* $\mathrm{TED}_{i,k}$ *is bounded, then the end-to-end delays of the packets in the* kth *bulk are also bounded. Let* $A^1_{i,k}(j)$ *and* $D^{Ni}_{i,k}(j)$ *be the arrival time at the ingress node and the departure time at the egress node of the* jth *packet in the* kth *bulk of flow* i, *respectively. The arrival and departure times of the corresponding* TED *packet are denoted as* $A^1_{i,k}(\mathrm{TED})$ *and* $D^{Ni}_{i,k}(\mathrm{TED})$, *respectively. Then, we get the following relation*:

$$D^{Ni}_{i,k}(\mathrm{TED}) - A^1_{i,k}(\mathrm{TED}) \leqslant d_i^{\max} \rightarrow D^{Ni}_{i,k}(j) - A^1_{i,k}(j) \leqslant d_i^{\max} + p_i, \qquad (2)$$

where $d_i^{\max}$ *is the end-to-end delay bound of flow* i *and* p_i *is the period during which we insert* TED *packets into the flow traffic of flow* i.

By Theorem 1, as long as all the TED packets meet their deadlines, so do all the packets in the flow. Therefore, in the reminder of this paper, we

Please cite this article in press as: Y.-C. Tu et al., Enhanced bulk scheduling for supporting delay ..., Comput. Netw. (2008), doi:10.1016/j.comnet.2007.11.020

will only consider the TED packets in our bulk scheduling scheme and use $A^n_{i,k}$ and $D^n_{i,k}$ to represent $A^n_{i,k}(\mathrm{TED})$ and $D^n_{i,k}(\mathrm{TED})$ if not otherwise specified.

3.2. BSS nodal scheduling in intermediate nodes

In the bulk scheduling scheme, all packets follow per-flow queueing in each node because different flows may have different end-to-end delay bound requirements. The BSS scheduler in the intermediate nodes performs the following tasks:

1. When a TED packet arrives at an intermediate node, its arrival time is recorded for calculation of the remaining time that it can use.
2. When a TED packet becomes the first TED packet in its flow queue, we modify the scheduling priority or service rate of the corresponding flow, according to the information in the TED packet.
3. When the TED packet leaves the intermediate node, we update the information stored in the TED packet.

We classify flows into two groups, T-flows and N-flows, where T-flows are the flows that submit the delay bound requirements via TED packets and N-flows do not. Each flow i has a demand rate r^i_n such that the first TED packet of flow i will not miss its deadline if flow i has a service rate higher than r^i_n at intermediate node h_n. We will propose three different methods for computing r^i_n later. For N-flows, the demand rate is a constant value. Fig. 2 shows the BSS scheduler architecture, where TB and NB are the sets of all backlogged T-flows and N-flows, respectively. The BSS scheduler aims to satisfy the demands of all T-flows in order to enforce their end-to-end delay bound and then allocate the residual bandwidth to N-flows.

In the literature, the general processor sharing (GPS) [22] scheduler is recognized as a scheduling policy that can guarantee the delay bound and achieve the fairest service for each flow. Without loss of generality, in this paper, we assume that a GPS-based scheduling algorithm, such as Weighted Fair Queueing (WFQ) [22], Worst-case Fair Weighted Fair Queueing (WF^{2}Q) [23], or Deficit Round Robin (DRR) [24], is used as the scheduling algorithm in each node. We recommend using the DRR scheduling algorithm because it has low computational complexity and is easy to implement. Table 2 lists the actual bandwidths with the GPS-based scheduler under different conditions. We divide all situations in to three cases. In case (A), the summation of all T-flows' demand rates is more than link capacity, so that we proportionally allocate bandwidth to each T-flows, i.e., $r^n_i \cdot C/\sum_{i\in \mathrm{TB}} r^n_i$ to flow i, and no bandwidth for N-flows. In case (B), the summation of all T-flows' demand rates is less than link capacity and there is no N-flow waiting for service, so that all T-flows can get their service rate higher that its demand. And in case (C), the summation of all T-flows' demand rates is less than link capacity and there are N-flows waiting for service. In this case, we serve T-flows only with their demand rates and the remained bandwidth is proportionally allocated to each N-flows. We can see that it is only in case (A) that the demand rate of T-flows cannot be satisfied but that compared to weighted scheduling bandwidth sharing with other N-flows, the BSS scheduler still provides better service for T-flows. Also, with some form of admission control and

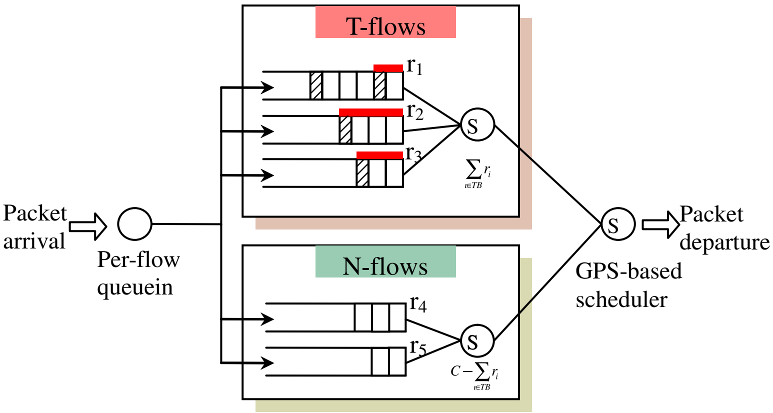

Fig. 2. The scheduler architecture in the intermediate nodes.

Please cite this article in press as: Y.-C. Tu et al., Enhanced bulk scheduling for supporting delay ..., Comput. Netw. (2008), doi:10.1016/j.comnet.2007.11.020

Table 2
The actual bandwidths in intermediate nodes

	Case (A): $\sum_{i\in\mathrm{TB}} r_i^n \geqslant C$	Case (B): $\sum_{i\in\mathrm{TB}} r_i^n < C \wedge \mathrm{NB} = \phi$	Case (C): $\sum_{i\in\mathrm{TB}} r_i^n < C \wedge \mathrm{NB} \neq \phi$
T-flows	$r_i^n \cdot C \Big/ \sum_{i\in\mathrm{TB}} r_i^n$	$r_i^n \cdot C \Big/ \sum_{i\in\mathrm{TB}} r_i^n$	r_i^n
N-flows	0	0	$r_i^n \cdot \left(C - \sum_{i\in\mathrm{TB}} r_i^n\right) \Big/ \sum_{i\in\mathrm{NB}} r_i^n$

bandwidth reservation using a bandwidth reservation protocol, such as RSVP [6] or ST2 [25], the occurrence of case (A) can be controlled. Furthermore, in the case (C), the scheduler only provides the bandwidth that each T-flow demands, so the N-flows can actually get higher service quality than weighted scheduling can provide. Note that we use, but do not limit schedulers to, the GPS-based scheduler for serving all flows. Any other scheduler, such as the priority-based scheduler, can be used in the bulk scheduling scheme.

