Harmonic Proportional Bandwidth Allocation and Scheduling for Service Differentiation on Streaming Servers

Xiaobo Zhou, *Member*, *IEEE Computer Society*, and Cheng-Zhong Xu, *Senior Member*, *IEEE*

Abstract—To provide ubiquitous access to the proliferating rich media on the Internet, scalable streaming servers must be able to provide differentiated services to various client requests. Recent advances of transcoding technology make network-I/O bandwidth usages at the server communication ports controllable by request schedulers on the fly. In this article, we propose a transcoding-enabled bandwidth allocation scheme for service differentiation on streaming servers. It aims to deliver high bit rate streams to high priority request classes without overcompromising low priority request classes. We investigate the problem of providing differentiated streaming services at application level in two aspects: stream bandwidth allocation and request scheduling. We formulate the bandwidth allocation problem as an optimization of a harmonic utility function of the stream quality factors and derive the optimal streaming bit rates for requests of different classes under various server load conditions. We prove that the optimal allocation, referred to as harmonic proportional allocation, not only maximizes the system utility function, but also guarantees proportional fair sharing between classes with different prespecified differentiation weights. We evaluate the allocation scheme, in combination with two popular request scheduling approaches, via extensive simulations and compare it with an absolute differentiation strategy and a proportional-share strategy tailored from relative differentiation in networking. Simulation results show that the harmonic proportional allocation scheme can meet the objective of relative differentiation in both short and long timescales and greatly enhance the service availability and maintain low queueing delay when the streaming system is highly loaded.

Index Terms—Service differentiation, harmonic proportional bandwidth allocation, video transcoding, streaming bit rate, feedback queue.

1 Introduction

Scalable streaming servers must be able to provide different levels of quality of service (QoS) to clients. It is because clients are different in their visiting interests, access patterns, service charges, and receiving devices. They can connect to a streaming server using a wide variety of devices, ranging from set-top boxes to PDAs. Their capabilities to receive, process, store, and display continuous media (i.e., video and audio) can vary greatly. Given the diversity of client devices and their needs, the server has to tailor the content and provide different QoS levels accordingly. From the perspective of the server, the request arrival rate changes with time and, hence, it is nonstationary. Allocating resources to accommodate the potential peak arrival rate may not be cost-effective, if not impossible. In other words, it is desirable to provide different QoS levels to different requests during various access periods. To the end, there is a growing demand for replacing the current same-service-to-all paradigm with a model that treats client requests differently based on the access patterns of clients and the resource capacities of servers. Service differentiation aims to provide predictable and controllable per-class QoS levels to requests of different classes. It can also enhance the service availability because a request may be admitted and processed with the negotiated and degraded service quality by a heavily loaded server rather than being simply rejected.

This service differentiation problem was first addressed in the network core. Differentiated Services has been an active research topic in the arena of networking since its architecture was formulated by IETF in 1998 [7]. Its goal is to define configurable types of packet forwarding, so as to provide per-hop differentiated services for large aggregates of network traffic. Network alone is not sufficient to support end-to-end differentiation. There are recent studies on service differentiation provisioning in Web applications at the server side. For conventional Web applications, CPU cycle and disk I/O bandwidth are major resource constraints and responsiveness is a primary QoS metric. Current responsiveness differentiation strategies mostly rely on admission control and priority-based resource allocation and scheduling on individual servers [1], [11], [24], [42], [43], and node partitioning in server clusters [8], [45]. The characteristics and requirements of streaming services differ significantly from those of conventional Web services. I/O-intensive streaming services such as Video-on-Demand (VoD) are usually constrained by disk-I/O and especially network-I/O bandwidth at the server communication ports [2], [14], [19], [20], [33], [44]. The service

- *X. Zhou is with the Department of Computer Science, University of Colorado at Colorado Springs, 1420 Austin Bluffs Parkway, PO Box 7150, Colorado Springs, CO 80933. E-mail: zbo@cs.uccs.edu.*
- *C.-Z. Xu is with the Department of Electrical and Computer Engineering, Wayne State University, 5050 Anthony Wayne Drive, Detroit, MI 48202. E-mail: czxu@ece.eng.wayne.edu.*

Manuscript received 15 July 2002; revised 11 Nov. 2003; accepted 11 Feb. 2004.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number 116959.

quality is measured not only by *startup latency*, but also by allocated stream bandwidth—*streaming bit rate*. The responsiveness oriented strategies are not sufficient for service differentiation on streaming servers. For example, priority-based request scheduling approaches are not sufficient because they cannot differentiate streaming bit rate. Dynamic node-partitioning strategies are not applicable because streaming services are continuous and long-lived.

There were studies on QoS-aware bandwidth management for rich media Web services; see [9], [16] for examples. Their focuses were on transformation of the format, color depth, and sizes of images as well as rich-texts to make a good trade off between user-perceived document quality and transmission time. This paper shares the objective of those efforts on content adaptation, but focuses on server-side bandwidth management for streaming applications. The characteristics of streaming media vary tremendously according to the content, the compression scheme, and the encoding scheme. At application level, a primary performance metric perceived by clients is the media quality (i.e., streaming bit rate). There are essentially two encoding schemes: constant bit rate (CBR) and variable bit rate (VBR). CBR scheme maintains a constant streaming bit rate by varying media quality. It generates predictable media file sizes and simplifies the allocation of server and network resources. Many server-side network-I/O bandwidth allocation schemes and media placement strategies were addressed in CBR context [2], [14], [18], [19], [20], [25], [33], [44]. In contrast, VBR scheme ensures constant media quality by varying streaming bit rate. However, VBR streams have high variability in resource requirements, which can lead to low I/O bandwidth utilization on servers [4], [5], [15], [38]. Both schemes have their advantages and disadvantages. In this article, we address the problem of providing differentiated streaming services in the CBR context.

Without loss of generality, we consider video as an example of streaming media. The idea of performing bandwidth management through adaptation of video frame rate and color depth was demonstrated by early experimental video gateway systems; see [3] for an example. Layered video coding techniques represent video in a hierarchy of quality enhancement layers. There are recent studies on distributing layered encoded video through proxy caching and multicast [22], [39]. Note that the number of enhancement layers is rather limited. Recent advances in real-time transcoding technology make it possible to dynamically transform a video stream from its original encoding bit rate to degraded ones at a fine-grained level [29], [40]. In other words, they can adjust the bit rate of a CBR-encoded video stream on the fly according to the allocated network-I/O bandwidth and make the bandwidth usages controllable by the server's request scheduler at the application level.

In this article, we investigate QoS-adaptive bandwidth allocation schemes for service differentiation on streaming servers by taking advantage of transcoding. It complements the existing efforts on QoS-fixed bandwidth allocation for system scalability [2], [14], [19], [20], [33]. Fig. 1 shows a queueing network with N client request classes in a streaming cluster with M servers. It assumes a dispatcher

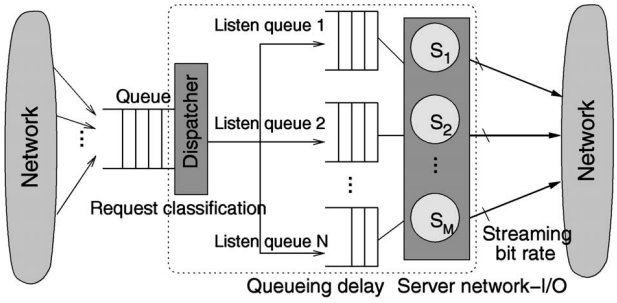


Fig. 1. A queueing network model for a streaming cluster.

to make admission decisions and determine the server for each incoming request. For each admitted and classified request by the dispatcher, two key quality factors are the streaming bit rate and the delay in the listen queue. To provide differentiated services, the dispatcher-based request scheduler needs to determine: 1) what network-I/O bandwidth (bit rate) should be allocated to the stream for the request and 2) when the stream should be scheduled to deliver.

The primary contributions of this work are:

1. We formulate the problem of network-I/O bandwidth allocation at the application level for service differentiation as an optimization of a harmonic utility function of the stream quality factors. The optimization considers the service availability and differentiation constraints. We derive an optimal bandwidth allocation scheme, called harmonic proportional allocation, for request classes with different priorities under various server load conditions.
2. We prove that the harmonic proportional bandwidth allocation scheme not only maximizes the utility function, but also guarantees proportional bandwidth sharing between the classes with respect to their predefined differentiation weights. We give a feedback queue technique to reduce the impact of the variance of interarrival time distribution on differentiation performance.
3. We conduct extensive simulations of the allocation scheme with two popular request scheduling approaches, and compare it with an absolute differentiation strategy and a proportional-share differentiation strategy tailored from the arenas of packet networks and operating systems. Simulation results show the harmonic proportional bandwidth allocation scheme achieves the objective of relative differentiation in both short and long timescales and enhances the service availability to a great extent when the system load is high.

The rest of the paper is organized as follows: Section 2 gives a brief review of previous related work. Section 3 models the network-I/O bandwidth allocation problem for service differentiation. Section 4 presents two bandwidth allocation schemes: harmonic proportional allocation and proportional-share allocation. Section 5 discusses a number of implementation issues of the allocation schemes. Section 6 shows a comprehensive performance evaluation. Section 7 concludes the article with remarks on future work.

