Bit-Rate Allocation for Multiple Video Streams Using a Pricing-Based Mechanism

Mayank Tiwari, *Member, IEEE*, Theodore Groves, and Pamela Cosman, *Fellow, IEEE*

***Abstract*—We consider the problem of bit-rate allocation for multiple video users sharing a common transmission channel. Previously, overall quality of multiple users was improved by exploiting relative video complexity. Users with high-complexity video benefit at the expense of video quality reduction for other users with simpler videos. The quality of all users can be improved by collectively allocating the bit rate in a centralized fashion which requires sharing video information with a central controller. In this paper, we present an informationally decentralized bit-rate allocation for multiple users where a user only needs to inform his demand to an allocator. Each user separately calculates his bit-rate demand based on his video complexity and bit-rate price, where the bit-rate price is announced by the allocator. The allocator adjusts the bit-rate price for the next period based on the bit rate demanded by the users and the total available bit-rate supply. Simulation results show that all users improve their quality by the pricing-based decentralized bit-rate allocation method compared with their allocation when acting individually. The results of our proposed method are comparable to the centralized bit-rate allocation.**

***Index Terms*—Decentralized allocation, H.264/AVC, rate control, rate–distortion optimization, video compression.**

I. Introduction

THE growth in simultaneous video transmission over communication channels by multiple users has stimulated efforts to better allocate shared resources such as bit rate among users. Instead of equally dividing available bit rate among videos, a number of joint bit-rate allocation algorithms have been proposed to improve the overall video quality [1]–[4]. However, the overall quality improvement comes at the expense of lowering the quality of some of the videos. The improvement is achieved by reallocating bits in every time period (or slot) from videos whose quality suffers least from reducing their allocated bit rate to those videos benefitting most from an increase in allocated bit rate.

In [5], we proposed a joint bit-rate allocation scheme based on competitive equilibrium theory that improves the quality of all videos. The quality improvement was achieved by reallocating the bits for each video from those time slots when a reduction in bit rate hurts little to other slots when increased bit rate increases quality the most. The method in [5] allocated bits among video streams at each slot and within a video stream across slots. This is possible if there are many videos, some of whose quality can be improved by reducing their allocation in one slot for an increased allocation in some other (later) slot, and other videos in the same slot whose quality could be improved by the reverse exchange. The method described in [5] is a centralized bit-rate allocation method where all of the users send their true rate–distortion (RD) information to a central controller who, in turn, decides the bit rate allocated to each user at each slot.

Implementation of these schemes requires communication of specific information about individual videos at every slot, namely, their RD curve, or the rate at which quality increases as more bits are received. This complicated information must be communicated accurately. While the amount of information can be reduced by fitting the RD curve and sending only the curve-fit parameters, the fit could be poor for some video segments. Also, the approach is problematic if some users, for certain time periods, want to employ other criteria besides RD curves to determine their allocation (e.g., a video can be, during a certain time slot, high motion with a demanding RD curve and yet can be deemed unimportant by the user during that slot) but the centralized system would need greater information transfer to allow for this type of flexibility in the allocation criteria.

Various decentralized algorithms have been proposed [6]–[8] for joint bit-rate allocation for multiple video streams. An auction mechanism was used in [6] to allocate bit rate in a cross-layer optimization. A distributed bit-rate allocation was proposed in [7] to minimize the total mean squared error (MSE) of all of the videos, but suffers from high price fluctuations and the fact that not all videos will improve their video quality. Nash bargaining solution (NBS) and Kalai–Smorodinsky bargaining solution (KSBS) approaches were proposed in [8] for multi-user resource allocation where the NBS was used to maximize the overall system utility, and the KSBS was used to ensure that all users incur the same utility penalty relative to the maximum achievable utility. However, in [8], the initial utility was assumed to be zero, and all of the available resources were not allocated initially. If all of the available resources were allocated initially in [8], then the bargaining solution reallocation would cause some users to suffer reduced utility compared with the initial allocation.

A pricing mechanism for multiuser resource allocation in wireless multimedia applications was discussed in [9]. A method similar to [8], the initial utility was assumed to be zero and the

Manuscript received February 03, 2010; revised October 06, 2010; accepted March 22, 2011. Date of publication April 21, 2011; date of current version October 19, 2011. This work was supported by the National Science Foundation and the Office of Naval Research. Part of this work appeared in the IEEE International Conference on Image Processing, November 2009.

M. Tiwari was with the Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, CA 92093–0407 USA. He is now with Qualcomm Inc., San Diego, CA 92121 USA.

T. Groves is with the Department of Economics, University of California San Diego, La Jolla, CA 92093-0508 USA.

P. Cosman is with the Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, CA 92093-0407 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2011.2146262

total utility achieved at the final allocation was shown as the total improvement. Again, if all of the available resources were allocated initially, then the method described in [9] would result in a decrease in utility for some users. Moreover, in [9], increments in the video quality were assumed to be of equal value at any quality level to calculate a penalty for a strategic user, which is a very strong and unrealistic assumption. Generally, the value of a unit increase in video quality is different at different video quality levels, unlike the linear behavior assumed in [9].

Rate control using pricing mechanisms has been extensively studied for communication networks [10]–[16]. Using a pricing mechanism in [10], the proportional fairness criterion was implemented. Such methods are specifically designed for networking problems where link prices, routing, and proportional fairness are of importance in the medium-access control layer. These algorithms, however, do not take into account the characteristics of the source, and applying such algorithms for video transmission may not result in improving the quality of all of the video streams.

Compared with [5], we propose a scheme which requires simpler information to be exchanged and which does not require a heavy computational burden on a central controller. This scheme is modeled on price-guided procedures discussed in the economics literature [17] that are characterized as decentralized, as various video transmitters (users) only communicate their bit-rate demands in response to the bit-rate price announced by a bit-rate allocator in a slot. By contrast, in a centralized procedure (e.g., [5]), each user communicates the private information that is necessary for the bit-rate calculation (e.g., rate–distortion curves) and the allocator decides on an allocation for all of the users. In our decentralized procedure, the allocator adjusts the user's demands to equalize the aggregate allocation to the available supply and announces the price for the next slot. With this price-guided allocation scheme, instead of using bits at a constant rate, users will increase their demand in slots during which their videos are more complex (e.g., high motion) and reduce their demand in slots of low complexity. Permitting the amount of bit rate used in each slot to vary increases the efficiency of each user's total bit rate use by giving more of the resource when it is most valuable (in terms of lowering MSE) and less when it is less valuable. The use of a price to guide users' choices of demand reflects the relative scarcity of available bit rate in each slot. When all users request more bits than the average, scarcity is greater and the price is higher, thus moderating the demands. Our simulation results show that each user benefits from this price-based decentralized bit-rate allocation mechanism compared with the equal bit-rate allocation to all users. The performance of this algorithm is comparable to the centralized bit-rate allocation introduced in [5], where all users send their RD curves to the allocator. The equilibrium price is not achieved in this price-based decentralized allocation method because only one iteration for price adjustment is made. However, the simulations show that the bit-rate price in this method closely follows the equilibrium price. In the centralized approach, the computations grow exponentially with the number of users and the central controller is responsible for all of the computations. In the decentralized approach, the computational complexity remain constant for the allocator and is independent of the number of users. Each user makes the bit-rate demand calculations independently.

An elementary version of pricing-based decentralized bit-rate allocation was introduced in [18]. In this paper, we extend [18] to include a delay buffer, iterative pricing, discussion on price manipulating users, and variable start and end times for users.

The remainder of this paper is organized as follows. Section II provides the general description of the pricing-based decentralized bit-rate allocation process for individual users. Section III discusses various aspects of the bit-rate allocation process in detail. Simulation results are given in Section IV, and Section V concludes the paper.

II. Pricing-Based Decentralized Bit-Rate Allocation

Suppose there are N video users sharing the available bit rate. The video stream of each user is divided into slots. In this work, we consider one slot to be one group of pictures (GOP), but a slot can be larger or smaller than a GOP. We will use the terms GOP and slot interchangeably. We assume that the videos are synchronized at the GOP level. Such synchronization can be achieved by a small amount of buffering of the input videos at the expense of a small amount of delay. For different GOP sizes, the synchronization can be achieved by the formation of a "Super GOP" as described in [2], where a super GOP is the least common multiple of GOPs for all of the users. The problem of synchronization does not exist when a slot is of a frame size if all of the videos are transmitted at the same frame rate.