3.3. Per-bulk service rate estimation

We denote the time instance when $\mathrm{TED}_{i,k}$ becomes the first TED packet in the flow queue of flow i as its scheduling time at h_n. In the remainder of this paper, we will use $A_{i,k}^n$, $S_{i,k}^n$, and $D_{i,k}^n$ to represent the arrival time, schedule time, and departure time of $\mathrm{TED}_{i,k}$ at h_n. Note that we ignore all negligible delays, such as propagation delay, as they do not affect the results. With the above definitions, we get the following equations:

$$A_{i,k}^n = D_{i,k}^{n-1} \quad \text{for } n = 2 \text{ to } N_i, \tag{3}$$

$$S_{i,k}^n = \max(A_{i,k}^n, D_{i,k-1}^n) \quad \text{for } n = 1 \text{ to } N_i. \tag{4}$$

With these terms, the nodal delay time of $\mathrm{TED}_{i,k}$ at h_n is $D_{i,k}^n - A_{i,k}^n$. We divide the nodal delay time of $\mathrm{TED}_{i,k}$ at h_n into two parts – the waiting time $w_{i,k}^n$ and processing time $\delta_{i,k}^n$. The waiting time $w_{i,k}^n$, which is equal to $S_{i,k}^n - A_{i,k}^n$, is the period from the arrival time of $\mathrm{TED}_{i,k}$ to the time when it becomes the first TED packet in its flow queue at h_n. The processing time $\delta_{i,k}^n$, which is equal to $D_{i,k}^n - S_{i,k}^n$, is the period from the time when $\mathrm{TED}_{i,k}$ becomes the first TED packet in its flow queue to the departure time of $\mathrm{TED}_{i,k}$ at h_n. If $\mathrm{TED}_{i,k}$ is the first TED packet in its queue to arrive at h_n, then the waiting time is equal to zero. Therefore, the composition of the end-to-end delay of $\mathrm{TED}_{i,k}$ consists of the waiting and processing times at all nodes that $\mathrm{TED}_{i,k}$ passes.

The objective of our scheduler in the intermediate nodes is to minimize the impact of flow i on other flows that pass through this node, i.e., h_n, under the constraint that all TED packets can meet their end-to-end delay bounds, i.e.,

$$\sum_{n=1}^{Ni} \left(w_{i,k}^n + \delta_{i,k}^n\right) \leqslant d_i^{\max} \quad \text{for all } k. \tag{5}$$

In the end-to-end delay, the waiting time $w_{i,k}^n$ is not changeable after $\mathrm{TED}_{i,k}$ becomes the first TED packet, so the scheduler can only control the processing time of $\mathrm{TED}_{i,k}$. However, there is a tradeoff between minimizing the impact on other flows and enforcing end-to-end delay bounds for flow i. A short processing time $\delta_{i,k}^n$ of $\mathrm{TED}_{i,k}$ at h_n shortens its end-to-end delay but reduces the bandwidths assigned to other flows and incurs longer delays. Therefore, the end-to-end delay bound assignment for TED packets and controlling the processing times for TED packets at intermediate nodes are the major issues with the bulk scheduling scheme.

In order to serve $\mathrm{TED}_{i,k}$ at h_n with a proper processing time, we denote $\Delta_{i,k}^n$ as the *processing index* assigned to flow i at h_n when $\mathrm{TED}_{i,k}$ becomes the first TED packet at h_n. The BSS scheduler determines the service rate of each flow by means of its processing index. A lower $\Delta_{i,k}^n$ of flow i means that it requires a higher bandwidth to be served at h_n. In addition, the residual end-to-end delay bound when $\mathrm{TED}_{i,k}$ arrives at h_n is represented by $d_{i,k}^n$ and defined as follows:

$$\begin{aligned} d_{i,k}^n &= d_i^{\max} - (A_{i,k}^n - A_{i,k}^1) \\ &= d_i^{\max} - \sum_{m=1}^{n-1} \left(w_{i,k}^m + \delta_{i,k}^m\right). \end{aligned} \tag{6}$$

We assign the value of $\Delta_{i,k}^n$ based on the expected processing time at h_n. We hope that if the actual nodal processing time $\delta_{i,k}^n$ is equal to $\Delta_{i,k}^n$ at each node in the path of flow i, then the end-to-end delay of $\mathrm{TED}_{i,k}$ is exactly equal to $d_i^{\max}$. Note that we use the residual end-to-end delay bounds rather than the absolute global deadlines of TED packets for scheduling, so there is no time synchronization problem among all the nodes. In the following, we will propose three methods for estimating the nodal processing indexes of $\text{TED}_{i,k}$ at its nth intermediate nodes.

3.3.1. Global even distribution (GED)

The first method is based on the scheme proposed in [17] and the nodal delay assignment in [21]. The end-to-end delay bound is evenly allocated to all nodes in the path. Each node gets its initial nodal delay bound when the connection is established. $\Delta^n_{i,k}$ is estimated as follows:

$$\Delta^n_{i,k} = d^n_{i,k} - (N_i - n) \cdot d^{\max}_i / N_i - w^n_{i,k}. \tag{7}$$

3.3.2. Fair allocation among residual on arrival (FAR-A)

In the second method, we calculate the processing index $\Delta^n_{i,k}$ based on the residual end-to-end delay bound when $\text{TED}_{i,k}$ arrives at h_n. We follow the guideline that the residual end-to-end delay bound is evenly allocated among residual nodes. Therefore, the expected nodal delay bound of $\text{TED}_{i,k}$ at h_n is $d^n_{i,k}/(N_i - n + 1)$, and $\Delta^n_{i,k}$ is

$$\Delta^n_{i,k} = d^n_{i,k}/(N_i - n + 1) - w^n_{i,k}. \tag{8}$$

3.3.3. Fair allocation among residual on schedule (FAR-S)

In this method, we also allocate the residual end-to-end delay bound evenly among the residual nodes. Unlike FAR-A we take the waiting time of $\text{TED}_{i,k}$ in h_n into account when we determine the residual nodal delay bound in this method. Note that the actual residual end-to-end delay bound when $\text{TED}_{i,k}$ becomes the leading TED packet, i.e., the schedule time of $\text{TED}_{i,k}$ at h_n, is $d^n_{i,k} - w^n_{i,k}$. Since we want to allocate the residual end-to-end delay bound evenly among all residual nodes, $\Delta^n_{i,k}$ is calculated as follows:

$$\Delta^n_{i,k} = (d^n_{i,k} - w^n_{i,k})/(N_i - n + 1). \tag{9}$$

In the three methods described above, the residual end-to-end delay time $d^n_{i,k}$ and residual hop count $N_i - n + 1$ are stored in TED packets, while the waiting time $w^n_{i,k}$ is measured by the intermediate node h_n. The estimations are based one the assumption that $w^n_{i,k} < d^n_{i,k}$ because $\text{TED}_{i,k}$ has missed its end-to-end deadline if $w^n_{i,k} \geqslant d^n_{i,k}$. If the nodal processing time is guaranteed to be less than the processing index, then $\text{TED}_{i,k}$ will definitely meet its end-to-end deadline since $w^n_{i,k} + \Delta^n_{i,k}$ is less than $d^n_{i,k}$ with all three methods at h_n.

With the processing index $\Delta^n_{i,k}$, we calculate the demand rate of flow i as follows:

$$r_i(t) = \overline{B}^n_{i,k}/\Delta^n_{i,k} \quad \text{for } S^n_{i,k+1} > t \geqslant S^n_{i,k}, \tag{10}$$

where $\overline{B}^n_{i,k}$ is the amount of data that queues before $\text{TED}_{i,k}$ plus the size of $\text{TED}_{i,k}$ when $\text{TED}_{i,k}$ becomes the leading TED packet in node h_n. In the case of $w^n_{i,k} \geqslant d^n_{i,k}$, the processing index is less than or equal to zero, which means that $\text{TED}_{i,k}$ cannot possibly meet its end-to-end deadline even we allocate all of the link capacity to flow i, so we can drop this bulk or allocate a default bandwidth for flow i.

3.4. TED period and end-to-end delay bound assignment

According to Theorem 1, the end-to-end delay bound guarantee for all packets is $d^{\max}_i + p_i$ if the end-to-end delay bound guarantee of TED packets of flow i is $d^{\max}_i$. In other words, the end-to-end delay bound guarantee for TED packets is not equivalent to that for flow i. We will now present two strategies for assigning end-to-end delay bound to TED packets.

3.4.1. Explicit end-to-end delay bound assignment

The explicit assignment sets the end-to-end delay bound for TED packets with the value $d^{\max}_i - p_i$. Hence, we can guarantee the end-to-end delay bound for all packets in flow i with $d^{\max}_i$. With the explicit assignment strategy, a large TED period p_i forces the intermediate nodes to schedule TED packets with small end-to-end delay bounds, requiring high bandwidth. On the other hand, a small TED period will result in excessive TED packets.