2 Related Work

Streaming technology is critical to many popular multimedia applications, such as VoD, distance learning, digital library, etc. Early studies focused on data layout and data retrieval, memory buffering, admission control, and disk scheduling on individual servers; see [32] for an instance. Recent focuses are on scalability and reliability on distributed servers [10], [17], [18], [25], [41], [44]. Streaming services are I/O-intensive. Multicast and periodic broadcast strategies have long been studied for reducing disk-I/O and network-I/O requirements at server side; see MFQL batching and multicast [2], chaining [33], patching [20], and skyscraper broadcast [19] for representatives. In those bandwidth allocation schemes, the primary QoS metric, streaming bit rate, is fixed and constant (i.e., CBR) and it is the same to all streams. Anastasiadis et al. proposed lexicographically optimal algorithms for bandwidth smoothing of VBR streams on servers and at the network edge for broadband traffic [4], [5]. A primary objective is to improve the throughput and scalability of the streaming systems by VBR bandwidth smoothing. In this paper, we investigate QoS-adaptive bandwidth allocation schemes enabled by transcoding-based bandwidth adaptation for providing differentiated streaming services in the context of CBR scheme. Our work is complementary to the previous efforts.

The primary objective of service differentiation is to provide differentiated levels of QoS to different traffic classes by dynamically allocating available resources. Its idea stemmed from QoS adaptation and differentiation in the network core. There are many efforts in providing proportional queueing-delay differentiation in packet forwarding and loss differentiation in packet dropping; see [13], [27] for examples of research in the networking community.

Servers are a major force in end-to-end service differentiation. There are many efforts on providing responsiveness differentiation in Web services [1], [8], [11], [24], [42], [43], [45]. Strategies are mostly based on priority-based resource allocation and scheduling with admission control. For instance, Chen and Mohapatra [11] addressed strict priority scheduling strategies for controlling CPU utilization on Internet servers. The results showed that responsiveness differentiation can be achieved, but the quality spacings among different classes cannot be quantitatively controlled. In [42], [43], we proposed processing rate allocation strategies to achieve proportional slowdown differentiation on various Web servers. In [45], Zhu et al. adopted a multiserver queueing model to guide node-based resource allocation optimization in server clusters. Based on the measured workload of different request classes and their priority levels, a dynamic node partitioning strategy adaptively partitions the server nodes of a cluster and allocates them to handle requests of different classes. OS support for service differentiation has also been addressed in prior efforts [6], [28], [34]. Sundaram et al. presented a multimedia operating system, QLinux, which employs hierarchical class-specific schedulers to meet the diverse performance requirements of Web and multimedia applications [34].

There were a few studies that focused on network-I/O bandwidth scheduling for service differentiation on multimedia servers. Fox et al. adopted a proxy-based approach to content adaptation, in which proxy agents placed between clients and servers perform aggressive computation and storage on behalf of clients [16]. Image adaptation techniques are used to adapt content to network and client variations. Chandra et al. designed a quality-aware image transcoding technique to provide differentiated multimedia Web services at application level [9]. Image transcoding is used by the server to customize the size of image objects (JPEG) and, hence, manage the available network-I/O bandwidth of the server. The primary performance metric is the image quality factor. The bandwidth management schemes allow the server to provide acceptable response time to clients by trading off image quality for image size. Like the work in [16], this approach also does not quantitatively control quality spacings between different classes.

This article investigates the problem of transcoding-enabled bandwidth allocation for providing differentiated streaming services. The objective is to provide high bit rate streams to high priority request classes without overcompromising low priority request classes according to server network-I/O bandwidth constraints and workloads. Although general resource management models, such as Q-RAM [30] and FARA [31], can be applied to the control of bandwidth allocation on streaming servers, their objectives are mostly on maximizing overall system performance by content adaptation. Adaptability aside, service differentiation schemes also demand QoS predictability and fairness. In this paper, we define a harmonic function of the quality factors as the optimization function. The derived allocation scheme not only maximizes the overall system performance, but also provides a guarantee of proportional fair sharing between the request classes with respect to their differentiation weights. Our work differs from prior approaches in that it offers predictable and fair quality spacings to different classes based on theoretical foundation of resource allocation. Transcoding is the enabling technology for our work. Previous work on transcoding was mostly on algorithms and performance evaluations [29], [40]. The focus of this paper is on its impact on providing differentiated streaming services.

Our work is also related to the call level bandwidth allocation and admission control in telecommunication networks. Kelly et al. analyzed two classes of rate control algorithms for communication networks [23]. Their focus was on the stability and fairness issues. Our work shares their objective on proportional fairness provisioning, and focuses on service differentiation on streaming servers. The proposed allocation schemes can be applied to call-level constant bit rate allocation at a bandwidth bottleneck, if there is the support of some kind of transcoding. However, the focus of our work concerns the differentiation predictability and controllability, while call admission and bandwidth allocation strategies primarily concern the call blocking probability, and throughput of links and the whole network.

3 Bandwidth Allocation for Differentiated Streaming Services

The objective of the network-I/O bandwidth allocation problem is to determine stream bandwidth of each request

class in such a way that the overall QoS is optimized and, meanwhile, the stream qualities are guaranteed to be proportional to their prespecified differentiation weights. The weights can be determined by clients' priorities, receiving devices, payments, etc. Divide the scheduling process into a sequence of short intervals of bandwidth allocation and request scheduling. The bandwidth allocation decision needs to be carried out in each interval, based on the measured bandwidth release rate and the predicted arrival rate of request classes.

3.1 Service Differentiation Models and Properties

Within the service differentiation infrastructure, incoming requests from different clients are grouped into N classes that are sorted in a nonincreasing order according to their desired levels of QoS. There are two basic types of service differentiation schemes [7]. One is absolute service differentiation, in which each request class receives an absolute share of resource usages (network-I/O bandwidth in this context). When this scheme is applied to service differentiation on a streaming server, it requires the server to statically maintain multiple replicas of a video encoded with different bit rates regarding to the certain QoS levels if there is no support of adaptive transcoding. A primary concern with this scheme is its weak ability of adaptation to fluctuating arrival rates from various clients. In Section 6, we will show that, without a priori knowledge about the clients' access patterns, this scheme could lead to a low resource utilization.

The second one is relative service differentiation. In this scheme, service quality of class i is better or at least no worse than class $i+1$ for $1 \leq i \leq N-1$. The term "or no worse" is necessary since, in heavy-load conditions, all request classes will tend to experience their minimum QoS levels. In this context, applications and clients do not get an absolute service quality assurance. Instead, this differentiation scheme assures that the class with a higher desired level of QoS (referred to as a higher class) will receive relatively better service quality than the class with a lower desired level of QoS (a lower class). So, it is up to the applications and clients to select appropriate QoS levels that best meet their requirements, cost, and device constraints. For relative differentiation, a streaming server has to assign different QoS levels to clients based on the dynamic load conditions. Therefore, it needs the support of adaptive content adaptation techniques such as transcoding since it is high costly, if not impossible, to have sufficient number of replicas a priori for the differentiation needs.

In order for a relative service differentiation scheme to be effective, it should satisfy two basic properties [13]:

1. *Predictability:* The differentiation should be consistent. Higher classes should receive better or no worse services than lower classes independent of variations of class load conditions.
2. *Controllability:* The scheduler must contain a number of controllable parameters, which are adjustable for quality spacings between classes.

In streaming services, a primary QoS metric is the allocated stream bandwidth (streaming bit rate). It has lower bound and upper bound. For example, 1 Mbps could be referred to as the lower bound of streaming bit rate of an MPEG-I movie for general clients. Different client classes may expect different lower bounds. The upper bound is the encoding bit rate of the video because today's transcoding techniques can only dynamically degrade the streaming bit rate. We argue that an effective relative service differentiation scheme on streaming servers should meet the following additional requirements:

1. *Upper and lower bounds:* Quality guarantees should be provided for all requests. Admission control is needed to prevent the system from being overloaded.
2. *Availability:* One goal of offering differentiated services on streaming servers is to serve as many requests as possible at sufficiently acceptable QoS level, to gain and retain the business. If the available network-I/O bandwidth at server side is enough to provide the lower bounds of QoS level to all requests, rejection rate could be minimized.
3. *Fairness:* Requests from lower classes should not be overcompromised for requests of higher classes.

3.2 Network-I/O Bandwidth Allocation

The basic idea of network-I/O bandwidth allocation for providing differentiated streaming services is to divide the allocation and scheduling process into a sequence of short intervals. In each interval, based on the measured bandwidth release rate of the servers and the predicted arrival rate of the classes, the stream bandwidth (bit rate) of each class is determined based on differentiation requirements and available resources. The streams for the request classes are allocated and then scheduled according to specific scheduling approaches.

In the following, we consider the bandwidth allocation problem for N request classes. We define a *channel* as the bandwidth unit allocated to a stream. It is the lower bound of streaming bit rate for the class which has the minimum QoS expectation. Let λ_i be the arrival rate of requests in class i in an allocation interval. Let μ_i be the rate of channel allocation to requests in class i. We define a *quality factor* q_i of requests in class i $(1 \leq i \leq N)$ as

$$q_i = \frac{\mu_i}{\lambda_i}. \tag{1}$$

It represents the quality of request class i in the current bandwidth allocation interval. For example, if the rate of channel allocation to class i is 8 per time unit and the request arrival rate of class i is 4 per time unit, the quality for requests in class i in the current interval is two channels. We define the quality factor of a stream as a linear utility function of its allocated bandwidth because the user-perceived quality of a CBR-encoded video can be seen as proportional to its streaming bit rate.

Let B denote the bound of the aggregate channel allocation rate of the servers during the current bandwidth allocation interval. It is the ratio of the number of channels to be released in the current interval plus the unused ones from previous intervals to the length of the allocation interval. We have the resource constraint of

$$\sum_{i=1}^{N} \mu_i \leq B. \tag{2}$$

From the system's perspective, service availability is an essential objective. Suppose the request arrival rate of class i in current scheduling period is 4 per time unit and the rate of channel allocation to class i is 3 per time unit because of heavy system load. Although the calculated quality factor for class i is 0.75, due to the existence of lower bound of streaming bit rate, at least one request from this class may be rejected and other requests receive the lower bound of streaming bit rate.