For user n, the video stream starts at slot t_n and ends at slot T_n. The entire system time is set on the basis of the start and end time of all of the video streams. The system time starts at T_s when the first user enters the system (we assume the time axis is shifted so that $T_s = 1$) and the system time ends when the last user exits the system at time $T = max(T_n)$.

The utility of user n at slot t, denoted by $U_{n,t}(x_{n,t})$, is taken to be the negative of his MSE at a bit rate of $x_{n,t}$, given by the RD curve for that slot. Although our simulations took utility to be the negative of MSE, our results will hold qualitatively for any sufficiently smooth monotonic decreasing convex utility function of distortion. A user's goal is to maximize his utility or, equivalently, minimize his total MSE, given his resources, across all slots.

At time t, let $M_{n,t}$ be the available money for user n and p_t be the bit-rate price. Upon entering the system, each user is allocated a specific amount of credit—called "money"—when he starts sending his video. The initial allocation is based on the (expected) length of time of his video and the (expected) average channel bit rate (r_n) over this time span, valued at the (normalized or expected) average price $(\bar{p}_{t_n})$. The initial allocation of money for user n is

$$M_{n,t_n} = (T_n - t_n + 1) \cdot r_n \cdot \bar{p}_{t_n}, \qquad \forall\, n = 1 \ldots N. \quad (1)$$

The utility optimization problem for user n over all his slots is given by

$$\max_{\{x_{n,t}\}} \sum_{t=t_n}^{T_n} U_{n,t}(x_{n,t}) \quad (2)$$

subject to the constraint

$$\sum_{t=t_n}^{T_n} p_t \cdot x_{n,t} \leq M_{n,t_n}. \tag{3}$$

The constraint in (3) requires the money spent over all slots to be less than or equal to the total allocated money. Solving (2) under the constraint of (3) gives the optimal demand for each user in all slots. The bit-rate prices (p_t) in (3) are ideally the equilibrium prices which would equate the total demand in each slot with the total supply. However, for a real-time problem, the RD functions for future slots are unknown, as are the future prices which depend on all video users' (unknown) RD functions in the future. To address this informational limitation, we consider a sequential process. In each slot, a user will reoptimize his decision for the current and all future slots using expected values for future prices and RD functions. If the future slots are identical in expectation (for example, the future environment is perceived as stationary), then the decision problem in each slot is just an optimization problem with two decisions only—the demand $x_{n,t}$ for the current slot and $\bar{x}_{n,t}$, the common demand for each of the remaining $(T_n - t)$ slots. Given the price at slot t and an expected price $\bar{p}_t$ for the future expected RD function, the new optimization problem becomes

$$\begin{aligned} &\max_{x_{n,t},\bar{x}_{n,t}} \left[U_{n,t}(x_{n,t}) + (T_n - t) \cdot \bar{U}_{n,t}(\bar{x}_{n,t})\right] \\ &\text{s.t. } p_t \cdot x_{n,t} + (T_n - t) \cdot \bar{p}_t \cdot \bar{x}_{n,t} \leq M_{n,t} \end{aligned} \tag{4}$$

where $\bar{U}_{n,t}(\bar{x}_{n,t})$ is the future average utility for user n at slot t.

User n at slot t thus makes a demand of $x^*_{n,t}$ bits, where $x^*_{n,t}$ is the solution of (4). This information is sent to the allocator. Based on the demand from each user, the allocator makes a decision on the number of bits to be allocated to each user.

A. Outline of the Pricing-Based Decentralized Bit-Rate Allocation Algorithm

The outline of our pricing-based decentralized rate allocation algorithm is given in this section. The details of the algorithm are discussed in the next section.

1) **Initial conditions**: Initially, the allocator announces a bit-rate price (p_1) at slot 1. Without loss of generality, we set $p_1 = 1$. Each user is allocated an initial amount of money as given by (1) at the start of his video.
2) **Bit-rate demand**: Each user knows the bit-rate price (p_t) and their own utility function $(U_{n,t}(x_{n,t}))$ for the current slot. Each user also estimates an average future price $(\bar{p}_t)$ and future average utility function $(\bar{U}_{n,t}(\bar{x}_{n,t}))$. The estimation of future average price is explained in Section III-C, and methods for estimating average utility functions for the remaining future slots are given in Section III-A. Using (4), a user calculates his bit-rate demand, $x^*_{n,t}$, which is then transmitted to the allocator. Details of this calculation are in Section III-B.
3) **Bit-rate and price adjustment**: As the sum of the demands from all of the users might not equal the available supply, the allocator needs to normalize the demands to the available supply. Possible approaches are given in Section III-D. The allocator also determines the new bit-rate price for the next slot based on the difference between supply and demand for the current slot. Details of this calculation are also in Section III-D.
4) **Available money**: When the allocation is received by the users, they encode their video at the allocated bit rate and transmit through the shared channel. The users then recalculate their total available money for the next slot by subtracting the amount of money just spent. Details of this calculation can be found in Section III-E.
5) Steps 2)–4) are repeated for each slot until all of the videos are transmitted.

III. Bit-Rate Allocation for Multiple Video Streams

Here, we discuss in detail the steps involved in the pricing-based decentralized bit-rate allocation.

A. Estimating the Average RD Function for the Future Slots

Bit-rate allocation for multiple video streams, as given by (4), requires the estimation of average RD characteristics for future slots. We considered several estimation approaches. They differ in the amount of information a user holds at the time of making the forecast. These methods were described in [5] and are briefly discussed below.

Let $D_{n,t}(r_{n,t})$ denote the MSE distortion at rate $r_{n,t}$. We approximate the RD curve using

$$D_{n,t}(r_{n,t}) = a_{n,t} + \frac{b_{n,t}}{r_{n,t} + d_{n,t}} \tag{5}$$

where $a_{n,t}$, $b_{n,t}$, and $d_{n,t}$ are curve-fitting coefficients for user n at slot t and are determined numerically. This model is widely used for video RD curves [8], [9], [19]. Other curve-fitting models with reduced complexity can be used, for example [20]. The utility of a user is defined by the negative sum of his MSE at any slot $(U_{n,t}(x_{n,t}) = -D_{n,t}(x_{n,t}))$. We examine the following two methods to estimate the future average RD curves.

1) **PRE**: The average future RD curve is estimated by averaging the past RD curves. Specifically, we take the average of the individual curve fitting coefficients (a, b, and d) separately for a user over all past slots (t_n to $t-1$, for user n). This is an *ex post* model where users have no information about the future slots, as in the real-time case. Generally, if averaged over a sufficiently long period of time, the complexity of most video streams can be assumed to be almost stationary, and the average RD function of past slots will be a good approximation model for the average RD function for future slots.
2) **REM**: This *ex ante* approximation model assumes each user knows the approximate average RD function for his video over the *remaining* slots ($t+1$ to T_n, for user n). This assumption would hold for archival video. The curve is obtained by averaging the individual curve fitting coefficients (a, b, and d) separately for a user over all of the remaining slots. Empirically averaging the actual curves and applying a standard curve fitting technique, the estimated coefficients are extremely close to those from averaging the coefficients individually.

We expect REM to perform better than PRE because future video information is used in REM whereas no future information is used in PRE.

B. Bit Rate Demanded by the User

Using (4) and (5), we get the user's per slot decision problem

$$\max_{x_{n,t},\bar{x}_{n,t}} -\left(a_{n,t}+\frac{b_{n,t}}{x_{n,t}+d_{n,t}}\right) - (T_n-t)\cdot\left(\bar{a}_{n,t}+\frac{\bar{b}_{n,t}}{\bar{x}_{n,t}+\bar{d}_{n,t}}\right)$$
$$\text{s.t. } p_t\cdot x_{n,t}+(T_n-t)\cdot\bar{p}_t\cdot\bar{x}_{n,t}\le M_{n,t}\quad \forall n=1\ldots N \quad (6)$$

where $\bar{a}_{n,t}+(\bar{b}_{n,t}/(\bar{x}_{n,t}+\bar{d}_{n,t}))$ is the predicted average RD function for user n for all future slots ($t+1$ to T_n). At any slot, a user tries to reduce his sum of MSE for the current slot and estimated MSE for all of the future slots given the bit-rate price for the current slot and the expected bit-rate price for all future slots.