3.4.2. Implicit end-to-end delay bound assignment

Although explicit assignment provides a tight delay bound for all packets, it may result in a high bandwidth requirement and overhead for intermediate nodes. Compared to explicit assignment, implicit assignment provides a relaxed end-to-end delay bound for all packets by setting the end-to-end delay bound for TED packets exactly equal to $d^{\max}_i$. We will show that with our proposed bulk scheduling algorithms, we can enforce end-to-end delay bounds of $d^{\max}_i$ for all packets in the kth bulk by keeping the period of TED packets shorter than or equal to the processing index of $\text{TED}_{i,k}$ at its egress node, i.e., $\Delta^{Ni}_{i,k} \geqslant p_i$, if the service rates of each bulk of flow i at all intermediate nodes are stable.

Theorem 2. *Suppose the sending rate of the* kth *bulk of flow i at the sender and the service rate of flow i while* $\mathrm{TED}_{i,k}$ *is being processed at* h_{Ni} *are constant, if the processing index of* $\mathrm{TED}_{i,k}$ *at* h_{Ni} *is more than or equal to the TED period of flow i, i.e.,* $\Delta_{i,k}^{Ni} \geqslant p_i$, *and if its processing time is not longer than its processing index, then the end-to-end delay of each packet in the* kth *bulk will be shorter than or equal to the end-to-end delay bound of* $\mathrm{TED}_{i,k}$, *i.e.,* $d_i^{\max}$.

Note that the value of processing index $\Delta_{i,k}^{Ni}$ must be less than or equal to $d_{i,k}^{Ni} - w_{i,k}^{Ni}$. Therefore, this assignment is not suitable for the heavy loaded network environment or a flow with many hops because packets may have little residual end-to-end delay bound or long waiting time at the egress node, which makes the value of $d_{i,k}^{Ni} - w_{i,k}^{Ni}$ less than p_i.

3.5. The implementation of the bulk scheduling scheme

To implement the bulk scheduling scheme in a network, it is necessary to add some components in existing ingress, egress and intermediate nodes. We show the prototype of the implementation of the bulk scheduling scheme in Fig. 3. We add the TED adders and removers in existing ingress and egress nodes, respectively. In each intermediate node, the TED interpreter and modifier are embedded to the existing intermediate nodes to get information from TED packets, adjust the service rates of priorities of all flows, and modify the contain of TED packets for other intermediate nodes.

4. Enhancements of the bulk scheduling scheme

In the bulk scheduling scheme, the scheduler in the intermediate nodes dynamically allocates bandwidth for each bulk to enforce its delay bound. However, TED packets will still miss their deadlines because of the limited bandwidth resources available to all flows and due to improper bandwidth allocation. Here, we propose two enhancements of the bulk scheduling scheme to alleviate this problem: a drop missed policy for late packets and a feedback mechanism to refine bandwidth allocation.

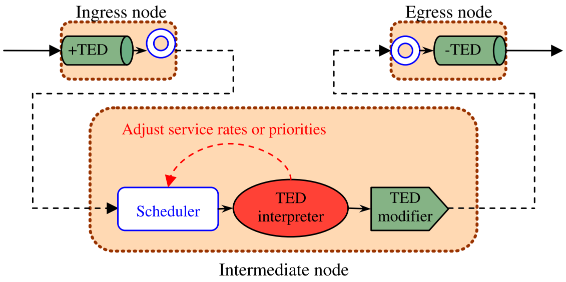

Fig. 3. The prototype implementation of the BSS.

4.1. Drop missed policy

The residual end-to-end delay bound information in TED packets can also be used in the dropping policy, which determines whether or not that a packet should be dropped at the intermediate nodes. In Theorem 3 below, we will prove that if we detect that a TED packet has missed its end-to-end deadline, then all the packets in the same flow and are queueing in front of the TED packet will also miss their end-to-end deadlines. Therefore, we can drop these packets that have missed their end-to-end deadlines and serve other packets in the node to improve utilization of the node.

Theorem 3. *Suppose that the packet sequence is not disordered when the packets of each flow pass each node and that the* TED *packets and data packets in flow i have the same end-to-end delay bound* $d_i^{\max}$. *At any time t, if a* TED *packet* $\mathrm{TED}_{i,k}$ *misses its end-to-end deadline at* h_n, *then all the packets that are in the same flow and are queueing in front of* $\mathrm{TED}_{i,k}$ *in* h_n *will also miss their end-to-end deadlines.*

Since a TED packet only records the residual end-to-end delay bound when it leaves the previous node rather than the absolute end-to-end deadline, we must calculate the actual residual end-to-end delay bound when it is scheduled, i.e., the recorded residual end-to-end delay bounds in TED packets minus the waiting time of the TED packets. According to Theorem 3, if the TED packet misses its end-to-end deadline and the drop missed policy is applied, we drop the packets that are queueing in front of the TED packet. As a result, we can allocate bandwidth to those packets that can still arrive at their destinations in time.

4.2. Feedback mechanism

When a TED packet misses its end-to-end deadline, besides the drop missed policy, we propose using a feedback mechanism to let the ingress nodes add some extra information to the TED packets to inform the intermediate nodes to speedup for those

TED packets that have a tendency to miss their end-to-end deadlines. In the proposed bulk scheduling scheme, the processing time $\delta^n_{i,k}$ is associated with our method for scheduling TED$_{i,k}$ at h_n. However, the waiting time is determined by the departure time of the previous TED packet in the same flow at h_n. This means that the end-to-end delay of TED$_{i,k}$ is associated not only with the scheduling of TED$_{i,k}$ but also with the delay conditions of previous TED packets in the same flow. Hence, we get the following property.

Property 1. *If the waiting time of* TED$_{i,k}$ *at* h_n *is more than zero, then the cumulated delay when* TED$_{i,k}$ *leaves node* h_n *is independent of the delay time that* TED$_{i,k}$ *has experienced in its previous journey, i.e., from* h_1 *to* h_{n-1}*. Also, the cumulated delay, i.e.,* $D^n_{i,k} - A^1_{i,k}$*, can be represented as follows:*

$$D^n_{i,k} - A^1_{i,k} = \left(D^n_{i,k-1} - A^1_{i,k-1}\right) + \left(\delta^n_{i,k} - p_i\right). \tag{11}$$

According to Property 1, we say that TED$_{i,k}$ has a deferment at h_n if the waiting time of TED$_{i,k}$, i.e., $w^n_{i,k}$, is more than zero. And the last-deferment hop of TED$_{i,k}$ represents the last hop for which TED$_{i,k}$ has a deferment in the path of flow i.

Property 1 shows that the cumulated delay of TED$_{i,k-1}$ will propagate to the next TED packet TED$_{i,k}$ if the waiting time of TED$_{i,k}$ is more than zero. Furthermore, any nodal delay time reductions before the last-deferment hop of TED$_{i,k}$ is of no use if the departure time of TED$_{i,k-1}$ at the last-deferment hop is unchanged. Let TED$_{i,k'}$, where $k' < k$, be the TED packet that has no deferment at h_n and the waiting times of TED$_{i,q}$ ($k' < q \leqslant k$) are all more than zero. With Property 1, the cumulated delay of TED$_{i,k}$ at h_n can be represented as follows:

$$\begin{aligned} D^n_{i,k} - A^1_{i,k} &= \left(D^n_{i,k'} - A^n_{i,k'}\right) + \sum_{k'<q\leqslant k} \delta^n_{i,q} - (k-k')\cdot p_i \\ &= A^n_{i,k'} - A^1_{i,k} + \sum_{k'\leqslant q\leqslant k} \delta^n_{i,q}. \end{aligned} \tag{12}$$