Let L_i denote the lower bound of streaming bit rate for class i. Assume all video objects have the same initial encoding bit rate U. It represents the upper bound of streaming bit rate of all classes. We consider the network-I/O bandwidth allocation for service differentiation when $\sum_{i=1}^{N} L_i\lambda_i \leq B < \sum_{i=1}^{N} U\lambda_i$. That is, the total number of available channels is enough to guarantee the minimum QoS level, but not enough to support the maximum QoS level for all contending classes. Otherwise, the problem is either trivial or infeasible. For example, when $B \geq \sum_{i=1}^{N} U\lambda_i$, we can simply give each class a quality factor U, the upper bound of streaming bit rate. When $B < \sum_{i=1}^{N} L_i\lambda_i$, the admission control must do rejection and we do not consider provisioning of service differentiation. Hence, for service availability, we have an additional constraint:

$$L_i \leq q_i \leq U \qquad 1 \leq i \leq N. \tag{3}$$

4 Harmonic Proportional Share Allocation Scheme

In this section, we first present a proportional-share bandwidth allocation scheme tailored from proportional-share scheduling in packet networks and operating systems. Then, we propose a harmonic proportional allocation scheme that not only optimizes an overall system utility function, but also ensures proportional bandwidth sharing between request classes. Consider N contending request classes with predefined quality differentiation weights $\delta_1, \delta_2, \cdots, \delta_N$. Since relative service differentiation requires class i would receive better or no worse services than class $i+1$, without loss of generality, weights δ_i are sorted in a nonincreasing order as $\delta_1 \geq \delta_2 \geq \cdots \geq \delta_N$.

4.1 Proportional-Share Bandwidth Allocation

The proportional-share bandwidth allocation scheme borrows its idea from proportional-share scheduling in the arenas of networking and operating systems. At the OS level, there has also been a renewal of interest in fair-share schedulers, which is now usually called proportional-share scheduling. For instance, Waldspurger and Weihl proposed lottery scheduling for fair share resource management for CPU utilization [36]. At the network level, it is usually called fair queueing. Dovrolis et al. recently proposed a proportional model and various algorithms to ensure the per-hop queueing delay of the packets in different classes to be proportional to their predefined differentiation weights [13]. The proportional model is interesting because it is fair and predictable. It has been accepted as an important relative differentiation model in the network core.

Our proportional-share bandwidth allocation scheme for differentiated streaming services assigns quality factors q_i to request classes in proportion to their quality differentiation weights δ_i. Recall that a request class i needs to maintain a lower bound of quality factor L_i. For service availability, in each bandwidth allocation interval, the proportional-share allocation scheme states that for any two classes i and j, $1 \leq i, j \leq N$,

$$\frac{q_i - L_i}{q_j - L_j} = \frac{\delta_i}{\delta_j}, \tag{4}$$

subject to constraints of (2) and (3).

According to constraint (2), the objective of (4) leads to a proportional bandwidth allocation rate

$$\mu_i^* = L_i\lambda_i + \left(B - \sum_{k=1}^{N} L_k\lambda_k\right) \frac{\delta_i\lambda_i}{\sum_{k=1}^{N} \delta_k\lambda_k}. \tag{5}$$

It follows that the proportional quality factor of class i is calculated as:

$$q_i^* = L_i + \left(B - \sum_{k=1}^{N} L_k\lambda_k\right) \frac{\delta_i}{\sum_{k=1}^{N} \delta_k\lambda_k}. \tag{6}$$

The quality factor of (6) reveals that the proportional-share allocation scheme generates consistent and predictable schedules for requests of different classes on streaming servers. The classes with higher weights δ_i receive better or no worse services than the classes with lower weights δ_i, independent of variations of the class loads. The quality factor of each request class i is controlled by its channel allocation rate μ_i.

4.2 Harmonic Proportional Allocation

Note that the proportional-share allocation scheme that aims to control the interclass quality spacings does not necessarily yield best overall system QoS. To optimize the overall system QoS and meanwhile ensuring quality spacings between the classes, we define a weighted harmonic function of the quality factors of all the classes, which ensures the lower bounds of streaming bit rate, as the optimization function. Specifically, we formulate the bandwidth allocation for service differentiation as the following optimization problem:

$$\text{Minimize } \sum_{i=1}^{N} \delta_i \frac{1}{q_i - L_i} \tag{7}$$

$$\text{Subject to constraints (2) and (3).}$$

The minimization of the harmonic objective function (7) requires that requests of higher classes would be allocated more bandwidth, but this biased allocation should not overcompromise the share of requests from lower classes. Note that, when $\sum_{k=1}^{N} L_k\lambda_k = B$, the quality factor q_i of each class i is equal to its lower bound L_i and there is no need for optimization and differentiation due to the service availability requirement.

The objective function (7) is continuous and it is convex and separable in its variables. The resource allocation constraint (2) describes the total amount of resource to be allocated. Constraint (3) ensures the positivity of variables. This optimization becomes a special case of the resource allocation problem. This kind of resource allocation optimization has been applied in many systems [2], [30], [37]. We

define the optimization function as the weighted harmonic function of the quality factors of all classes. It implies that the classes with higher differentiation weights get higher QoS factors and, hence, differentiation predictability is achieved. Interestingly, the derived allocation scheme also guarantees a proportional share allocation between the classes. The rationale behind the objective function is its feasibility, differentiation predictability and proportional fairness properties, as we discussed in Section 3.1.

The optimization above is essentially a continuous convex separable resource allocation problem. According to the foundations of resource allocation optimization theory [21], its optimal solution occurs only if the first order derivatives of each component function of (7) over variables $\mu_1, \mu_2, \ldots, \mu_N$ are equal. Specifically, the optimal solution of (7) occurs when

$$-\frac{\delta_i \lambda_i}{(\mu_i - L_i\lambda_i)^2} = -\frac{\delta_j \lambda_j}{(\mu_j - L_j\lambda_j)^2} \tag{8}$$

for any classes i and j, $1 \le i, j \le N$.

Combining with the constraint (2), the set of equations (8) leads to the optimal allocation scheme

$$\mu_i^* = L_i\lambda_i + \left(B - \sum_{k=1}^{N} L_k\lambda_k\right)\frac{\sqrt{\delta_i\lambda_i}}{\sum_{k=1}^{N}\sqrt{\delta_k\lambda_k}} \qquad 1 \le i \le N. \tag{9}$$

As a result, the optimal quality factor of class i, $1 \le i \le N$, is calculated as

$$q_i^* = L_i + \left(B - \sum_{k=1}^{N} L_k\lambda_k\right)\frac{\sqrt{\delta_i\lambda_i}}{\lambda_i\sum_{k=1}^{N}\sqrt{\delta_k\lambda_k}}. \tag{10}$$

To show the implications of the derived allocation scheme on system behavior, we give the following basic properties regarding the controllability and dynamics due to the optimal allocation scheme:

1. If the differentiation parameter of a class increases, the quality factor of all other classes decreases, while the quality factor of that class increases.
2. The quality factor of a class i decreases with the increase of arrival rate of each class j.
3. Increasing the load of a higher class causes a larger decrease in the quality factor of a class than increasing the load of a lower class.
4. Suppose that a fraction of the class i load shifts to class j, while the aggregate load remains the same. If $i > j$, the quality factor of class i increases, while that of other classes decreases. If $i < j$, the quality factor of class j decreases, while that of other classes increases.

In addition, the optimal allocation scheme has the property of proportional fairness. That is,

Theorem 1. *The optimal allocation scheme of (9), referred to harmonic proportional bandwidth allocation, guarantees a proportional share distribution of excess bandwidth over the minimum requirements between classes with different differentiation weights.*

Proof. Let $\tilde{\lambda}_i$ be the normalized request arrival rates and $\tilde{\lambda}_i = \delta_i\lambda_i$. Define $\tilde{\mu}_i$ as the allocation of excess bandwidth to class i over the minimum requirement $L_i\lambda_i$. According to (9), we have

$$\tilde{\mu}_i = \mu_i^* - L_i\lambda_i = \left(B - \sum_{k=1}^{N} L_k\lambda_k\right)\frac{\tilde{\lambda}_i^{\frac{1}{2}}}{\sum_{k=1}^{N}\tilde{\lambda}_k^{\frac{1}{2}}}. \tag{11}$$

The allocation of $\tilde{\mu}_i$ yields an increment of quality factor, $\tilde{q}_i = \tilde{\mu}_i/\lambda_i$, to the minimum qualify factor of L_i. That is, $\tilde{q}_i = q_i^* - L_i$. By comparing the quality factor increments of two classes i and j, we obtain the quality spacing between the classes, that is:

$$\frac{\tilde{q}_i}{\tilde{q}_j} = \frac{\lambda_j\tilde{\lambda}_i^{\frac{1}{2}}}{\lambda_i\tilde{\lambda}_j^{\frac{1}{2}}} = \sqrt{\frac{\lambda_j}{\lambda_i}}\sqrt{\frac{\delta_i}{\delta_j}}. \tag{12}$$

Equation (12) indicates that the ratio of excess bandwidth allocation between the two classes with given arrival rates is proportional to their predefined differentiation weights. This completes the proof. □

Note that when system load is heavy, that is, $\sum_{k=1}^{N} L_k\lambda_k$ is close to the bound B, all requests are going to be allocated their lower bounds of the bit rate, which would minimize the rejection rate when the system is heavily loaded.

Recall that the quality factor q_i must be less than or equal to the upper bound of streaming bit rate U. According to (10), the calculated q_i could be greater than U when the system is lightly loaded. As a result, certain request classes may not be able to use all their allocated channels. To improve the channel utilization, we can redistribute the excess channels to other request classes or simply leave them for calculating the channel release rate in the next allocation interval.