We solve (6) using a Lagrange multiplier approach [21] for each user separately. All of the users calculate their bit-rate demand for the current slot. The bit-rate demand for user n in slot t is given by

$$x^*_{n,t}=\sqrt{\frac{b_{n,t}}{p_t}}\cdot\frac{M_{n,t}+p_t\cdot d_{n,t}+(T_n-t)\cdot\bar{p}_t\cdot\bar{d}_{n,t}}{\sqrt{p_t.b_{n,t}}+(T_n-t)\cdot\sqrt{\bar{p}_t\cdot\bar{b}_{n,t}}}-d_{n,t}\quad \forall n=1\ldots N \quad (7)$$

where $x^*_{n,t}$ is the only information that is conveyed to the allocator by the user.

C. Normalization of $\bar{p}_t$

The parameter $(\bar{p}_t)$ represents the average price at all future slots beyond current time t. This parameter is required to determine the bit-rate demand for the current slot for each user with respect to the average demand at future slots.

Under our stationarity assumption about the future, we can reduce the user's T_n slot problem to a sequence of two-slot problems, given by (4). Additionally, by (1) and (4), a user's budget is homogeneous of degree zero in prices so that a user's optimal demand in the current slot $(x^*_{n,t})$ and average future demand $(\bar{x}_{n,t})$ depend only on the price ratio, $p_t/\bar{p}_t$. Thus, without loss of generality, we normalize the future average price $\bar{p}_t$ to unity, $\bar{p}_t=1$.

D. Bit-Rate Allocation and Price Adjustment by the Allocator

The market clearing price or an equilibrium price is defined as the price at which total demand equals total supply. Since the price p_t will in general not be the equilibrium price, the sum of bit rate demanded by all of the users may differ from the total available bit rate at any slot. Excess demand is defined to be the difference between total bit-rate demand and total bit-rate supply (R_t). The allocator has options to equalize the total demand and supply.

1) **Normalized demand**: The allocator may normalize the individual demands to balance total demand and supply

$$\hat{x}_{n,t}=x^*_{n,t}\cdot\frac{R_t}{\sum_{n=1}^{N}x^*_{n,t}}. \quad (8)$$

The normalized allocations $(\hat{x}_{n,t})$ are sent back to the users who encode their videos using the allocated bit rate. The price for the next slot is adjusted by the allocator based on the current excess demand

$$p_{t+1}=p_t+\alpha_p\cdot\left(\frac{\sum_{n=1}^{N}x^*_{n,t}-R_t}{R_t}\right),\quad p_1=1 \quad (9)$$

where the price-adjustment parameter α_p is a design choice to regulate the price variation, to be discussed later. If aggregate demands are similar from one slot to the next, then a price-adjustment rule based on excess demand will provide the appropriate signal about the relative scarcity of bit rate available next slot.

2) **Iterative pricing**: The price at each slot could, in principle, be iterated until the market clearing price is achieved. The final price achieved by these iterations would be the equilibrium price for that slot.
The initial price for the first iteration of slot t is taken to be the final price achieved at slot $t-1$ (that is, $p_t^{(1)}=p_{t-1}^{(\text{final})}$). For the iterative pricing at each slot, the initial price is announced by the allocator and the users calculate their demand which is given by (4). If the total demand is not equal to the total supply, then the allocator adjusts the price using (10) and the new price is sent back to the users to recalculate their demand. This process is iterated until total demand equals total supply, at which point the market clearing (or equilibrium) price is achieved.
At the i^{th} iteration, the bit-rate price for the next iteration is set as

$$p_t^{(i+1)}=p_t^{(i)}+\delta_p\cdot\left(\frac{\sum_{n=1}^{N}x^{*(i)}_{n,t}-R_t}{R_t}\right) \quad (10)$$

where $x^{*(i)}_{n,t}$ is the bit-rate demand by user n at the ith iteration in slot t. δ_p is the iterative price adjustment parameter and is determined heuristically. It affects the speed of convergence. An iterative pricing method and its convergence issues are discussed in [9].
Ideally, in each slot, several iterations of price and demand messages would be exchanged between the allocator and the users, as given by (10). However, in a real-time process, iteration can become a bottleneck. As we will show, the bit-rate price calculated by the allocator without any iterations follows the competitive equilibrium price closely, and iteration within a slot produces little quality improvement.

3) **Delay buffer**: If iterative pricing is used, each user would be allocated exactly the number of bits demanded. However, with only a single exchange of price and demand information in each slot, the user will generally not receive the exact bit rate demanded. This problem can be solved by using a delay buffer. The size of the delay buffer directly corresponds to the actual time delay (in seconds). For example, for five users at an average bit rate of 100 kbps, the delay buffer size of 500 kb is equivalent to the 1-s delay. The total demand may not equal total supply, but the buffer is drained at the rate of total supply. Any extra bit rate demanded will be stored in the delay buffer which will be

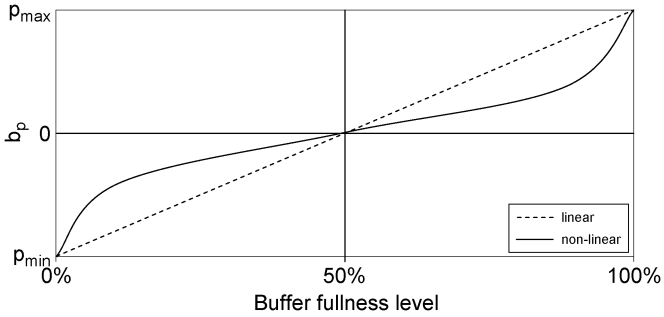

Fig. 1. Additional price adjustment with the buffer fullness level.

drained during those periods when the total demand is less than the total supply. The bit-rate price will still vary as given in (9). In this method, we assume that the delay buffer is arbitrarily large, as needed.

4) **Limited delay buffer and normalized demand**: In any practical scenario, the size of the buffer is limited and predefined. Excess demand is accommodated as long as the delay buffer does not overflow. If, however, the available buffer space is smaller than the excess bit-rate demand, the user's demands are normalized in accordance with the available space. Suppose B_t is the available buffer space at slot t. In case of imminent buffer overflow (i.e., $B_t \leq \sum_{n=1}^{N} x^*_{n,t} - R_t$), the normalization is given by

$$\begin{aligned}\hat{x}_{n,t} &= x^*_{n,t} - \frac{x^*_{n,t}}{\sum_{n=1}^{N} x^*_{n,t}} \cdot \left(\sum_{n=1}^{N} x^*_{n,t} - (R_t + B_t) \right) \\ &= x^*_{n,t} \cdot \frac{(R_t + B_t)}{\sum_{n=1}^{N} x^*_{n,t}}. \end{aligned} \tag{11}$$

However, to reduce the chance of buffer overflow and underflow, we add an extra parameter to the price adjustment function to take into account the level of buffer fullness. We modify (9) for the limited delay buffer case as

$$p_{t+1} = p_t + \alpha_p \cdot \left(\frac{\sum_{n=1}^{N} x^*_{n,t} - R_t}{R_t} \right) + b_p(B_l), \qquad p_1 = 1 \tag{12}$$

where $b_p(B_l)$ is the buffer-level price-adjustment factor which is a function of buffer fullness level (B_l).