Theorem 4. *Suppose the nodal processing times of* TED$_{i,q}$*, where* $k' \leqslant q \leqslant k$*, at* $h_{n'}$ *are all reduced by* $\bar{\delta}$ *and the nodal delay times at other intermediate nodes are unchanged. Performing the reduction on the last-deferment hop reduces the end-to-end delay of* TED$_{i,k}$ *the most.*

The objective of feedback mechanism is to increase the bandwidths at some nodes for flows that may miss their end-to-end deadlines so that the missed deadline condition can be reduced or eliminated following refinement. Therefore, when a TED$_{i,k}$ arrives at its egress node with a residual end-to-end delay bound that is less than zero, the egress node will send an acknowledge to the ingress node to request a speedup for the flow i. Note that the acknowledges are delivered by special control packets, which denoted as ACK packets, in our bulk scheduling scheme. According to Theorem 4, we can remove the deferment condition at the last-deferment hop in the speedup and design the feedback mechanism for the bulk scheduling scheme as follows:

1. When a TED packet TED$_{i,k}$ has a deferment at hop h_n, the intermediate node h_n records the residual hops count and the expected weight, which is calculated as follows:

   $$\text{expected weight} = \sum_{k'\leqslant q<k} \delta^n_{i,q} \Bigg/ \left(\sum_{k'\leqslant q<k} \delta^n_{i,q} - w^n_{i,q}\right). \tag{13}$$

   Note that we only need to record the last-deferment hop, so the TED packet size will not be increased by the number of deferment increases.
2. When the egress node receives TED$_{i,k}$ with a residual end-to-end delay bound that is less than zero, it sends an ACK, which contains the residual hop count of the last-deferment hop and the expected weight, to the ingress node of flow i.
3. When the ingress node receives the ACK, it stores the information in later TED packets to inform h_n that the bandwidth of flow i must be increased to the expected weight times of the original estimation. In the following, we will show that the deferment at h_n can be eliminated following this speedup.

With this feedback mechanism, once the same traffic pattern appears, the nodal processing times following speedup at the last-deferment hop h_n becomes

$$\hat{\delta}^n_{i,q} = \delta^n_{i,q} \cdot \left(\sum_{k'\leqslant q<k} \delta^n_{i,q} - w^n_{i,q}\right) \Bigg/ \sum_{k'\leqslant q<k} \delta^n_{i,q} \tag{14}$$

and the waiting time of TED$_{i,k}$ at h_n becomes

$$\begin{aligned} \hat{w}^n_{i,k} &= A^n_{i,k'} + \sum_{k'\leqslant q<k} \hat{\delta}^n_{i,q} - A^n_{i,k} \\ &= A^n_{i,k'} + \sum_{k'\leqslant q<k} \delta^n_{i,q} - w^n_{i,k} - A^n_{i,k} = 0. \end{aligned} \tag{15}$$

Please cite this article in press as: Y.-C. Tu et al., Enhanced bulk scheduling for supporting delay ..., Comput. Netw. (2008), doi:10.1016/j.comnet.2007.11.020

In other words, the deferment at h_n no longer exists. It is expected that a new last-deferment hop may occur following the speedup. If this happens, the feedback mechanism and speedup procedure will be applied again, in which case the TED packets will have to store more than one speedup message. An n-stage feedback mechanism is one that can carry n speedup messages in TED packets and resolve the last n deferment hops in its flow path.

5. Performance evaluation

In this section, we will evaluate the missed deadline ratio and end-to-end delay performance of the proposed bulk scheduling scheme based on simulations results. All of the simulations were performed using the ns-2 network simulator [26] with minor modifications. We show the benefit of the proposed bulk scheduling scheme by comparing with traditional fixed bandwidth allocation schemes. All flows in our simulations are divided into two groups: T-flows and N-flows. T-flows are the flows that have end-to-end delay requirements and are scheduled with special behaviors, such as being reserved with some bandwidth in traditional bandwidth reservation scheme or being schedule with TED packets in bulk scheduling scheme; and others are N-flows.

Each T-flow has an end-to-end delay bound, which means the pre-buffered data in client will be exhausted if a packet's end-to-end delay is longer than this bound. Since a delay constrained streaming application only concerns whether the packets arrives destination in time, we use the missed deadline ratio, which is defined as the number of missed deadline packets (including dropped packets) divided by the total number of offered packets, to be the quality index for T-flows. Besides, we show the average end-to-end delay of N-flows as the indication of the scheduler impacts on N-flows because the end-to-end delay bound is not guaranteed for N-flows.

We ignore all uncontrollable and negligible delay such as link propagation delay in our simulation. If not specified, the default traffic source of each flow follows the exponential ON/OFF model with mean burst/idle period 5000/5000 ms and sending rate during burst period 512 Kbps to emulate the behavior of video-conferencing. All flows use the UDP as their transport protocol. The period of each simulation is 3000 s. The default value of p_i for each flow is 100 ms and explicit end-to-end delay bound assignment for TED packets is applied.

5.1. Single-hop case

In this set of single-hop simulations, we consider the network topology as shown in Fig. 4, where five flows share a single link. Each flow i is with the sender S_i and receiver R_i. Three T-flows, Flow 1, 2 and 3, have end-to-end delay bounds of 500 ms, 1000 ms and 1500 ms, respectively; while the other two flows, Flow 4 and 5, are N-flows. All packets of the five flows pass through the single link, where the link capacity is 1500 bps. We perform three levels of bandwidth reservation schemes-reservation with peak rates, 0.8 peak rates and mean rates of the flows. The results are shown in Table 3, in which we can see that the BSS provides the best quality for T-flows like the peak rate reservation but has shorter delays for N-flows. Then, we increased the delay bounds for Flow 1, 2 and 3 to be 2000 ms, 2500 ms and 3000 ms and the results show that the BSS can provides even shorter delays for N-flows, which means the BSS has high flexibility. Note that the BSS with implicit end-to-end delay bound assignment has worse qualities than explicit end-to-end delay bound assignment for T-flows because the traffic load of each flow is variable-bit-rate and the congested may occur sometimes that the source rate and service rate cannot always be constant.

5.1.1. Performance with different TED period lengths

In this section, we examine the performance of bulk scheduling scheme with different TED periods. Since our BSS insert TED packets into each flow periodically, a short TED period may result in high overhead on traffic load. However, a long TED means the BSS has less control on dynamic bandwidth allocations, which causes a worse performance on QoS provision. In Fig. 5, we show the results of different end-to-end delay bound assignments under different TED period. We can see that the explicit end-to-end delay bound can always provide full quality for T-flows under different TED periods, but the implicit assignment provides lower quality for T-flows with longer TED period because it can

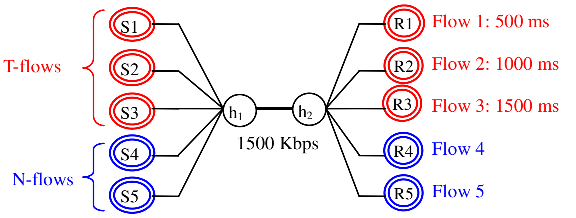

Fig. 4. The network topology in the single-hop case simulation.