Remark. According to (11), given fixed λ_i, the classes with higher δ_i get more portion of available network-I/O bandwidth. However, by (12), $\tilde{q}_i \ge \tilde{q}_j$ if and only if $\frac{\delta_i}{\lambda_i} \ge \frac{\delta_j}{\lambda_j}$ holds. Otherwise, the predictability property of service differentiation becomes violated. For differentiation predictability, one solution is temporary weight promotion, as suggested in the context of server node partitioning [45]. That is, the request scheduler temporarily increases quality differentiation weight δ_i, based on the current request arrival rates, so as to ensure $\frac{\delta_i}{\lambda_i} \ge \frac{\delta_j}{\lambda_j}$. The derived proportional-share bandwidth allocation scheme (5) does not have this requirement.

5 Implementation Issues

We built a simulation model for the evaluation of the network-I/O bandwidth allocation schemes with two popular request schedulers on streaming servers. The model divides the simulation process into a sequence of short intervals and performs bandwidth allocation and scheduling functions based on the predicted arrival rates of the request classes and measured bandwidth release rate of the servers in each interval. A fairly accurate estimation of the parameters is needed so that the proposed bandwidth allocation schemes can adapt to the dynamically changing values.

Estimation of Request Arrival Rate. The request arrival rate of each class (λ_i) is estimated by counting the number

of requests from each class occurring in a moving window of certain immediate past periods. The moving window estimation approach based on history information has been used in many similar experiments [2], [12]. A smoothing technique based on a decaying function is applied to take weighted averages over past estimates.

Measure of Bandwidth Allocation Rate Bound (B). Since the focus of this paper is on service differentiation provisioning, we assume that the streaming servers provide support for a video playback function only. Because of the continuous nature of streaming services, it is feasible to measure the bandwidth to be released in the current allocation interval. We employ a similar smoothing window to calculate the bound of bandwidth allocation rate in each allocation interval. It takes into account the bandwidth to be released in the current allocation interval and the excess bandwidth not used in the past allocation intervals.

Admission Control. The derived bandwidth allocation schemes in (5) and (9) ensure that the requests in higher classes always get better services in terms of streaming bit rate. However, streaming services usually have a maximum acceptable waiting time [2]. If the aggregate channel requirement of all classes ($\sum_{k=1}^{N} L_k\lambda_k$) exceeds the bound of bandwidth allocation rate (B) in the current allocation interval due to some bursty traffic, the dispatcher in the streaming cluster imposes admission control and drops those requests which have waited in the queue longer than the maximum acceptable waiting time. The requests to be rejected are first from the lowest differentiated class, and then the second lowest class, and so on.

Feedback Queue. We note that accuracy of the arrival rate estimations in an allocation interval is affected by the variance of interarrival time distributions. It fluctuates due to the existence of bursty traffic. When the actual arrival rates exceed estimated arrival rates, streaming bandwidth would be over-allocated to current requests, leading to queueing delay of subsequent requests. On the other hand, some network-I/O bandwidth would be wasted due to undermeasured streaming bit rates if the actual arrival rates are overestimated. To reduce the impact of the variance on bandwidth allocations, we introduce a feedback queue as a smoothing technique. It takes into account the number of backlogged requests into the estimation of the arrival rates. It calculates $\lambda_i = \lambda_i + \alpha \times l_i$, where l_i is the number of backlogged requests of class i at the end of the past allocation interval and α, $0 \leq \alpha \leq 1$, is a scaling parameter, indicating the percentage of the number of backlogged request in a queue to be included in the calculation of the arrival rate.

Request Schedulers. We implemented the proposed bandwidth allocation schemes with two popular request scheduling approaches. A strict priority scheduler picks requests from higher classes before processing those from lower classes. Requests of lower classes are only executed if no request exists in any higher classes in the current scheduling interval. A First-Come-First-Served with First-Fit backfill (FCFS/FF) [26] picks requests of different classes in the order they arrive, as long as sufficient system resources are available to meet their requirements. In the case that the head-of-the-line request is blocked due to the lack of sufficient resources, first-fit backfilling searches further down the listen queue for the first request that is able to be scheduled immediately.

6 Performance Evaluation

In this section, we examine the impact of service differentiation on system performance and the impact of various bandwidth allocation schemes with the two request schedulers on service differentiation provisioning in terms of streaming bit rate and queueing delay.

The experiments assumed that the videos were encoded with a bit rate of 8 Mbps, a typical value for high-quality MPEG movies. The minimal acceptable bit rate was assumed to be the same for all request classes. It was set to 1 Mbps, a typical value for low-quality MPEG movies. Thus, 8 Mbps and 1 Mbps were the upper and lower bound of streaming bit rate for transcoding, respectively. Video transcoding was used to degrade the streaming bit rate on the fly according to the results of network-I/O bandwidth allocation in (5) and (9). The streaming cluster consisted of eight servers and each server had a network-I/O bandwidth of 1.8 Gbps. In total, the streaming capacity of the cluster ranged from 20 requests at 8 Mbps per minute to 160 requests at 1 Mbps per minute. The aggregate arrival rate (λ) ranged from 20 to 160 requests per minute. Like related experiments in [2], [19], the access patterns were generated by a Poisson process and each video lasted two hours. The maximum acceptable queueing delay was set to four minutes [2]. If the queueing delay of a request exceeded four minutes, either the admission control rejected the request or the client dropped the request.

Due to the lack of service differentiation workload on streaming servers, we adopted a service differentiation workload tailored from an eBay online auction that was used for analysis of queueing-delay differentiation in [45]. It consisted of three classes of client requests, 10 percent requests from registered clients for bidding and posting (Class A, premium), 40 percent requests from registered clients for browsing and searching (Class B, ordinary), and 50 percent requests from unregistered clients (Class C, basic). That is, their request arrival ratios $(\lambda_a, \lambda_b, \lambda_c)$ were $(1, 4, 5)$. Their quality differentiation weights $(\delta_a, \delta_b, \delta_c)$ were assumed to be (4, 2, 1). In the simulation, one allocation and scheduling interval was five minutes. The shorter the interval, the more sensitive the differentiation will be. The longer the interval, the more stable and predictable the differentiation will be. There is a trade off between sensitivity and stability. The workload was estimated as the average in past four intervals. Each representative result reported in this section is an average of 500 runs.

6.1 Impact of Service Differentiation on System Performance

We first examine the system performance due to various differentiated bandwidth allocation schemes. In addition to the proposed proportional-share and harmonic proportional allocation schemes, two static bandwidth allocation schemes are included. A static nonuniform bandwidth allocation scheme provides absolute service differentiation. It allocates the fixed streaming bit rate 4Mbps, 2Mbps, and 1Mbps to requests from class A, B, and C, respectively. A static uniform bandwidth allocation scheme supports no service differentiation; we considered three uniform encoding bit rates: 2Mbps, 3Mbps and 4Mbps for all requests. The streams are delivered according to FCFS/FF scheduling approach, unless otherwise specified.

Fig. 2 shows the impact of service differentiation on the average rejection rate. It shows that the harmonic proportional allocation scheme guarantees service availability.

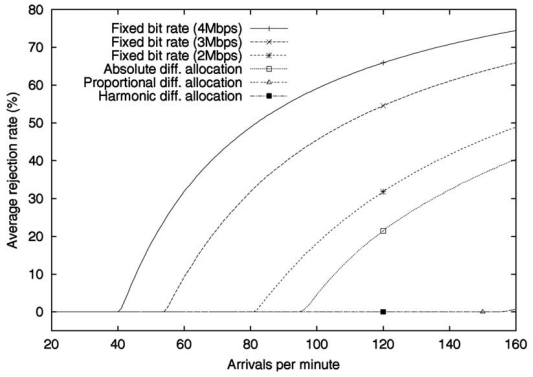


Fig. 2. Impact of service differentiation on average rejection rate.

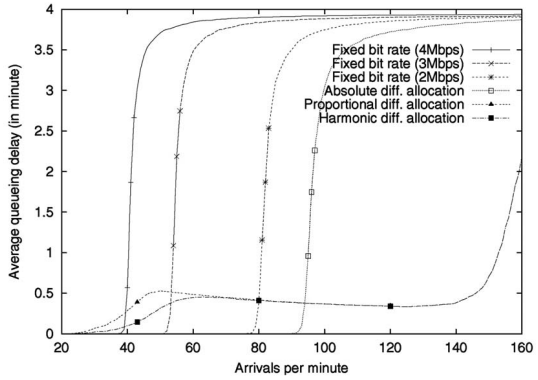


Fig. 4. Impact of service differentiation on average queueing delay.

This is achieved by degrading the streaming bit rates adaptively with transcoding according to system load. The proportional-share allocation scheme achieves results similar to those of the harmonic allocation scheme. The absolute differentiation allocation scheme cannot adapt to system load dynamically and, so, it cannot guarantee service availability. Like the system with the absolute differentiation allocation, the average rejection rate in the system without service differentiation increases abruptly after arrival rate exceeds corresponding knee points. The figure reveals that both the harmonic proportional and proportional-share bandwidth allocation schemes can achieve high throughput and high service availability when servers are heavily loaded.

Fig. 3 shows that the harmonic proportional allocation scheme enables the streaming system to efficiently and adaptively manage its available network-I/O bandwidth. The proportional-share allocation scheme obtains similar results, which were omitted for brevity. On the other hand, the absolute differentiation allocation does not provide such an adaptivity. Like the system without any service differentiation, the system with the absolute differentiation allocation wastes considerable streaming bandwidth when system load is light. They can fully utilize their bandwidth when arrival rate exceeds some knee points. However, as shown in Fig. 2, this utilization comes at the cost of driving rejection rate to unacceptable levels.