In Fig. 1, we show two examples of varying b_p with B_l. In the first, b_p varies linearly with B_l (dotted line). When the buffer fullness is 50%, b_p is 0, which means no adjustment is made to the price based on buffer fullness. When the buffer fullness exceeds 50%, $b_p > 0$ and increases linearly, and when the buffer fullness is less than 50%, then $b_p < 0$ and decreases linearly. The second example shows a monotonic variation of b_p with B_l. Here, if the buffer fullness is around 50%, then the price variation is low. As the buffer fullness approaches 100%, b_p grows rapidly to avoid buffer overflow. If the buffer fullness approaches 0%, b_p is reduced rapidly to avoid buffer underflow. In case of buffer overflow, we normalize the demand as given in (11). In case of an empty buffer, we proportionally increase the demand such that the available bit rate is fully utilized. In both cases, the idea is to increase the bit-rate price when total demand exceeds total supply to discourage the users from demanding more, and to decrease the price when demand is less than supply to encourage more demand.

E. Wealth Adjustment by the User

The users send their bit-rate demands $(x^*_{n,t})$ to the allocator. The allocator sends back the actual bit-rate allocation $(\hat{x}_{n,t})$ as discussed previously. Then, users encode their video streams at the allocated bit rates and transmit over the shared channel. Users reduce their remaining wealth as follows:

$$M_{n,t+1} = M_{n,t} - p_t \cdot \hat{x}_{n,t} \quad \forall n = 1 \ldots N \tag{13}$$

where $\hat{x}_{n,t}$ is the actual allocated bit rate for user n at slot t.

The wealth of a user is reduced at each slot until he transmits all of his video or runs out of money. If a user calculates his optimal demand as given in (7), then the user will always preserve money for the future until all of his video is transmitted.

F. Price Manipulation

As a general proposition, a user could potentially benefit by departing from the honest reporting of his bit-rate demand at any slot. By honest reporting of demand, we mean reporting the $x^*_{n,t}$, which results from (7). By either exaggerating or understating his demand, he can alter both his allocation and expenditure in the current slot and also influence the price he will face next slot. Thus, he could potentially gain more utility at the next slot than he would sacrifice at the current slot by not demanding his $x^*_{n,t}$ bits this slot.

For example, by demanding less than $x^*_{n,t}$ bits in the current slot, a user would receive less bit rate than otherwise, thus lowering his utility this slot. However, in addition to preserving more money to spend for bit rate next slot, the price will be lower than otherwise, because the excess demand will be less. Hence, he may be able to acquire more bit rate next slot. It is not guaranteed that he will experience a net benefit, since every other user will also face the lower price and hence demand more as well. Whether such a strategy actually would pay off in higher utility next slot to offset the loss in utility in the current slot would depend crucially on the demands of other users.

Although for any slot there exists, in general, some deviation from honest reporting of a user's demand that would benefit him, knowing even the direction of the deviation (that is, should the user demand more or less) requires him to know more about the other users' aggregate demand than may be plausible to assume. Additionally, under our model assumptions and specifications, it is a standard economic theory result [22] that the potential utility increase which a user would be able to achieve becomes vanishingly small as the total number of users increases. For this paper we have followed in the tradition of competitive analysis and assumed users ignore any potential influence that their current demand will have on future prices.

IV. Results

Our simulation results were generated using H.264/AVC [23] reference software JM 11.0 [24] with the baseline profile. The test videos were taken from a 72-min travel documentary. The frame rate of each video is 30 frames per second. Each test video is 250 s long (total 7500 frames) at a resolution of 352

$\times$ 240 pixels (SIF). We chose 12 such test streams (denoted g1 to g12). The GOP size is 15 frames (I-P-P-P) and the frames in a GOP are encoded using H.264 rate control [25]. The decentralized rate-allocation method for multiple video streams can be used for any GOP size or structure, frame rate, video length, or resolution. We considered a lossless channel. The quality of a video is reported in terms of average peak signal-to-noise ratio (PSNR). We first calculate the frame-level MSE for any video stream. Then, the MSE is averaged across all frames of a video and converted to PSNR. Each video stream contained various types of scenes with varying camera motions such as zooming and panning. The high-motion scenes included dancing, bike racing, and a vegetable market. The low-motion scenes showed buildings, maps, sculptures, and scenery.

An upper bound on the video quality can be approximated by using the exact RD function for all of the users at all slots. Suppose each user is endowed with some initial wealth. The users are assumed to allocate their wealth at each slot depending on the video complexity. We call this method **FULL**. This method can only be used for archival videos where the RD coefficients are calculated off-line. Each user uses this bit-rate allocation criterion among his slots independently. The bit rate at each slot is adjusted like other methods, as discussed in Section III-D. Note that we assume a constant price at all slots. Therefore, it is not a real upper bound. The real upper bound can be calculated by solving (2) and (3) (for all slots and for all users simultaneously) and computing the market clearing price such that the total demand is equal to the total supply, which is an extremely large computational problem, and the complexity grows with the numbers of users and slots.

In this paper, we compare our multiplexing methods using the pricing-based decentralized bit-rate allocation to the constant rate allocation, **EQUAL**. Here, each slot in a video receives an equal number of bits to encode that segment of video. Note that rate control algorithms, such as [25], used in conjunction with most current video standards strive to achieve equal rate allocation for all GOPs, similar to EQUAL when a slot is of GOP length. This comparison is analogous to current multiplexing practices [1], [2], [8], [9] where the results are compared with the equal bit-rate allocation to all users. In addition, we also compare our method with the MINAVE method described in [1], which improves the average video quality by allocating bits to videos based on their relative complexity.

A. Constant Rate and Constant Number of Users at All Slots

We start with the scenario of bit-rate allocation for multiple video streams where we consider a constant bit-rate (CBR) channel, and all of the users are present at all 500 slots. There is no buffer, and aggregate user demand is normalized to equal the supply at each slot. The price adjustment parameter $\alpha_p = 0.1$. Price fluctuation increases if α_p is large, resulting in a large fluctuation in demand. The price adjustment cannot track excess demand properly if α_p is very small. For our simulations, α_p was not optimized for any set of video streams; it might be possible to improve multiplexing performance by tuning this parameter for the given videos.

The PSNR versus bit-rate results for multiplexing four streams using our decentralized allocation are shown in Fig. 2. There are four curves in each plot, one for each bit-rate allocation method. EQUAL is our baseline case where each video stream receives an equal share of available bits at each slot. The other bit-rate allocation methods outperform EQUAL for all videos.

In the FULL method, each user knows his RD characteristics for all slots in advance. Depending on the video complexity at any slot, a user will make a bit-rate demand. The demands are normalized based on the total available bit rate at that slot. FULL almost always performs the best among all methods. The improvement of FULL over EQUAL varies from 0.62–0.86 dB for g11 to 0.94–1.44 dB for g8. This quality improvement is attributed to not only the varying number of bits at each slot but also the advance knowledge of the video characteristics.

Instead of precise knowledge of RD characteristics at each slot, suppose the users only know the estimated average RD characteristics for the remaining slots. Then, the REM allocation provides quality improvement for all videos. Each user calculates demand at any slot compared to his average demand for the remaining slots. If the video is more complex in the current slot compared to the expected average for the remaining slots, then the user will spend more wealth for the current slot. The results in Fig. 2 show that the quality of all of the videos is improved. Quality improvement with REM varies from 0.60–0.87 dB for g11 to 0.95–1.56 dB for g8. All streams benefit from this multiplexing method and the amount of PSNR improvement depends on the video characteristics. Generally, the improvement is more for the videos whose complexity varies substantially over time. The results from Fig. 2 are consolidated in Table I showing the average PSNR for all of the four videos.

In the real-time video transmission scenario with no input buffering of raw frames other than the current GOP, the users generally have no knowledge of their future video. With PRE bit-rate allocation, the user's average demand for future slots is estimated to equal the average demand for all past slots. PRE bit-rate allocation still improves the quality of all videos. The quality improvement over EQUAL varies from 0.46 to 0.83 dB for g9 to 0.92 to 1.47 dB for g8. This method performs worse than FULL and REM because of the lack of any knowledge about the future video.

In general, the pricing-based decentralized rate-allocation method for multiple video streams improves the quality for each video. The MINAVE method [1] has a different objective: to reduce the *average* MSE across users. MINAVE applied to time-domain RD curves produces average quality of 32.91 dB for the four streams of Fig. 2 at 95 kb per slot per user, which is higher than the average PSNR values for REM and PRE allocation methods. The MINAVE method does better in minimizing the *average* MSE for all users, however, one of the users experiences worse PSNR compared with EQUAL, whereas the others experience better PSNR. In contrast, with our method, all four users are better off compared to EQUAL.