Table 3
The simulation results in single-hop case

	Resv. (Peak)	Resv. (0.8 * Peak)	Resv. (Mean)	BSS (implicit)	BSS (explicit)
Delay bound for Flow 1/2/3: 500/1000/1500 ms					
Flow 1	0%	13.63%	42.72%	0.18%	0%
Flow 2	0%	5.87%	28.19%	0.38%	0%
Flow 3	0%	0.74%	22.21%	0.07%	0%
N-flows	2079 ms	1889 ms	1107 ms	1575 ms	1617 ms
Delay bound for Flow 1/2/3: 2000/2500/3000 ms					
Flow 1	0%	0.27%	18.28%	0.01%	0%
Flow 2	0%	0.18%	12.24%	0.09%	0%
Flow 3	0%	0%	9.53%	0.03%	0%
N-flows	2079 ms	1889 ms	1107 ms	1016 ms	1043 ms

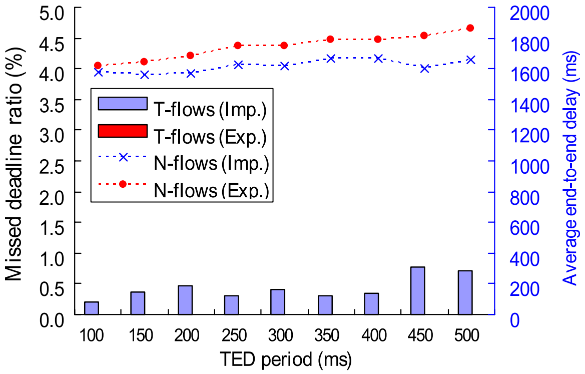

Fig. 5. The performances of the BSS with different delay bound assignments under different TED period.

only guarantee the end-to-end delay bound with $d_i^{\max} + p_i$ as shown in Theorem 1. Besides, a longer TED period causes longer delays for N-flows especially with the explicit end-to-end delay bound assignment. In our simulations, the TED packet size, which contains the IP header and other information, is 50 bytes. Note that the BSS only need the information of residual end-to-end delay bound for scheduling, so that the size of TED packets can be only 32 bytes. But we reserved more 18 bytes space in TED packets for future developments. These TED packets bring the extra traffic loads of 4 Kbps and 800 bps to each flow when TED periods are 100 ms and 500 ms. The overhead is small in our simulation, so the results cannot show the effect of the extra traffic load of TED packets. If the network is very congested or the data rate of each flow is low, e.g. less than 10 Kbps, a longer TED period would be chosen.

5.1.2. Bulk scheduling scheme in a lossy network environment

In a lossy network, TED packets may be lost during transmission. In this simulation, we set the packet lost rate between 0% and 10% to examine the performance of the bulk scheduling scheme when packets may be lost. The results are shown in Fig. 6. One can see that the lost packets have little impact on our bulk scheduling scheme since the bulk scheduler maintained the bandwidth of the previous bulk when a TED packet was lost.

5.2. Multi-hop case

In the following, we evaluate the performance of bulk scheduling scheme on flows with multi-hops to understand its behaviors in a complicated network environment. Fig. 7 shows the network topology used in this simulation, where the link capacity of each link is 1500 Kbps. There are 10 T-flows (one

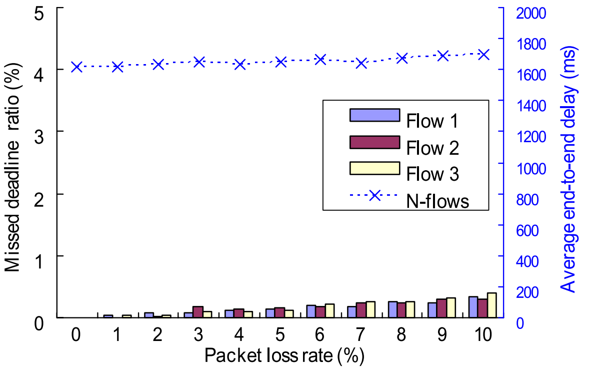

Fig. 6. The performances of BSS with explicit TED delay bound assignment under different packet lost rates.

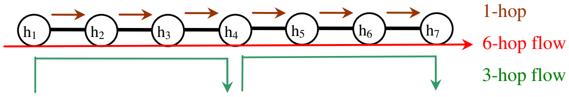

Fig. 7. The network topology for the multi-hop simulation.

Please cite this article in press as: Y.-C. Tu et al., Enhanced bulk scheduling for supporting delay ..., Comput. Netw. (2008), doi:10.1016/j.comnet.2007.11.020

Table 4
The simulation results of non-bulk and bulk scheduling in the multi-hop case

Without BSS	Resv. (Peak)	Resv. (0.8 * Peak)	Resv. (Mean)
6-Hops	0%	45.08%	81.11%
3-Hops	0%	30.02%	67.49%
1-Hop	0%	14.31%	37.06%
N-flow	3509 ms	2954 ms	1423 ms

With BSS (Explicit)	None	GED	FAR-A	FAR-S
6-Hops	25.45%	2.28%	2.06%	2.68%
3-Hops	12.68%	1.91%	1.65%	1.61%
1-Hop	3.45%	1.64%	1.17%	0.51%
N-flow	3043 ms	3108 ms	3118 ms	3094 ms

for 6-hops, two for 3-hops and six for 1-hop) and other twelve 1-hop N-flows, two for each link. Table 4 lists the results of the end-to-end delay bound of each T-flows being 1000 ms. Besides three traditional bandwidth reservation schemes, the simulation results of the bulk scheduling scheme with different nodal processing index assignments are also listed in Table 4, where the algorithm "none" means that all intermediate nodes schedule a packet with it own end-to-end deadline, i.e., the deadline is globally seen for all nodes, and the nodal processing index of $\mathrm{TED}_{i,k}$ at h_n is equal to its residual end-to-end delay bound $d^n_{i,k}$.

Within the table, we can see that reservation with peak rate scheme has 0% missed deadline ratio for T-flows but result in long delays for N-flows and T-flows have high missed deadline ratios both on reservation with 0.8 * peak and mean rates. With our bulk scheduling scheme, T-flows have better performances than the result of reservation with 0.8 * peak or mean rate.

The bulk scheduling scheme cannot achieve 100% quality on T-flows because the nodal processing index assignments are based on the prediction of congestions on residual hops, which is difficult to be completely accurate. But we still can see that the proposed three algorithms, i.e., GED, FAR-A and FAR-S, have almost zero missed deadline ratios on T-flows and much better performances than the BSS without any processing index assignment algorithm.

With our proposed three processing index assignments, the bulk scheduling scheme performs well both on T-flows and N-flows. The FAR-S has least influence on others flows when the 6-hops T-flow pass through each node so the 3-hops, 1-hop T-flows and N-flows have the better performances compared to FAR-A and GED. However, this causes the flows with longer paths to have worse qualities. Compared to FAR-S, the algorithms GED and FAR-A provide better fairness to the flows with different path lengths, but will result in worse overall network utilization. We show the results with the three algorithms under different end-to-end delay bounds for T-flows in Fig. 8.

5.3. Performance improvement resulting from the drop missed policy and feedback mechanism

In the previous simulations, we did not apply any enhancements, because the enhancements have little effect when most packets of T-flows have already met their end-to-end deadlines. In order to show the benefit of the improvements, we reserve 50 Kbps for N-flows in each link so that the T-flows will have high missed deadline ratios. In the following simulation results, we focus on the improvement on 6-hops flow with FAR-S nodal delay bound assignment. Fig. 9 shows the results of bulk scheduling with the feedback mechanism and drop missed policy. The 1-stage feedback mechanism can reduce the missed deadline ratios of T-flows about 5%. With multi-stage feedback, such as the 3-stage feedback in this figure, we can have even lower missed deadline ratios than the 1-stage feedback. And combining the drop missed policy the bulk scheduling scheme can even more improve the quality of T-flows. Note that both the dropping policy and feedback mechanism are triggered by packets missing end-to-end deadline. Therefore, the improvements are little when the miss deadline ratios of all flows are low. In our design, the ACK packets only contain the information of last-deferment hop and expected weight so the ACK packet size is less than 30 bytes. Besides, the ACK packets are produced only when data packets have missed their end-to-end deadline, so the overhead of ACK packets is very small.