Fig. 4 shows the impact of service differentiation on the average queueing delay. It can be seen that without service differentiation or with the absolute service differentiation, the average queueing delay increases abruptly after arrival rate exceeds certain knee points and rejection occurs at the corresponding levels, as shown in Fig. 2. The average queueing delay is approaching and bounded by the maximum acceptable waiting time (four minutes). In contrast, the harmonic proportional and proportional-share allocation schemes maintain the average queueing delay in acceptable degrees at various arrival rates. The queueing delay is due to the variance of interarrival time distributions. It also shows that the harmonic allocation scheme yields slightly lower queueing delay than the proportional-share allocation when the system load is light.

In comparison with the absolute differentiation strategy, both the harmonic and proportional-share allocation schemes make it possible for streaming servers to achieve high throughput, high service availability, and low queueing delay when the servers are heavily loaded.

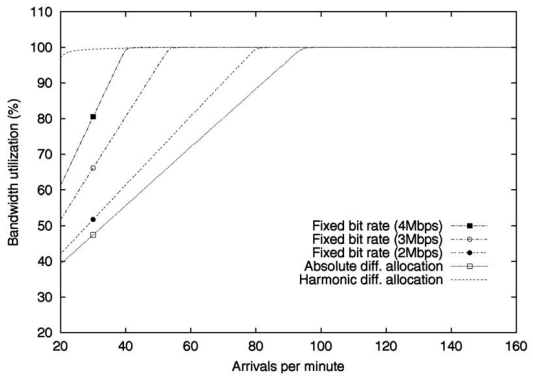


Fig. 3. Impact of service differentiation on bandwidth utilization.

6.2 Impact of Bandwidth Allocation Schemes

The second experiment was on the impact of the various bandwidth allocation schemes on differentiation provisioning in details. Fig. 5 shows a microscopic view of the streaming bit rate of individual requests in the three classes due to the harmonic proportional and proportional-share allocation schemes, when arrival rate is low (50 requests/minute), medium (80 requests/minute), and high (110 requests/minute), respectively. The simulation at each arrival rate was run for 60 minutes. Each point represents the streaming bit rate of individual requests from the classes in consecutive recording time units. It can be seen that both allocation schemes consistently enforce prespecified quality spacings between the classes. We find that both the harmonic proportional and proportional-share allocation schemes achieve short-term objective of service differentiation in terms of streaming bit rate. Although the harmonic proportional allocation scheme was proposed to maximize the overall quality factor of streams, it provides the similar level of proportional differentiation control over the interclass quality spacing as the proportional-share allocation scheme.

Fig. 6 shows the average streaming bit rate of requests from each class due to the harmonic proportional and proportional-share allocation schemes at various arrival rates. The transcoding-enabled bandwidth allocators degrade the streaming bit rate of each class adaptively with varying system load. When system load is light, requests from class A tend to receive the upper bound of streaming bit rate, i.e., 8 Mbps. When system load is moderate, all request classes get their fair shares. When system load is heavy, all request classes tend to receive the lower bound of

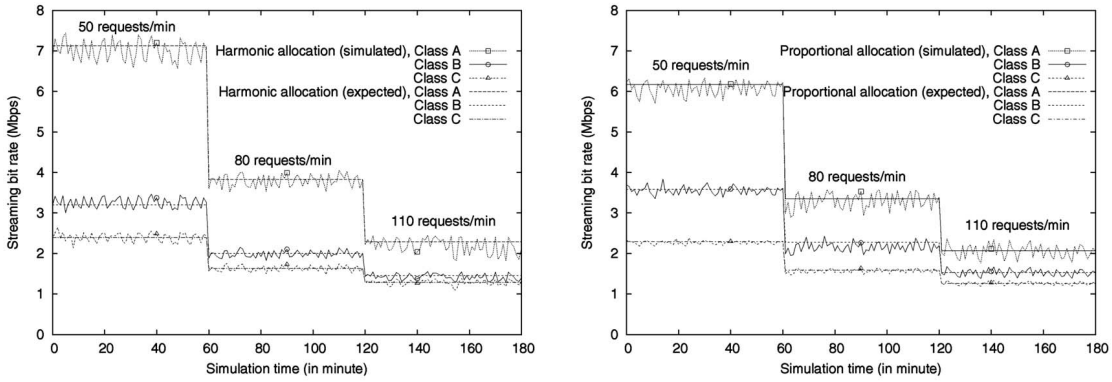


Fig. 5. A microscopic view of the streaming bit rate of requests due to the various bandwidth allocation schemes. (a) The harmonic proportional allocation. (b) The proportional-share allocation.

streaming bit rate, i.e., 1 Mbps. Furthermore, requests from class A receive higher streaming bit rate by the use of the harmonic allocation scheme than by the use of the proportional-share allocation scheme. In general, the harmonic allocation scheme favors requests from higher classes more than the proportional-share allocation scheme. The proportional-share allocation scheme adjusts the quality levels of request classes in proportion to their differentiation weights. The harmonic allocation scheme is also proportional, as shown by (12) and proven in Theorem 1. In all cases, requests from higher classes consistently receive better or no worse service quality than requests from lower classes. Evidently, both the harmonic proportional and proportional-share allocation schemes can achieve long-term objective of service differentiation in terms of streaming bit rate.

Fig. 7 shows the queueing delay of requests from the three classes due to the harmonic proportional and proportional-share allocation schemes at various arrival rates. The queueing delay is due to the variance of interarrival time distributions. When system load is light ($\lambda < 30$), the queueing delay of all classes is trivial. It is because some network-I/O bandwidth was unused during the past allocation intervals due to the existence of upper bound of streaming bit rate. When arrival rate exceeds 40 requests/minute, unexpectedly, we find a "queueing-delay dip" scenario. That is, the queueing delay initially increases and then marginally decreases as arrival rate increases and then increases significantly as arrival rate is close to the system's streaming capacity. Note that the queueing delay is not only affected by the variance of interarrival time distributions, but also affected by the differentiated bandwidth allocations. Because the backlogged requests in queues are not considered in the calculation of arrival rate of classes, the streaming bit rates are overallocated to current requests, leading to higher queueing delay of subsequent requests. As the arrival rate further increases, the requests are allocated with lower streaming bit rates so that the impact of the backlogged requests on the bandwidth overallocation decreases and, thus, the impact on queueing delay of subsequent requests decreases. The impact of the variance of interarrival time distributions on queueing delay dominates and it is significant when the system load is close to the system's streaming capacity ($150 < \lambda < 160$). In Section 6.4, we will show that the proposed feedback queue technique can mitigate this kind of queueing-delay dip scenarios.

Fig. 7 also shows that by the use of the two bandwidth allocation schemes, requests from class A have higher queueing delay compared to requests from classes B and C. Recall that the simulation assumed FCFS/FF request scheduling in combination with the bandwidth allocation schemes. Although FCFS/FF scheduling does not provide any queueing-delay differentiation between different classes, the QoS-aware bandwidth allocation schemes affect the performance metric of queueing delay, as well. Requests from higher classes tend to be allocated with higher

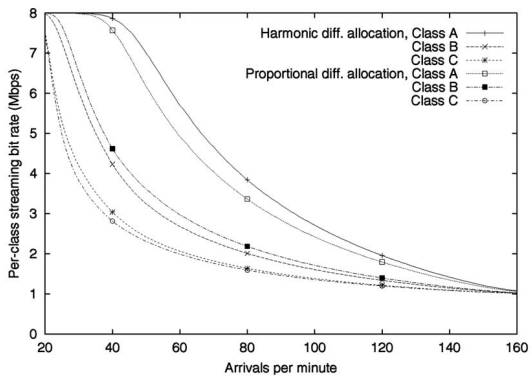


Fig. 6. Differentiated streaming bit rates of request classes.

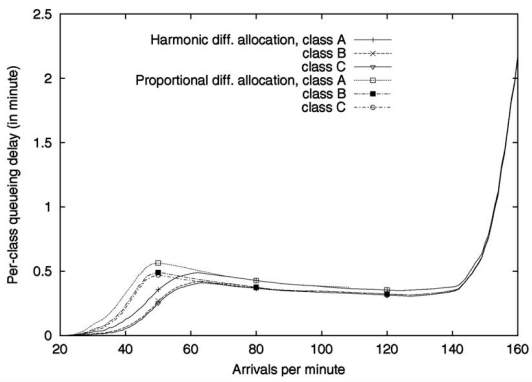


Fig. 7. Queueing delay of request classes.

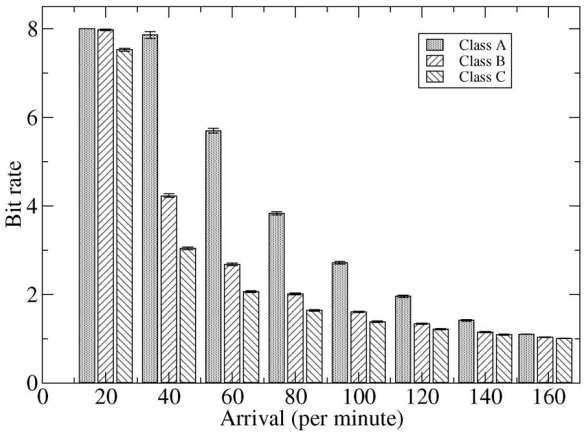


Fig. 8. Confidence intervals of simulated streaming bit rate.