B. Comparison With Centralized Bit-Rate Allocation

Video quality improvement for four of the six video streams when multiplexed together using price-based bit-rate allocation is shown in Fig. 3. These results are similar to the previous

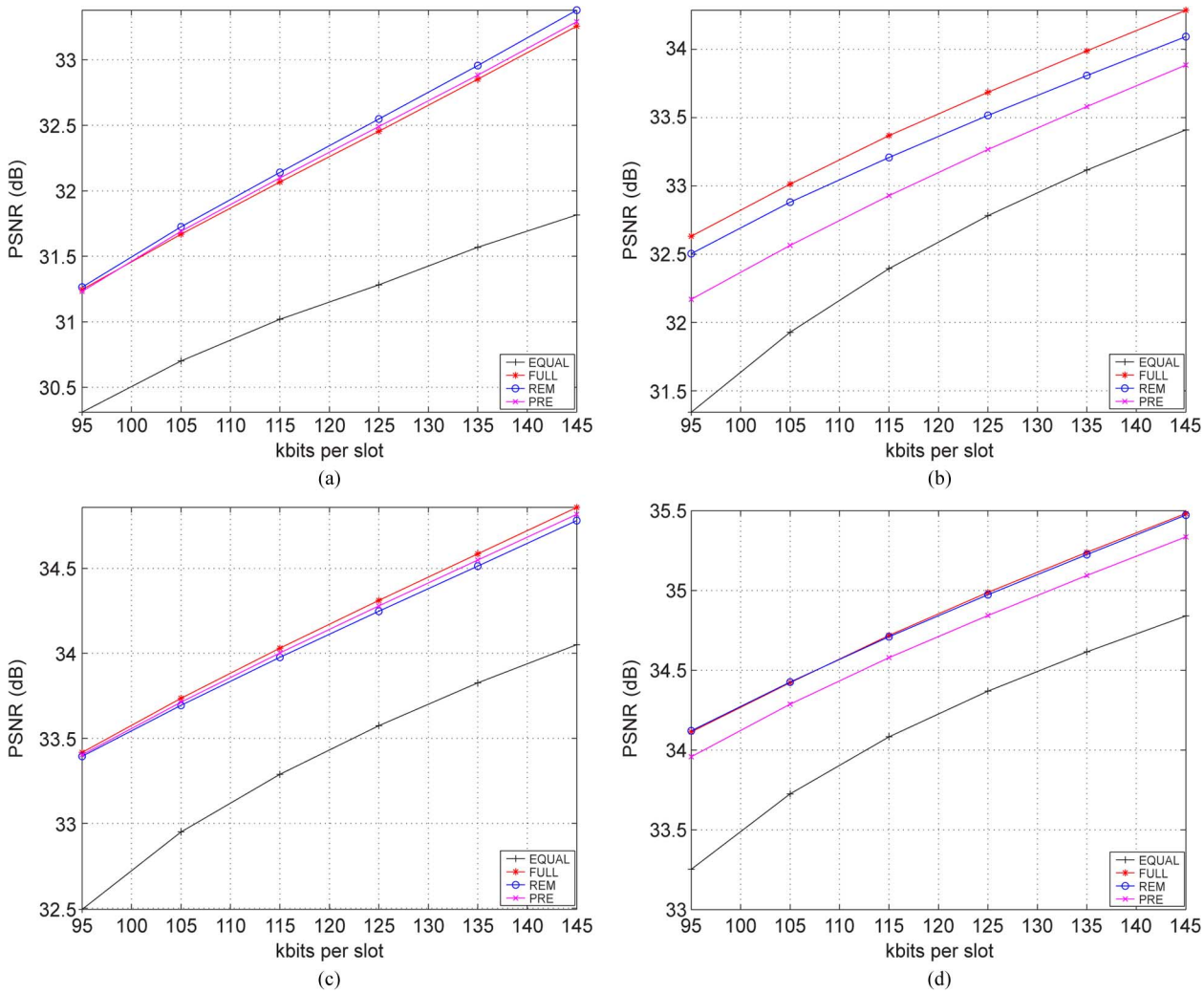

Fig. 2. PSNR performance versus bit rate for four multiplexed video streams. All of the video streams exist at all slots and the bit-rate demand is normalized by the total available supply. (a) g8 video stream. (b) g9 video stream. (c) g10 video stream. (d) g11 video stream.

TABLE I
AVERAGE PSNR (db) FOR ALL OF THE MULTIPLEXING METHODS AT VARIOUS BIT RATES (kb PER USER PER SLOT)

	Bit-rate (kbits per slot per user)					
	95	105	115	125	135	145
EQL	31.85	32.33	32.70	33.00	33.29	33.53
REM	32.82	33.18	33.51	33.82	34.13	34.43
PRE	32.69	33.06	33.40	33.72	34.03	34.34
FUL	32.85	33.22	33.56	33.87	34.18	34.49

case of multiplexing using four video streams. However, due to the increased number of streams, the effect of normalization is less; allocations to users are closer to actual demand. For FULL, the improvement for g8 is from 1.23 to 1.91 dB, much higher than 0.94 to 1.44 dB for the same video when multiplexing four streams. Similarly, REM improves the video quality from 0.62 to 0.92 dB for g11 to 1.21 to 1.98 dB for g8, and PRE improves the video quality from 0.50 to 0.77 dB for g11 to 1.07 to 1.74 dB for g8. All streams improve their quality compared with EQUAL allocation. The quality improvement for a stream increases with the increase in the number of videos multiplexed together.

Fig. 3 also compares the results between decentralized allocation and centralized competitive equilibrium allocation [5]. Generally, centralized allocation performs better because it uses more information; all users send their RD functions, and a central allocator computes the appropriate bit-rate allocation for all users simultaneously. In the decentralized allocation, the allocator has no information about user RD functions; only the bit-rate demand is conveyed. However, as seen in Fig. 3, the improvement from using centralized allocation (REM_central and PRE_central) is only around 0.2–0.3 dB over decentralized allocation for g7, g11, and g12 videos. The performances for the two methods are comparable for g9 and g10 videos. The decentralized allocation performs slightly better for g8.

In general, while centralized allocation slightly outperforms the decentralized allocation, the decentralized method has the advantage of reducing the amount of private information shared by the users, and removing the huge computational burden imposed on the allocator in the centralized approach. The computational complexity grows exponentially with the number of users in the centralized allocation. While efficient suboptimal centralized methods might be found which avoid having the computational burden be exponential in the number of users, the de-