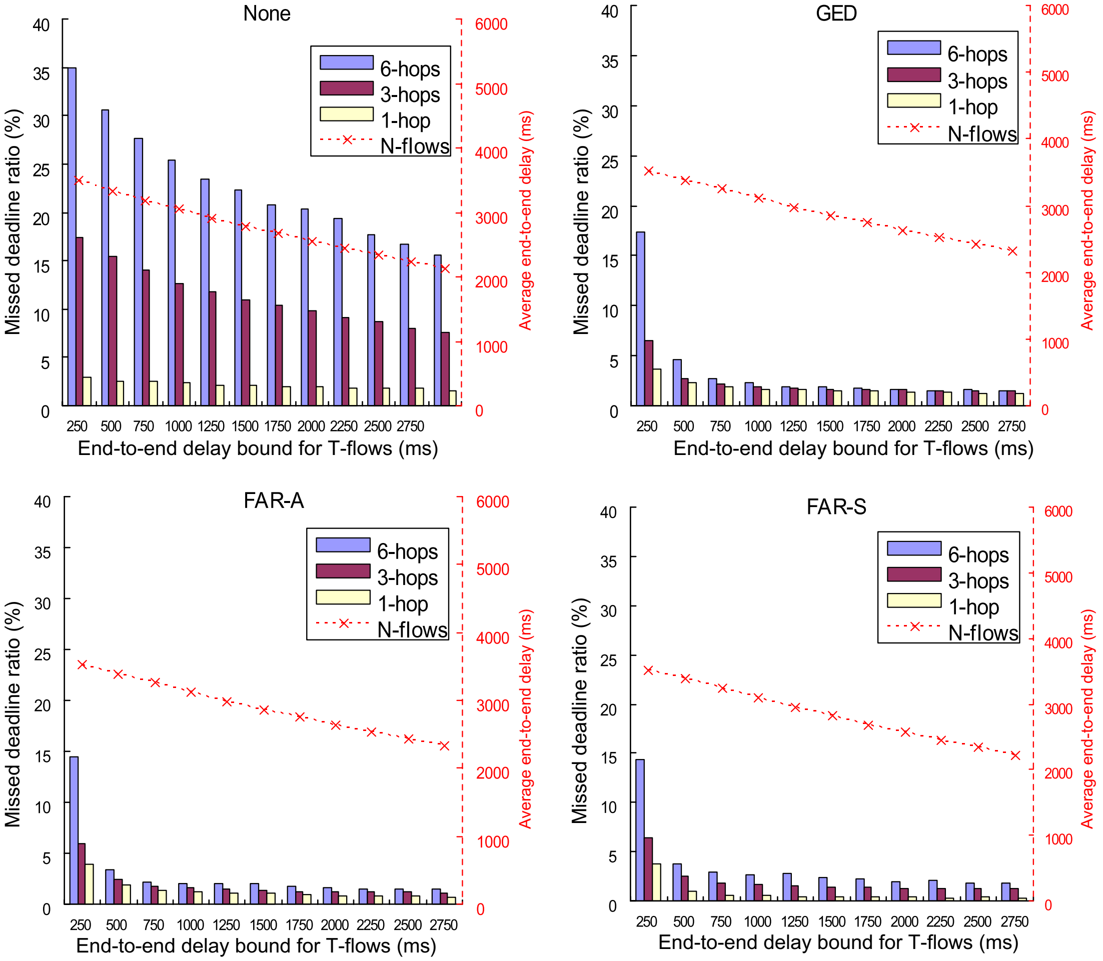

Fig. 8. The performance of bulk scheduling with GED, FAR-A and FAR-S under different end-to-end delay bounds.

5.4. Simulation with video traffic

In order to examine the performance of BSS for real variable-bit-rate traffics, we take ten traffic sources, driven from traces of H.263 encoded video from different films [27] in the following simulations to emulate the real-time streaming flows. We consider the six-hop topology as shown in Fig. 7, where the link capacity for each link is 1500 Kbps. There are two T-flows carrying with a video traffic and an ON/OFF model traffic, respectively from h_1 to h_6, while each link in the path has a best-effort N-flow as the background traffics. The T-flow with ON/OFF model traffic follows the exponential ON/OFF model with mean burst/idle period 5000/5000 ms and sending rate 512 Kbps during burst period.

Fig. 10 shows the missed deadline ratios of the T-flows with video traffic and ON/OFF model traffic when the video traffic is selected from the 10 films and the end-to-end delay bounds for T-flows are 1000 ms. We can see that both reservations with mean rate and double of mean rate perform bad

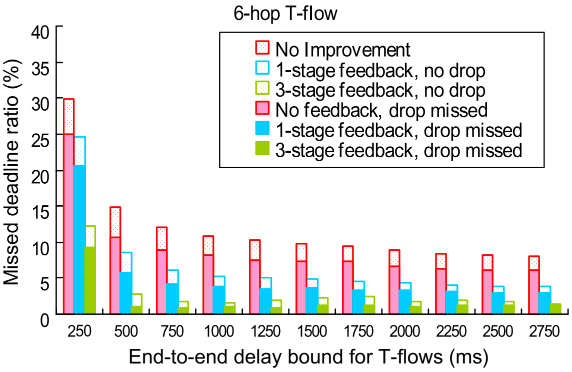

Fig. 9. The performance of bulk scheduling with the feedback mechanism and the drop missed policy.

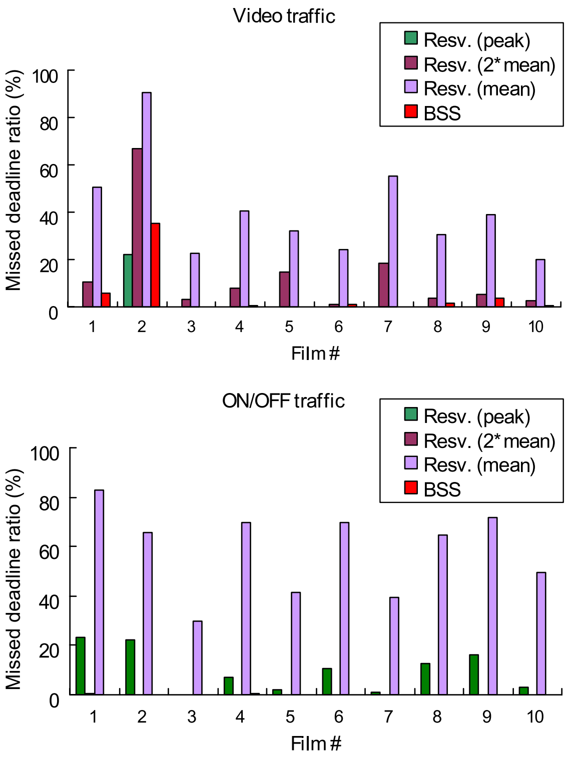

Fig. 10. The missed deadline ratio of the T-flows in the simulation with video traffics.

quality for the real-time video streaming because the variances of these videos are large. Our BSS provides much higher quality, i.e., less missed deadline ratio, as the reservation with peak rate. Note that the film 2 has high missed deadline ratio whichever reservation scheme we used. This is because the bandwidth requirement of the film is often more than link capacity, so that no one can serve this film with missed deadline ratio less than 20%. As shown in Fig. 10, although the reservation with peak rate provides high quality for the T-flow with video traffic, the missed deadline ratio of the T-flow with ON/OFF model traffic is higher than the BSS and reservation with double of mean rate. From these results, we have an observation that a flow with high variant traffic, such as the video traffic in our simulation, prefers the reservation with peak rate but results in the reduction of transmission quality on other flows. Thus, any fixed bandwidth reservation scheme must have tradeoff between flows with high and low variant traffics. But with our bulk scheduling scheme, each node dynamically adjusts the bandwidth reservation for each flow, so it can provide high qualities for all flows.

6. Conclusion and discussion

To summarize, the bulk scheduling scheme with three different nodal processing index estimation methods provides a novel framework that can theoretically guarantee end-to-end delay bounds for real-time flows with variable-bit-rate traffic. We have proved that if the end-to-end delay of TED packets is guaranteed, then the end-to-end delay of all packets is also guaranteed. In the proposed design, a TED packet only need to store the information of the values of the residual end-to-end delay bound, such that the size of a TED packet can be less than 50 bytes. Formally, extra load incurred by TED packets is the size of TED packets divided by the period of inserting TED packets, i.e., p_i, for each flow. Therefore, if we insert TED packets every 50 ms, the overhead is about 1 Kbps, which is very affordable on the Internet.