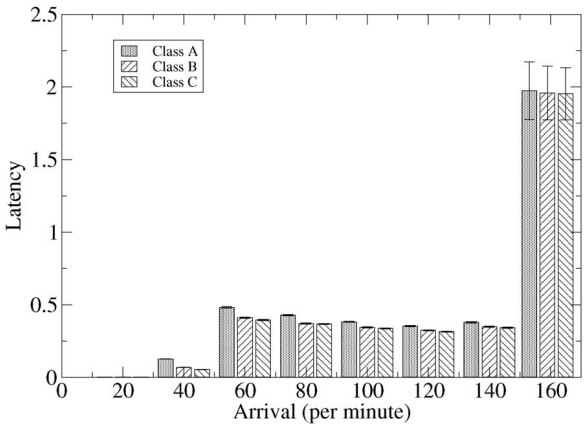


Fig. 9. Confidence intervals of simulated queueing delay.

streaming bit rates. This results in higher queueing delay. Due to the first-fit feature of the FCFS/FF scheduler, on the other hand, requests from lower classes have higher probabilities to be processed with the differentiated streaming bit rates. Fig. 7 also shows that compared with the proportional-share allocation scheme, the harmonic allocation postpones the emergence of queueing delay.

Figs. 6 and 7 illustrate the average values of the QoS metrics. In Figs. 8 and 9, we present 95 percent confidence intervals of the QoS metrics due to the harmonic proportional allocation, i.e., streaming bit rate (Mbps) and queueing latency (minute), together with their mean values for the three classes when the arrival rate is 20, 40, 60, 80, 100, 120, and 159 requests/minute. When the arrival rate is 20 requests/minute, all requests of class A receive the upper bound of streaming bit rate due to the bound's existence. The excess channels are redistributed to classes B and C. The available bandwidth is sufficient for streams of classes B and C and, hence, almost no queueing delay is observed. When the arrival rate is close to the system's streaming capacity (160 requests/minute), requests of all classes tend to receive the uniform lower bound of streaming bit rate (1 Mbps). The confidence intervals were obtained by the method of independent replication [35]. It can be seen that these bounds are uniformly tight. This can be explained by the fact that the variances are only due to the exponential interarrival time distributions. Results of the proportional-share allocation are similar. Due to the space considerations, we do not present confidence intervals for all experiments. We note that the data in Figs. 8 and 9 is entirely representative.

In summary, we observed that both the harmonic proportional and proportional-share allocation schemes can achieve the objective of service differentiation provisioning (in terms of streaming bit rate) in long and short timescales. They have positive side effects on the request queueing delay and the harmonic allocation scheme leads to lower queueing delay when the system is lightly loaded.

6.3 Impact of Request Scheduling Approaches

We have analyzed the proposed QoS-aware bandwidth allocation schemes in combination with the FCFS/FF request scheduling. The preceding experiments have shown that the FCFS/FF scheduler does not differentiate queueing-delay of requests from different classes, although the bandwidth allocation schemes can affect the queueing-delay of various request classes, as shown in Fig. 7. In this experiment, we investigate the relationship between the harmonic proportional bandwidth allocation and request scheduling approaches.

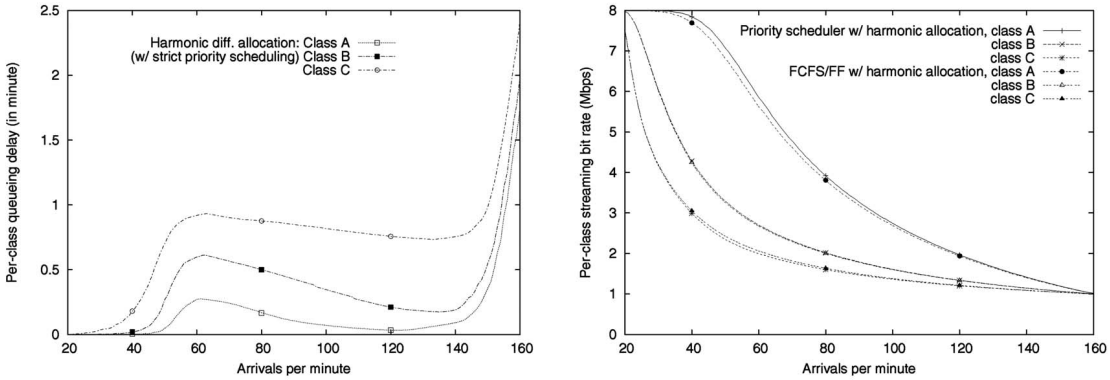
Fig. 10a shows the average queueing delay of requests from three classes due to the strict priority request scheduler. In the strict priority request scheduling, requests in a queue cannot be serviced until the higher priority queues are all empty in the current allocation interval. It shows that the priority scheduler imposes certain degrees of control over queue-delay differentiation between the request classes. The queueing delay of requests from class A is rather limited since the priority scheduler favors the requests of higher classes. This is achieved at the cost of higher queueing delay of requests of lower classes. Note that the strict priority scheduling itself cannot quantitatively control quality spacings between various classes. Time-dependent priority scheduling schemes, widely addressed in the packet scheduling in the network side [13], [27], deserve further studies in providing queueing-delay differentiation in streaming services.

Fig. 10b shows the streaming bit rate of request classes generated by a FCFS/FF scheduler and a priority scheduler in combination with the harmonic proportional allocation scheme. It shows that various scheduling schemes generate marginally different results by the use of the same bandwidth allocation scheme. Requests from higher classes have higher probabilities to be scheduled by the priority scheduler than by the FCFS/FF scheduler. Hence, their arrival rates calculated by the priority scheduler are less than those calculated by the FCFS/FF scheduler. This leads to higher streaming bit rate for requests from higher classes.

In summary, we found that the strict priority scheduler can differentiate queueing delay between different request classes. However, it cannot quantitatively guarantee quality spacings between various classes.

6.4 Impact of the Feedback Queue

Figs. 7 and 10a illustrate the queueing-delay dip scenarios. That is, the queueing delay increases and then marginally decreases and significantly increases again as the arrival rate varies within a certain range. As we discussed above, it is due to the impact of the backlogged requests on the bandwidth allocation calculations, together with the impact of the variance of interarrival time distributions. In this experiment, we investigate the impact of the feedback queue technique on mitigating the queueing-delay dip scenarios.



(a) (b)

Fig. 10. Impact of request schedulers with the harmonic proportional allocation (a) on queueing delay of request classes and (b) on differentiated streaming bit rates.

Fig. 11 shows the impact of the feedback queue technique on queueing delay due to the harmonic proportional allocation with the FCFS/FF scheduler and the strict priority scheduler, respectively. Evidently, the feedback queue technique can reduce queueing delay significantly because it reduces the impact of the variance of interarrival time distributions on the calculation of arrival rate of request classes. As all backlogged requests in the listen queues are included in the calculation of the arrival rates, the queueing-delay dip scenarios almost disappear. Fig. 11a shows that by the use of FCFS/FF scheduler with the feedback queue, requests from all classes receive lower queueing delays. Fig. 11b show that the priority scheduler favors requests from higher classes, which almost have no queueing delay.

Fig. 12a shows more sensitivity analyses of the impact of the feedback queue on queueing delay, as the scaling parameter α increases. Parameter α indicates the percentage of the number of backlogged request in a queue to be included in the calculation of the arrival rate. The results are due to the harmonic proportional allocation with the FCFS/FF scheduler. It is shown that the average queueing-delay of the classes decreases as parameter α increases. As the percentage of the number of backlogged requests included in the calculation of the arrival rates is increased, the mismatch between the estimated arrival rates and real arrival rates decreases. This reduces the queueing-delay of requests and mitigates the queueing-delay dip scenarios.

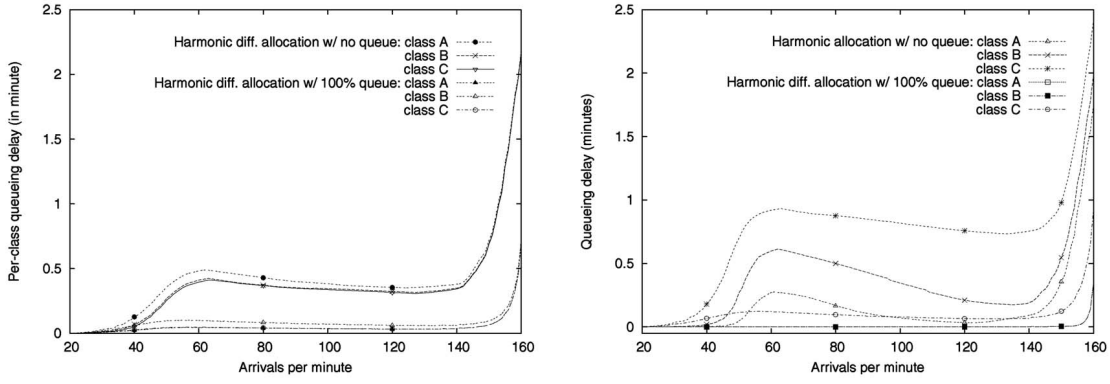
Fig. 12b shows the impact of the feedback queue technique with the harmonic proportional allocation and FCFS/FF scheduler on the streaming bit rate of request classes. When the feedback queue technique is used, there are marginal differences between streaming bit rates of class A only during light-load periods. The priority scheduler generated similar results, which are omitted.

In summary, the results show that the feedback queue technique can reduce queueing delay of requests and mitigate the queueing-delay dip scenarios significantly at a marginal cost of slightly lower streaming bit rates.

6.5 Impact of Request Arrival Ratio and Lower Bound of Streaming Bit Rate

In the previous experiments, the request arrival ratios of the three classes $(\lambda_a, \lambda_b, \lambda_c)$ were fixed to (1, 4, 5). The lower bounds of streaming bit rate for the three classes (L_a, L_b, L_c) were fixed equal to (1, 1, 1) Mbps. In this section, we examine the impact of these parameters on differentiation performance. In the experiments, the harmonic proportional allocation scheme was adopted.

Fig. 13 shows a microscopic view of the streaming bit rate of requests due to various request arrival ratios among the three classes. We varied the request arrival ratios $(\lambda_a, \lambda_b, \lambda_c)$ from (1, 4, 5) to (2, 3, 5), (5, 5, 10), and (3, 2, 5),



(a) (b)

Fig. 11. Impact of the feedback queue on queueing delay by the use of various schedulers. (a) FCFS/FF scheduling. (b) Priority scheduling.