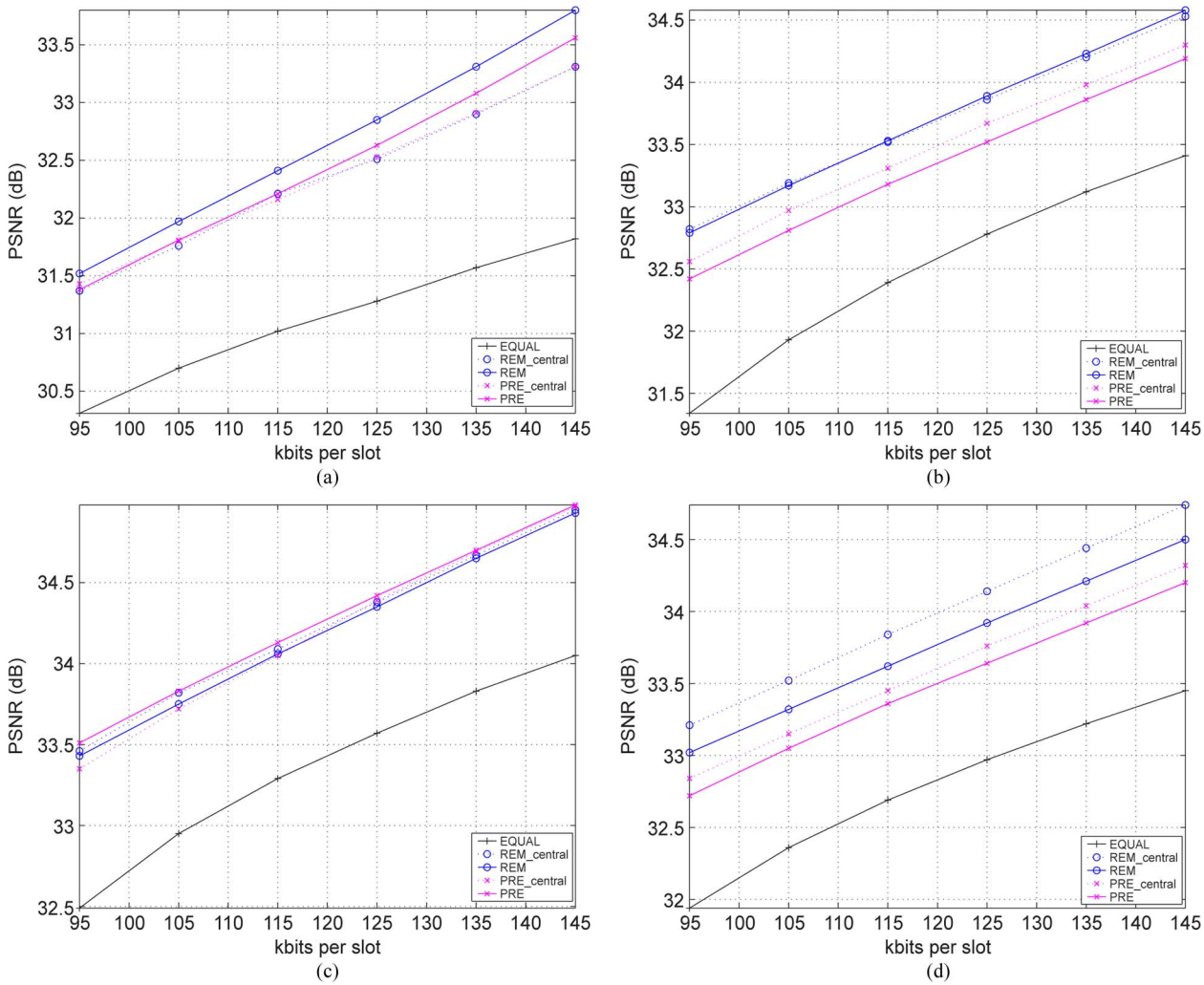

Fig. 3. PSNR performance versus bit rate for four of the six multiplexed video streams for comparing the performance of the proposed method with centralized bit-rate allocation. (a) g8 video stream. (b) g9 video stream. (c) g10 video stream. (d) g12 video stream.

centralized approach reduces the allocator's computation to a trivial normalization, and the calculation performed by each of the users is small and independent of the number of users.

C. Effect of Delay Buffer

In deriving the previous results, there was no delay buffer to store the demanded bit rate that is in excess of total available bit rate at any slot. With a buffer, the performance can improve dramatically. Fig. 4 shows the effect of a buffer on two out of the six multiplexed streams in Fig. 3. On the x-axis is buffer size (in terms of average bit rate per slot per user) and on the y-axis is the PSNR improvement over EQUAL. The videos have the average bit rate of 100 kb per slot per user. The two curves in each plot show the result for the price-based decentralized bit-rate allocation methods. A delay buffer of 0 kb per user (0 s) represents no delay buffer; all demands are normalized by the total available bit rate at each slot. Quality improves as buffer size increases. At any slot, buffer overflow is prevented by normalizing the demand when the buffer is full. For g8, quality improves from 1.10 dB at no delay to 1.55 dB for buffer size of 1100 kb per slot for PRE, and from 1.23 dB at no delay buffer to 1.94 dB at a buffer size of 1300 kb per slot for REM. The quality improvement saturates at a buffer size where all demands are accommodated in the buffer (no need for normalization), so further increase in the buffer size does not increase quality.

D. Iterative Pricing

Although our method for allocating bit rate is based on classical iterative price-guided procedures, we truncate the iterations before convergence to a demand/supply equilibrium. In fact, we only allow one round of price/demand information to be exchanged. The price sent to users is adjusted proportionally to the excess demand in the previous slot. Whether or not this is a clever choice depends on how similar the excess demand (at any price) in the current slot is to the excess demand in the previous slot. As an empirical observation, for our videos, successive aggregate excess demand functions are quite similar, so the price is adjusted in the right direction, and over time the sequence of adjusted prices tracks the sequence of equilibrium prices closely.

For multiplexing four streams, Fig. 5 compares the case when the price is iterated against the case when the price is broadcast only once. REM and PRE (solid curves) represent the normalized demand case when the price is announced only once; these

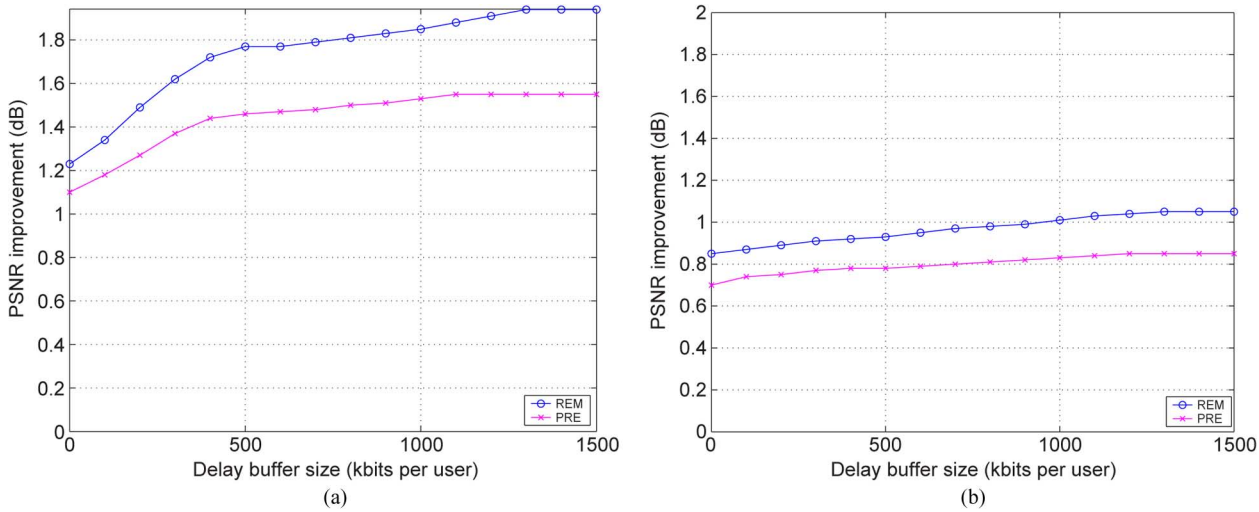

Fig. 4. PSNR improvement with delay buffer size for two video streams from six multiplexed streams at 100 kb per slot per user. (a) g8 video stream. (b) g11 video stream.