To implement the bulk scheduling scheme, three components are necessary to be integrated into routers: they are (1) the TED packets insertions in ingress routers, (2) the TED packet eliminations in egress routers, and (3) BSS schedulers in intermediate routers. Note that even if some routers do not apply our proposal bulk scheduling, the network still can work correctly despite the flows without inserting TED packets will be treated as N-flows and the routers without BSS schedulers should provide certain nodal delay bounds. If a router in the path cannot provide the guarantee of nodal delay bound, the BSS still can work by estimating the nodal delay time in the router that does not support our BSS. For example, suppose a flow pass throughput routers A, B, and C in sequence and only routers A and C support BSS, when a TED packet arrive at router C, the scheduler in router C can estimate the delay time in router B and subtract the delay time from residual end-to-end delay recorded in the TED packet. Therefore, the end-to-end delay bound still can be guaranteed after router C. The estimation of delay time can be exactly measured by router C if the clocks in routers A and C are synchronized. Otherwise, we can have a looser bound of delay time in router B according to the arrival times of consequent TED packets.

We have performed extensive simulations to fully examine the characteristics of the bulk scheduling scheme in various transmission environments. The results show that the bulk scheduling scheme can generally provide better end-to-end delay guarantees than rate-based scheduling algorithms can. In addition, it is difficult to assign proper weights to the flows in a rate-based system.

In this paper, we only apply our BSS with Intserv networks, therefore, we must have per-flow queueing to control the bandwidth for each flows. In the future, we will try to apply the BSS to Diffserv or MPLS networks. We believe the BSS can perform well in these networks with the assistance of TED packets in our BSS and dynamically adjusting the priorities or labels of packets.

Appendix

Proof of Theorem 1. Since TED packets are generated periodically with a period p_i and the packets of the same flow at each node are served in a FIFO manner, the arrival time of any packet j in the kth bulk at the ingress node, i.e., $A^1_{i,k}(j)$, is later than $A^1_{i,k}(\mathrm{TED}) - p_i$, and the departure time $D^{Ni}_{i,k}(j)$ is earlier than $D^{Ni}_{i,k}(\mathrm{TED})$. We get the following relation:

$$D^{Ni}_{i,k}(j) - A^1_{i,k}(j) \leqslant D^{Ni}_{i,k}(\mathrm{TED}) - \left(A^1_{i,k}(\mathrm{TED}) - p_i\right). \tag{16}$$

And with the end-to-end delay bound of the TED packets $d_i^{\max}$, we can obtain

$$D^{Ni}_{i,k}(j) - A^1_{i,k}(j) \leqslant d_i^{\max} + p_i. \quad \square \tag{17}$$

Proof of Theorem 2. We assume that the processing time of $\mathrm{TED}_{i,k}$ at h_{Ni} is exactly equal to its processing index. If all of the packets in the kth bulk can meet their end-to-end deadlines in this case, then they will also meet their deadlines because they have shorter end-to-end delays if $\delta^{Ni}_{i,k} < \Delta^{Ni}_{i,k}$.

Let $B_{i,k}$ be the total data amount in the kth bulk of flow i, and let $B_{i,k}(j)$ be the amount of data after the jth packet in the kth bulk flow i plus the size of $\mathrm{TED}_{i,k}$. Since the sending rate is constant, the relation between the arrival time of the jth packet and $\mathrm{TED}_{i,k}$ at the ingress node is as follows:

$$A^1_{i,k}(j) = A^1_{i,k}(\mathrm{TED}) - p_i \cdot B_{i,k}(j)/B_{i,k}. \tag{18}$$

If the jth packet leaves h_{Ni} before $\mathrm{TED}_{i,k}$ arrives at h_{Ni}, we get

$$D^{Ni}_{i,k}(j) \leqslant D^{Ni}_{i,k}(\mathrm{TED}) - \delta^{Ni}_{i,k} \leqslant D^{Ni}_{i,k}(\mathrm{TED}) - p_i. \tag{19}$$

Since $B_{i,k}(j)$ must be less than $B_{i,k}$, combining (18) and (19), we get

$$\begin{aligned} D^{Ni}_{i,k}(j) - A^{Ni}_{i,k}(j) &\leqslant D^{Ni}_{i,k}(\mathrm{TED}) - A^1_{i,k}(\mathrm{TED}) \\ &\quad - p_i \cdot (1 - B_{i,k}(j)/B_{i,k}) \\ &< D^{Ni}_{i,k}(\mathrm{TED}) - A^1_{i,k}(\mathrm{TED}) \\ &\leqslant d_i^{\max}. \end{aligned} \tag{20}$$

Otherwise, let $\overline{B}^{Ni}_{i,k}$ be the amount of data that queues before $\mathrm{TED}_{i,k}$ plus the size of $\mathrm{TED}_{i,k}$ when $\mathrm{TED}_{i,k}$ becomes the first TED packet at h_{Ni}. Since the service rate is constant while $\mathrm{TED}_{i,k}$ is being processed, we get the following relation:

$$\begin{aligned} D^{Ni}_{i,k}(j) &= D^{Ni}_{i,k}(\mathrm{TED}) - \delta^{Ni}_{i,k} \cdot B_{i,k}(j)/\overline{B}^n_{i,k} \\ &\leqslant D^{Ni}_{i,k}(\mathrm{TED}) - p_i \cdot B_{i,k}(j)/\overline{B}^n_{i,k}. \end{aligned} \tag{21}$$

Combining (18) and (21), we obtain the following relation:

$$\begin{aligned} D^{Ni}_{i,k}(j) - A^1_{i,k}(j) &\leqslant D^{Ni}_{i,k}(\mathrm{TED}) - A^1_{i,k}(\mathrm{TED}) \\ &\quad - p_i \cdot \left(B_{i,k}(j)/\overline{B}^n_{i,k} - B_{i,k}(j)/B_{i,k}\right) \\ &\leqslant D^{Ni}_{i,k}(\mathrm{TED}) - A^1_{i,k}(\mathrm{TED}) \leqslant d_i^{\max}. \end{aligned} \tag{22}$$

Therefore, the end-to-end delay of each packet in the kth bulk is also less than $d_i^{\max}$. □

Proof of Theorem 3. Since $\mathrm{TED}_{i,k}$ misses its end-to-end deadline, we get $t - A^1_{i,k}(\mathrm{TED}) > d_i^{\max}$. Also, the arrival time at the ingress node of any packet which is queueing in front of $\mathrm{TED}_{i,k}$ at h_n must be earlier than $A^1_{i,k}(\mathrm{TED})$, so they also miss their end-to-end deadlines at time t. □

Proof of Theorem 4. Let h_L be the last-deferment hop of $\mathrm{TED}_{i,k}$; we can divide all the hops in the path of flow i into three groups: (1) the hops before h_L, (2) the last-deferment hop h_L and (3) the hops after h_L. If reduction is performed at a hop in group 1, according to (20), this will only affect the value of $A^n_{i,k'}$, which will be decreased by $\overline{\delta}$ after reduction. So the end-to-end delay of $\mathrm{TED}_{i,k}$ will only be decreased by $\overline{\delta}$. If reduction is performed in h_L, the nodal processing times $\delta^n_{i,q}$, where $k' \leqslant q \leqslant k$, will all be decreased by $\overline{\delta}$, so the end-to-end delay of $\mathrm{TED}_{i,k}$ can be reduced by up to $(k - k' + 1) \cdot \overline{\delta}$. Otherwise, if the reduction is performed in a hop which belongs to group 3, the end-to-end delay of $\mathrm{TED}_{i,k}$ will only be decreased by $\overline{\delta}$ because the nodal delays of $\mathrm{TED}_{i,k}$ are all equal to the processing time at the nodes behind h_L. Therefore, performing reduction on the last-deferment hop will result in the greatest reduction of the end-to-end delay of $\mathrm{TED}_{i,k}$. □

References

[1] R. Chang, M.C. Chen, J. Ho, M. Ko, An effective and efficient traffic smoothing scheme for delivery of online VBR media stream, in: IEEE Proceedings of INFOCOM'99, New York, March 1999, pp. 447–454.