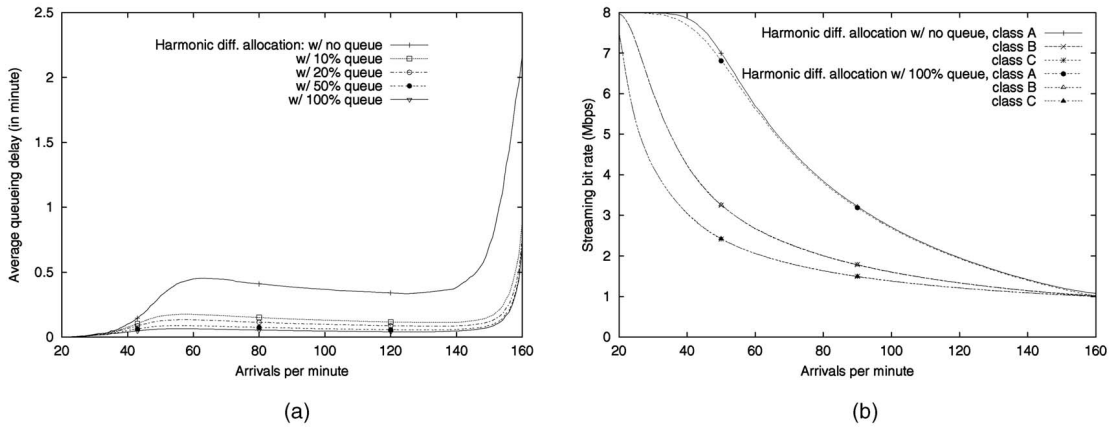


Fig. 12. Impact of the feedback queue with the harmonic allocation and FCFS/FF scheduler (a) on queueing delay and (b) on streaming bit rates.

respectively. The aggregate arrival rate of the three classes was kept to be same at 50 requests/minute. That is, the load of class C did not change. The differentiation weight of the three classes $(\delta_a, \delta_b, \delta_c)$ were fixed equal to (4, 2, 1). The simulation at each request arrival ratio was run and recorded for 40 minutes. It can be seen that, as a fraction of the class B load shifts to class A, the quality factor of class B increases, while that of class A and C decreases. This verifies the fourth property of the controllability and dynamics achieved by the allocation scheme (10).

Fig. 14 shows the average streaming bit rate of the three classes. The aggregate arrival rate varied from 50 to 80 requests/minute. We varied the lower bounds of streaming bit rate for the classes (L_a, L_b, L_c) from (1, 1, 1) Mbps to (2, 1.5, 1) Mbps and (4, 2, 1) Mbps, respectively. All three classes benefit from the lower bound promotions. However, note that these higher streaming bit rates are achieved at the cost of lower aggregate arrival rates. When the lower bounds were (4, 2, 1) Mbps, the maximum aggregate arrival rate supported by the streaming system was about 94 requests/minute before the rejection occurs, the same as the results of the absolute differentiation allocation illustrated in Fig. 2.

We also performed a wide range of sensitivity analyses. We varied the video duration, the server outgoing bandwidth, the maximum acceptable waiting time, and the differentiation weight ratio. While we do not have space to present all of the results, we note that we did not reach any significantly different conclusions regarding to the differentiation predictability, controllability, and fairness achieved by the proposed bandwidth allocation schemes.

7 Conclusions and Future Work

Recent advances of real-time transcoding technology make it possible for streaming servers to manage their limited network-I/O bandwidth and control the quality spacing between different request classes at a fine-grained level. In this article, we investigated the problem of differentiated bandwidth allocation for the delivery of high bit rate streams to high priority classes without overcompromising low priority classes on streaming servers. We formulated the bandwidth allocation problem as an optimization of a harmonic system utility function and derived the optimal streaming bit rates under various server load conditions. We proved that the optimal allocation scheme, referred to harmonic proportional allocation, guarantees proportional sharing in terms of streaming bit rate between classes with different differentiation weights. We also tailored a proportional-share bandwidth allocation scheme. We evaluated the bandwidth allocation schemes via extensive simulations. Simulation results showed that

1. The harmonic allocation and proportional-share bandwidth allocation schemes can enable the streaming servers to efficiently and adaptively manage their network-I/O bandwidth and, hence, achieve high service availability and maintain low queueing-delay when the servers are heavily loaded.

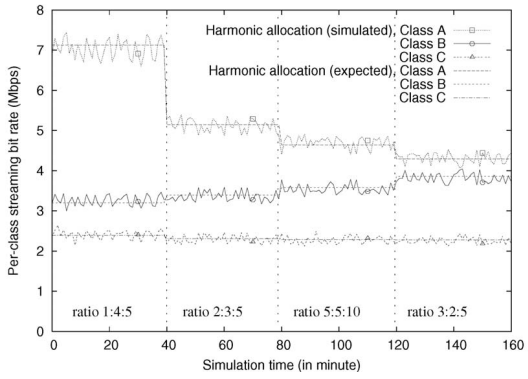


Fig. 13. Impact of request arrival ratios on the differentiation.

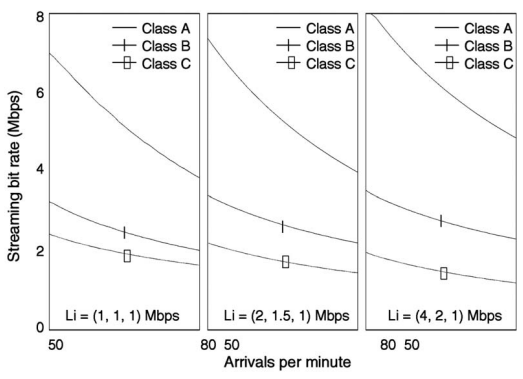


Fig. 14. Impact of lower bounds of streaming bit rate.

2. The harmonic bandwidth allocation scheme with respect to the overall system utility function can achieve the objective of proportional sharing in both short and long timescales.
3. The harmonic bandwidth allocation scheme, in combination with the strict priority request scheduling, provides certain degrees of control over the queueing-delay differentiation of the request classes, although no quality spacing between the classes is quantitatively controlled.
4. The bandwidth allocation schemes are based on estimates of request arrival rates of different classes in moving windows. The feedback queue technique increases the estimation accuracy and helps reduce the allocation variations caused by bursty traffic.

We note that this article considered no CPU overhead for online transcoding. As a matter of fact, the transcoding technology enables efficient utilization of network-I/O bandwidth at the cost of CPU cycles. The cost modeling of continuous media transcoding is still an open research issue. Given the cost model of transcoding, it is important to make a trade off between the network-I/O bandwidth and CPU power. Second, we addressed the problem of providing differentiated streaming services in the context of CBR encoding scheme. Another popular encoding scheme is VBR, which ensures constant video quality by varying streaming bit rate. Therefore, the problem of transcoding-enabled service differentiation provisioning on streaming servers deserves a further study.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their time and valuable suggestions. This work was supported in part by US National Science Foundation grants ACI-0203592 and CCR-9988266.

REFERENCES

[1] T.F. Abdelzaher, K.G. Shin, and N. Bhatti, "Performance Guarantees for Web Server End-Systems: A Control-Theoretical Approach," *IEEE Trans. Parallel and Distributed Systems,* vol. 13, no. 1, pp. 80-96, Jan. 2002.

[2] C.C. Aggarwal, J.L. Wolf, and P.S. Yu, "The Maximum Factor Queue Length Batching Scheme for Video-on-Demand Systems," *IEEE Trans. Computers,* vol. 50, no. 2, pp. 97-110, Feb. 2001.

[3] E. Amir, S. McCanne, and H. Zhang, "An Application Level Video Gateway," *Proc. ACM Multimedia Conf.,* pp. 255-265, 1995.

[4] S.V. Anastasiadis, K.C. Sevcik, and M. Stumm, "Server-Based Smoothing of Variable Bit-Rate Streams," *Proc. ACM Multimedia Conf.,* pp. 147-158, 2001.

[5] S.V. Anastasiadis, P. Varman, J.S. Vitter, and K. Yi, "Lexicographically Optimal Smoothing for Broadband Traffic Multiplexing," *Proc. ACM Symp. Principles of Distributed Computing,* pp. 68-77, 2002.

[6] G. Banga, P. Druschel, and J. Mogul, "Resource Containers: A New Facility for Resource Management in Server Systems," *Proc. USENIX Symp. Operating System Design and Implementation,* 1999.

[7] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," *IETF RFC 2475,* 1998.

[8] V. Cardellini, E. Casalicchio, M. Colajanni, and M. Mambelli, "Web Switch Support for Differentiated Services," *ACM SIGMETRICS Performance Evaluation Rev.,* vol. 29, no. 2, pp. 14-19, 2001.

[9] S. Chandra, C.S. Ellis, and A. Vahdat, "Application-Level Differentiated Multimedia Web Services Using Quality Aware Transcoding," *IEEE J. Selected Areas in Comm.,* vol. 18, no. 12, pp. 2544-2265, 2000.

[10] S. Chen, B. Shen, S. Wee, and X. Zhang, "Designs of High Quality Streaming Systems," *Proc. IEEE INFOCOM Conf.,* 2004.

[11] X. Chen and P. Mohapatra, "Performance Evaluation of Service Differentiating Internet Servers," *IEEE Trans. Computers,* vol. 51, no. 11, pp. 1368-1375, Nov. 2002.

[12] L. Cherkasova and P. Phaal, "Session-Based Admission Control: A Mechanism for Peak Load Management of Commercial Web Sites," *IEEE Trans. Computers,* vol. 51, no. 6, pp. 669-685, June 2002.