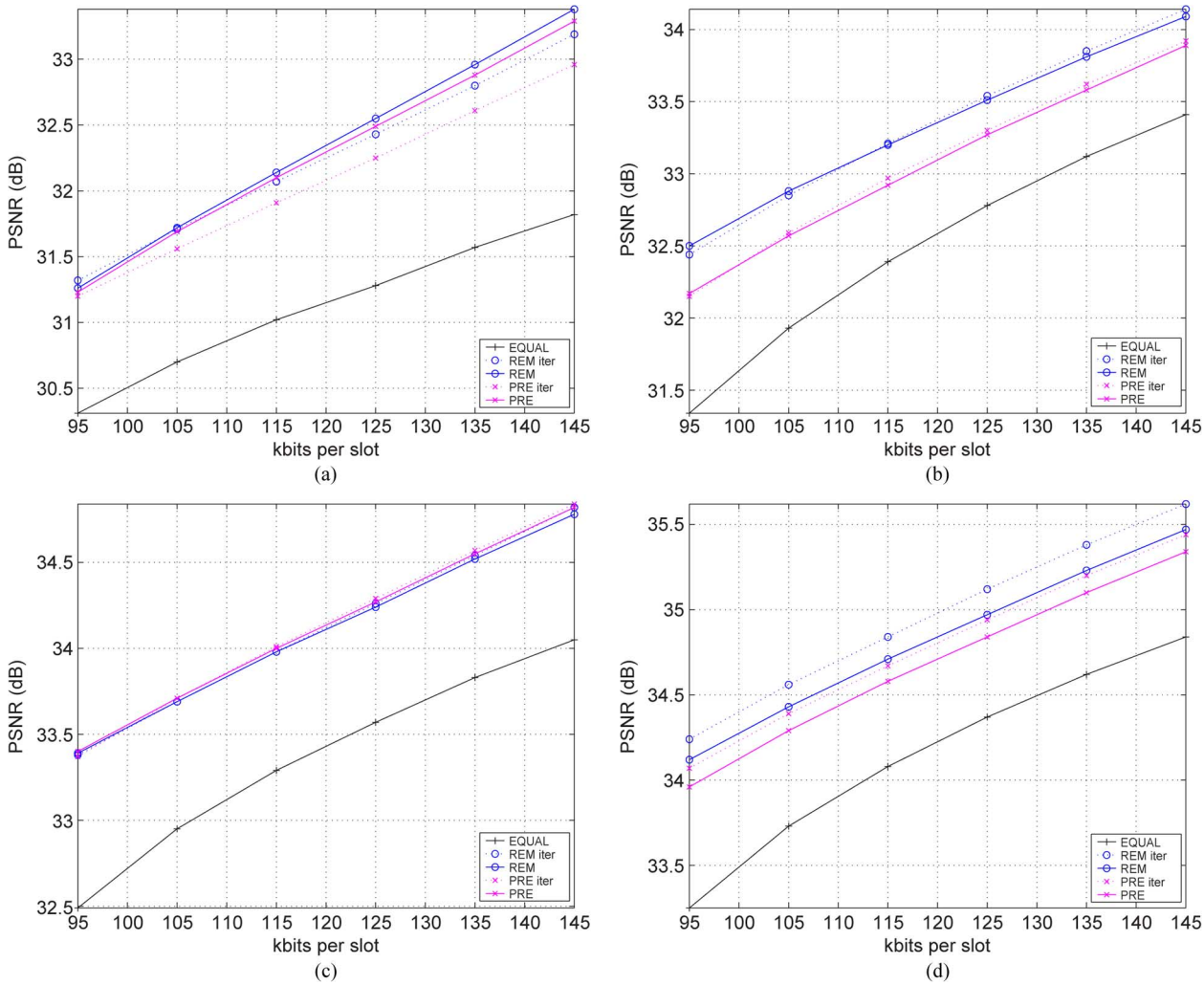

Fig. 5. PSNR performance versus bit rate for four multiplexed video streams for comparing the effect of iterative pricing ($\delta_p = 0.05$) and demand normalization. (a) g8 video stream. (b) g9 video stream. (c) g10 video stream. (d) g11 video stream.

are the same as in Fig. 2. "REM iter" and "PRE iter" (dotted curves) represent the quality achieved by iterating the price to obtain the equilibrium price for the REM and PRE methods. We find that, for g9 and g11, iterative pricing produces marginally better quality than the normalization procedure, and the trends are opposite for g8, while there is negligible difference between

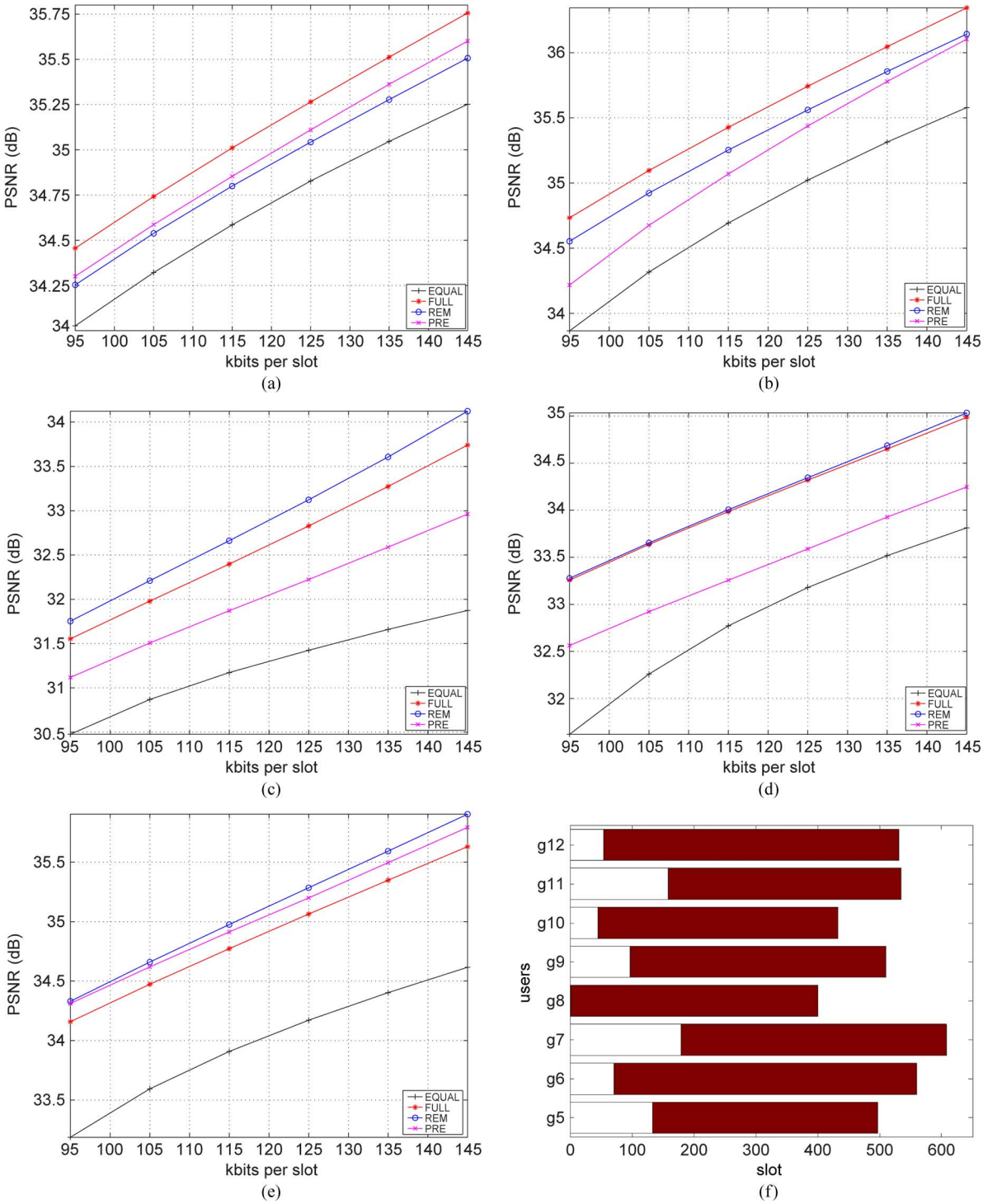

Fig. 6. PSNR performance versus bit rate for five of the eight multiplexed video streams with different start and end times. (a)–(e) show the results for five videos and (f) shows the start and end times of each video. (a) g5 video stream. (b) g6 video stream. (c) g8 video stream. (d) g9 video stream. (e) g10 video stream. (f) start and end times of 8 video users.

these two procedures for g10. While quality is overall similar, iterating over price involves sending messages back and forth which may not be suited for time critical applications.

E. Constant Bit Rate and Variable Number of Users at Any Slot

Until now, we have assumed that users have the same start time and video length. This would not be the case in a practical scenario. Fig. 6(f) shows an example profile of eight video users with different start times and video lengths. The start times are randomly chosen uniformly between slots 0–200, and the video lengths are random (uniform) between 300–500 slots.

We simulated our decentralized allocation for the videos with the profile of Fig. 6(f). The quality improvement for five of the eight videos is given in Fig. 6 for the CBR channel. All videos

benefit from the multiplexing. In general, quality improvement increases with the number of users participating in the multiplexing process, as shown previously. However, the quality improvement also depends on the amount of time overlap among users.