[2] S.J. Golestani, Congestion-free communication in high-speed packet networks, IEEE Transactions on Communications (1991) 1802–1812, December.

[3] D. Clark, S. Shenker, L. Zhang, Supporting real-time applications in an integrated services packet network: architecture and mechanism, in: Proceedings of ACM SIGCOMM'92, August 1992, pp. 14–26.

[4] H. Zhang, D. Ferrari, Rate-controlled static-priority queueing, in: Proceedings of INFOCOM'93, March 1993, pp. 227–236.

[5] S. Floyd, V. Jacobson, Link-sharing and resource management models for packet networks, IEEE/ACM Transactions Networking 3 (1995) 365–386. August.

[6] R. Braden et al., Resource reservation protocol (RSVP) – version 1 functional specification, in: IETF RFC 2205, September 1997.

[7] ATM Forum, ATM user-network interface (UNI) signaling specification version 4.1, ATM Forum, April 2002.

[8] G. Kesidis, J. Walrand, C.S. Chang, Effective bandwidths for multiclass Markov fluids and other ATM sources, IEEE/ACM Transactions on Networking 1 (1993) 424–428.

[9] D. Ferrari, D. Verma, A scheme for real-time channel establishment in wide-area network, IEEE Journal on Selected Areas in Communications (1990) 368–379. April.

[10] T.M. Chen, S.S. Liu, V.K. Samalam, The available bit rate service for data in ATM networks, IEEE Communication Magazine (1996) 56–71, May.

[11] ITU-T, Traffic control and congestion control in B-ISDN, ITU-T Rec. I.371, No. 6-14, 1995.

[12] G.G. Xie, S.S. Lam, Real-time block transfer under a link-sharing hierarchy, IEEE Transactions on Networking 6 (1) (1998).

[13] A. Parekh, R.G. Gallager, A generalized processor sharing approach to flow control in integrated services networks: the multiple node case, IEEE/ACM Transactions on Networking 2 (1994) 137–150. April.

[14] L. Georgiadis, R. Guerin, V. Peris, K.N. Sivarajan, Efficient network QoS provisioning based on per node traffic shaping, IEEE/ACM Transactions on Networking 4 (1996) 482–501. August.

[15] C. Li, E. Knightly, Coordinated multihop scheduling: a framework for end-to-end services, IEEE/ACM Transactions on Networking 10 (6) (2002), December.

[16] C. Li, E. Knightly, Schedulability criterion and performance analysis of coordinated schedulers, IEEE/ACM Transactions on Networking 13 (2) (2005). April.

[17] K. Zhu, Y. Zhuang, Y. Viniotis, Achieving end-to-end delay bounds by EDF scheduling without traffic shaping, in: Proceedings of IEEE INFOCOM 2001, vol. 3, April 2001, pp. 1493–1501.

[18] I. Stoica, H. Zhang, S. Shenker, R. Yavatkar, D. Stephens, A. Malis, Y. Bernet, Z. Wang, F. Baker, J. Wroclawski, C. Song, R. Wilder, Per hop behaviors based on dynamic packet states, Internet Draft. Available from: <http://www.cs.cmu.edu/~istoica/DPS/draft.txt>.

[19] S. Kent, R. Atkinson, Security architecture for the Internet protocol, IETF, RFC 2401, November 1998.

[20] M. Claypool, R. Kinicki, A. Kumar, Traffic sensitive active queue management, in: Proceedings of IEEE INFOCOM 2005, vol. 4, March 2005, pp. 2764–2769.

[21] A. Vagish, T. Znati, R. Melhem, Per-node delay assignment strategies for real-time high speed network, in: Proceedings of Globecom'99, pp. 1323–1327.

[22] A.K. Parekh, R.G. Gallager, A generalized processor sharing approach to flow control – the single-node case, IEEE/ACM Transactions on Networking 1 (3) (1993) 344–357, June.

[23] J. Bennett, H. Zhang, $\mathrm{WF}^2\mathrm{Q}$: worst-case fair weighted fair queueing, in: Proceedings of IEEE INFOCOM 1996, March 1996, pp. 120–128.

[24] M. Shreedhar, G. Varghese, Efficient fair queueing using deficit round robin, in: Proceedings of ACM SIGCOMM'95, August 1995, pp. 231–242.

[25] L. Delgrossi, L. Berger, Internet stream protocol version 2 (ST2), protocol specification – version ST2+, RFC 1819, Internet Engineering Task Force, August 1995.

[26] NS2, The network simulator NS (version 2). <http://www.isi.edu/nsnam/ns/>.

[27] P. Seeling, M. Reisslein, B. Kulapala, Network performance evaluation using frame size and quality traces of single-layer and two-layer video: a tutorial, IEEE Communications Surveys and Tutorials 6 (2) (2004) 58–78. Available from: <http://trace.eas.asu.edu>.

Yung-Cheng Tu was born in Taipei, Taiwan in 1974. He received the BS and MS degrees in Computer Science from National Tsing Hua University, Taiwan, in 1996 and 1998. He is currently a PhD student in the Department of Computer Science at National Tsing Hua University, Taiwan. His currently research interests include real-time scheduling, QoS management and wireless communication.

Meng Chang Chen received the BS and MS degrees in Computer Science from National Chiao-Tung University, Taiwan, in 1979 and 1981, respectively, and the PhD degree in Computer Science from the University of California, Los Angeles, in 1989. He was with AT&T Bell Labs from 1989 to 1992. He joined Institute of Information Science, Academia Sinica, Taiwan and has held Associate Research Fellowship since July 1996. From 1999 to 2002, he took additional responsibility as Deputy Director of the institute. From 2000 to the end of 2003, he served as the chair of Standards and Technology Transfer group of the National Science and Technology Program for Telecommunications Office (NTPO). His current research interests include wireless MAN/LAN, network systems with QoS supports, multimedia systems and transmissions, operating system, and data and knowledge engineering.

Yeali Sunny Sun was born in Taipei, Taiwan. She received her BS degree in Computer Science from National Taiwan University in 1982, and the MS and PhD degrees, both in Computer Science, from the University of California, Los Angeles (UCLA) in 1984 and 1988, respectively. From 1988 to 1993, she was with Bell Communications Research Inc. (Bellcore; now Telcordia), where she was involved in the area of planning and architecture design of information networking, broadband networks, and network and system management. In August 1993, she jointed National Taiwan University and is currently a professor and chairperson of the Department of Information Management. She was the director of the Information Networking Group in the Computer Center of National Taiwan University, responsible for managing the university campus network and Taiwan Academia Networks (TANet) Northern Regional POP (Point Of Presence) and leading an advanced networking research team prototyping and conducting a series of field trials and experiments on policy-based QoS delivery services, bandwidth management, and Internet pricing for congestion control. In 1996–2002, she served in the TANet Technical Committee, Steering Committee of the National Broadband Experimental Network (NBEN) and Internet2, and IP Committee of TWNIC. Her research interests are in the area of mobile Internet, quality of service (QoS), content classification, wireless mesh networks, multimedia content delivery, Internet pricing and network management, and performance modeling and evaluation.

Wei-Kuan Shih received the BS and MS degrees in Computer Science from the National Taiwan University, and the PhD degree in Computer Science from the University of Illinois, Urbana-Champaign. He is an Associate Professor in the Department of Computer Science at the National Tsing Hua University, Taiwan. His research interests include real-time systems and wireless communication.

Please cite this article in press as: Y.-C. Tu et al., Enhanced bulk scheduling for supporting delay ..., Comput. Netw. (2008), doi:10.1016/j.comnet.2007.11.020