[13] C. Dovrolis, D. Stiliadis, and P. Ramanathan, "Proportional Differentiated Services: Delay Differentiation and Packet Scheduling," *Proc. ACM SIGCOMM Conf.,* 1999.

[14] D. Eager, M. Vernon, and J. Zahorjan, "Minimizing Bandwidth Requirements for On-Demand Data Delivery," *IEEE Trans. Knowledge and Data Eng.,* vol. 13, no. 5, pp. 742-757, 2001.

[15] W.-C. Feng and J. Rexford, "Performance Evaluation of Smoothing Algorithms for Transmitting Prerecorded Variable-Bit-Rate Video," *IEEE Trans. Multimedia,* vol. 1, no. 3, pp. 302-312, 1999.

[16] A. Fox, S.D. Gribble, Y. Chawathe, and E.A. Brewer, "Adapting to Network and Client Variation Using Infrastructural Proxies: Lessons and Perspectives," *IEEE Personal Comm.,* vol. 5, no. 4, pp. 10-19, 1998.

[17] J. Gafsi and E.W. Biersack, "Modeling and Performance Comparison of Reliability Strategies for Distributed Video Servers," *IEEE Trans. Parallel and Distributed Systems,* vol. 11, no. 4, pp. 412-430, Apr. 2000.

[18] L. Golubchik, R.R. Muntz, C. Chou, and S. Berson, "Design of Fault-Tolerant Large-Scale VoD Servers: With Emphasis on High-Performance and Low-Cost," *IEEE Trans. Parallel and Distributed Systems,* vol. 12, no. 4, pp. 363-386, Apr. 2001.

[19] K.A. Hua and S. Sheu, "Skyscraper Broadcasting: A New Broadcasting Scheme for Metropolitan Video-on-Demand Systems," *Proc. ACM SIGCOMM Conf.,* pp. 89-100, 1997.

[20] K.A. Hua, C. Ying, and S. Sheu, "Patching: A Multicast Technique for True Video-on-Demand Systems," *Proc. ACM Multimedia Conf.,* pp. 191-200, 1998.

[21] T. Ibarkai and N. Katoh, *Resource Allocation Problem—Algorithmic Approaches.* MIT Press, 1988.

[22] J. Kangasharju, F. Hartanto, M. Reisslein, and K.W. Ross, "Distributing Layered Encoded Video through Caches," *IEEE Trans. Computers,* vol. 51, no. 6, pp. 622-636, June 2002.

[23] F.P. Kelly, A.K. Maulloo, and D.K.H.T. Tan, "Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability," *J. Operational Research Soc.,* vol. 49, pp. 237-252, 1998.

[24] S.C.M. Lee, J.C.S. Lui, and D.K.Y. Yau, "Admission Control and Dynamic Adaptation for a Proportional-Delay DiffServ-Enabled Web Server," *Proc. ACM SIGMETRICS Conf.,* 2002.

[25] Y.B. Lee and P.C. Wong, "Performance Analysis of a Pull-Based Parallel Video Server," *IEEE Trans. Parallel and Distributed Systems,* vol. 11, no. 12, pp. 1217-1231, Dec. 2000.

[26] W. Leinberger, G. Karypis, and V. Kumar, "Job Scheduling on the Presence of Multiple Resource Requirements," *Proc. SuperComputing Conf.,* 1999.

[27] M.K.H. Leung, J.C.S. Lui, and D.K.Y. Yau, "Adaptive Proportional Delay Differentiated Services: Characterization and Performance Evaluation," *IEEE/ACM Trans. Networking,* vol. 9, no. 6, pp. 908-817, 2001.

[28] C. Poellabauer, K. Schwan, and R. West, "Coordinated CPU and Event Scheduling for Distributed Multimedia Applications," *Proc. Ninth ACM Multimedia Conf.,* 2001.

[29] R. Puri, K.W. Lee, K. Ramchandran, and V. Bharghavan, "An Integrated Source Transcoding and Congestion Control Paradigm for Video Streaming in the Internet," *IEEE Trans. Multimedia,* vol. 3, no. 1, pp. 18-32, 2001.

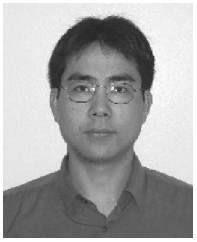
[30] R. Rajkumar, C. Lee, J. Lehoczky, and D. Siewiorek, "A Resource Allocation Model for QoS Management," *Proc. 19th IEEE Real-Time Systems Symp. (RTSS),* pp. 298-307, 1997.

[31] D. Rosu, K. Schwan, and S. Yalamanchili, "FARA: A Framework for Adaptive Resource Allocation in Complex Real-Time Systems," *Proc. IEEE Real-Time Technology and Applications Symp.,* 1998.

[32] P.J. Shenoy, P. Goyal, S. Rao, and H.M. Vin, "Symphony: An Integrated Multimedia File System," *Proc. ACM/SPIE Multimedia Computing and Networking Conf.,* pp. 124-138, 1998.

[33] S. Sheu, K.A. Hua, and W. Tavanapong, "Chaining: A Generalized Batching Technique for Video-on-Demand Systems," *Proc. Int'l Conf. Multimedia Computing and Systems (ICMCS),* pp. 110-117, 1997.

[34] V. Sundaram, A. Chandra, P. Goyal, P.J. Shenoy, J. Sahni, and H.M. Vin, "Application Performance in the QLinux Multimedia Operating System," *Proc. Eighth ACM Multimedia Conf.,* pp. 127-136, 2000.
[35] K. Trivedi, *Probability and Statistics with Reliability, Queueing, and Computer Science Applications.* Englewood Cliffs, N.J.: Prentice Hall, 1982.
[36] C.A. Waldspurger and W.E. Weihl, "Lottery Scheduling: Flexible Proportional-Share Resource Management," *Proc. First USENIX Symp. Operating System Design and Implementation,* 1994.
[37] J.L. Wolf and P.S. Yu, "On Balancing the Load in a Clustered Web Farm," *ACM Trans. Internet Technology,* vol. 1, no. 2, pp. 231-261, 2001.
[38] M. Wu, R.A. Joyce, H.-S. Wong, L. Guan, and S.-Y. Kung, "Dynamic Resource Allocation via Video Content and Short-Term Traffic Statistics," *IEEE Trans. Multimedia,* vol. 3, no. 2, pp. 186-199, 2001.
[39] J. Yang, "Deliver Multimedia Streams with Flexible QOS via a Multicast DAG," *Proc. IEEE 23rd Int'l Conf. Distributed Computing Systems (ICDCS),* 2003.
[40] J. Youn, M.T. Sun, and C.W. Lin, "Motion Vector Refinement for High-Performance Transcoding," *IEEE Trans. Multimedia,* vol. 1, no. 1, pp. 30-40, 1999.
[41] Z. Zhang, Y. Wang, D.H.C. Du, and D. Su, "Video Staging: A Proxy-Server-Based Approach to End-to-End Video Delivery over Wide-Area Networks," *IEEE/ACM Trans. Networking,* vol. 8, no. 4, pp. 429-442, 2000.
[42] X. Zhou, J. Wei, and C.-Z. Xu, "Modeling and Analysis of 2D Service Differentiation on e-Commerce Servers," *Proc. IEEE 24th Int'l Conf. Distributed Computing Systems (ICDCS),* Mar. 2004.
[43] X. Zhou, J. Wei, and C.-Z. Xu, "Processing Rate Allocation for Proportional Slowdown Differentiation on Internet Servers," *Proc. IEEE 18th Int'l Parallel and Distributed Processing Symp. (IPDPS),* Apr. 2004.
[44] X. Zhou and C.-Z. Xu, "Optimal Video Replication and Placement on a Cluster of Video-on-Demand Servers," *Proc. IEEE 31st Int'l Conf. Parallel Processing (ICPP),* pp. 547-555, 2002.
[45] H. Zhu, H. Tang, and T. Yang, "Demand-Driven Service Differentiation for Cluster-Based Network Servers," *Proc. IEEE INFOCOM Conf.,* pp. 679-688, 2001.



Xiaobo Zhou received the BS, MS, and PhD degrees in computer science from Nanjing University, China, in 1994, 1997, and 2000, respectively. He is an assistant professor in the Department of Computer Science, University of Colorado at Colorado Springs. His research interests are in distributed and Internet computing systems, broadband multimedia applications, and networking security. He has authored/coauthored approximately 30 papers in these areas. He was a visiting scientist in 1999 and a postdoctorate research associate in 2000 at Paderborn Center for Parallel Computing, University of Paderborn, Germany. From January 2001 to August 2003, he was a visiting assistant professor in the Department of Computer Science, Wayne State University, Detroit. He is a member of the IEEE Computer Society.

Cheng-Zhong Xu received the BS and MS degrees in computer science from Nanjing University in 1986 and 1989, respectively, and the PhD degree in computer science from the University of Hong Kong in 1993. He is an associate professor in the Department of Electrical and Computer Engineering of Wayne State University, Detroit, Michigan. Dr. Xu's research interests are in distributed and parallel systems, particularly in resource management for high performance cluster and grid computing and scalable and secure Internet services. He has published more than 60 papers in archived journals and peer-reviewed conference proceedings in these areas. He is the leading coauthor of the book *Load Balancing in Parallel Computers: Theory and Practice* (Kluwer Academic, 1997). He has served on technical program committees of numerous international conferences, including the most recent IEEE ICDCS'04, ICPP'04, and PDCS'04. He was a guest coeditor of *Journal of Parallel and Distributed Computing* for scalable Internet service and architecture. Dr. Xu's research was supported in part by the US National Science Foundation, NASA, and Cray Research. He is a recipient of the Faculty Research Award of Wayne State University in 2000, President's Award for Excellence in Teaching in 2002, and Career Development Chair Award in 2003. He is a senior member of IEEE.