With many users, each of whom may enter or leave the system at any time, changes in the number of users at any slot with respect to the total number of users will make little difference to other users. With different start times and video lengths, we have shown that all of the users still benefit from the multiplexing process. As the number of users increases, the system behaves more like the case of a constant number of users at all slots.

V. Conclusion

We have demonstrated various methods of price-based decentralized bit-rate allocation among multiple video streams. A user independently calculates his bit-rate demand for the current slot based on current price, available money, and relative video complexity for the current slot compared to the estimated average complexity for future slots. The demand is sent to the allocator who normalizes the total demand and sends the bit-rate price for the next slot based on the total demand and total available bit rate.

In comparison with existing multiplexing methods [1]–[4], our method improves quality for all users whereas previous methods, focussing on improving the average quality only, caused some users to improve at the expense of others.

In comparison to the centralized allocation [5], the proposed method makes the following contributions.

1) The burden of RD information exchange in the centralized method has been reduced to transmitting only the bit-rate demand.
2) The computational burden in centralized allocation increases exponentially with the number of users. In our proposed method, the computational burden is small, is shifted to individual users, and is independent of the number of users, yet the algorithm has similar performance.

In comparison with our own previously published work [18], the current paper increases the number of users from four to ten and makes the following contributions.

1) We considered the case of an output delay buffer where, instead of normalizing the demands at every slot, the allocator stores the excess demand and strives to allocate the actual bit rate demanded by each user. This further improves the quality since the bit-rate demand is met at almost all time slots.
2) We showed that our method using a single iteration for bit-rate price determination performs close to the case when the equilibrium price is achieved iteratively.
3) We examined the case where users start at different times and have different video lengths. All users improve quality; the trends are consistent with the case where users start and end at the same time.

References

[1] M. Tagliasacchi, G. Valenzise, and S. Tubaro, "Minimum variance optimal rate allocation for multiplexed H.264/AVC bitstreams," *IEEE Trans. Image Process.*, vol. 17, no. 7, pp. 1129–1143, Jul. 2008.

[2] L. Wang and A. Vincent, "Bit allocation and constraints for joint coding of multiple video programs," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 6, pp. 949–959, Sep. 1999.

[3] G. Su and M. Wu, "Efficient bandwidth resource allocation for low-delay multiuser video streaming," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 9, pp. 1124–1137, Sep. 2005.

[4] A. Fattahi, F. Fu, M. van der Schaar, and F. Paganini, "Mechanism-based resource allocation for multimedia transmission over spectrum agile wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 4, pp. 601–612, Apr. 2007.

[5] M. Tiwari, T. Groves, and P. Cosman, "Competitive equilibrium bitrate allocation for multiple video streams," *IEEE Trans. Image Process.*, vol. 19, no. 4, pp. 1009–1021, Apr. 2010.

[6] F. Fu and M. van der Schaar, "Noncollaborative resource management for wireless multimedia applications using mechanism design," *IEEE Trans. Multimedia*, vol. 9, no. 4, pp. 851–868, Jun. 2007.

[7] X. Zhu and B. Girod, "Distributed rate allocation for multi-stream video transmission over ad hoc networks," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2005, pp. 157–160.

[8] H. Park and M. van der Schaar, "Bargaining strategies for networked multimedia resource management," *IEEE Trans. Signal Process.*, vol. 55, no. 7, pp. 3496–3511, Jul. 2007.

[9] F. Fu, T. Stoenescuand, and M. van der Schaar, "A pricing mechanism for resource allocation in wireless multimedia applications," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 2, pp. 264–279, Aug. 2007.

[10] F. Kelly, A. Maulloo, and D. Tan, "Rate control for communication network: Shadow prices, proportional fairness and stability," *J. Oper. Res. Soc.*, vol. 49, pp. 237–252, 1998.

[11] F. Li, D. Tolley, N. Padhy, and J. Wang, "Framework for assessing the economic efficiencies of long-run network pricing models," *IEEE Trans. Power Syst.*, vol. 24, no. 4, pp. 1641–1648, Nov. 2009.

[12] A. Mohsenian-Rad, V. Wong, and V. Leung, "Two-fold pricing to guarantee individual profits and maximum social welfare in multi-hop wireless access networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 8, pp. 4110–4121, Aug. 2009.

[13] G. Zachhariadis and J. Barria, "Dynamic pricing and resource allocation using revenue management for multiservice networks," *IEEE Trans. Netw. Service Manag.*, vol. 5, no. 4, pp. 215–226, Dec. 2008.

[14] D. Niyato and E. Hossain, "Market-equilibrium, competitive, and cooperative pricing for spectrum sharing in cognitive radio networks: Analysis and comparison," *IEEE Trans. Wireless Commun.*, vol. 7, no. 11, pp. 4273–4283, Nov. 2008.

[15] D. Niyato and E. Hossain, "A game theoretic analysis of service competition and pricing in heterogeneous wireless access networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 12, pp. 5150–5155, Dec. 2008.

[16] D. Niyato and E. Hossain, "Competitive pricing for spectrum sharing in cognitive radio networks: Dynamic game, inefficiency of Nash equilibrium, and collusion," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 1, pp. 192–202, Jan. 2008.

[17] E. Malinvaud and M. Bacharach, "Decentralized procedures for planning," in *Activity Analysis in the Theory of Growth and Planning*. New York: St. Martin's, 1967, pp. 170–208.

[18] M. Tiwari, T. Groves, and P. Cosman, "Pricing-based decentralized rate allocation for multiple video streams," in *Proc. IEEE Int. Conf. Image Process.*, Nov. 2009, pp. 3065–3068.

[19] K. Stuhlmüller, N. Färber, M. Link, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 1012–1032, Jun. 2000.

[20] L. Lin and A. Ortega, "Bit-rate control using piecewise approximated rate-distortion characteristics," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 4, pp. 446–459, Aug. 1998.

[21] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[22] A. Mas-Colell, M. Whinston, and J. Green, *Microeconomic Theory*. Cambridge, U.K.: Oxford Univ. Press, 1995.

[23] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[24] *"H.264/AVC ref. Software,"* Fraunhofer HHI. [Online]. Available: http://iphome.hhi.de/suehring/tml

[25] K.-P. Lim, G. Sullivan, and T. Wiegand, "Text description of joint model reference encoding methods and decoding concealment methods," *JVT of ISO/IEC MPEG and ITU-T VCEG, JVT-K049*, Mar. 2004.

Mayank Tiwari (S'03–M'10) received the B.E. degree in electronics and communication engineering from the Indian Institute of Technology, Roorkee, India, in 1999, the M.S. degree in electrical engineering from the Arizona State University, Phoenix, in 2004, and the Ph.D. degree in electrical engineering from the University of California at San Diego, La Jolla, in 2010.

From June 1999 to May 2001, he was with Hughes Software Systems, Gurgaon, India, working on the design and development of video and speech codecs. From June 2001 to August 2002, he was with Cybernetics Infotech Inc., Rockville, MD, working on the design and development of low bit-rate speech codecs. Currently, he is with Qualcomm Inc., San Diego, CA, working in the QCT video systems group. His research interests include image and video compression and mobile multimedia communication.

Theodore Groves received the B.A. degree from Harvard University, Cambridge, MA, and the Ph.D. degree in economics from the University of California, Berkeley, in 1970.

Prior to joining the University of California at San Diego (UCSD), La Jolla, as a Professor of Economics in 1979, he was a faculty member with the University of Wisconsin, Madison, Northwestern University's Kellogg School of Management, and Stanford University. He was a founder of mechanism design theory and the discoverer of the "Groves Mechanism" for eliciting truthful information in an incentive-compatible manner. He and coauthor John Ledyard also developed the first general equilibrium solution to the "free rider problem" of public goods. He has also studied the Chinese economy's transition to a market economy, optimal policies for minimizing the occurrence of oil spills, the incentive compatibility of stated-preference surveys, and improved methods for video multiplexing. He is the Director of the Center for Environmental Economics in the Department of Economics, UCSD, and is involved in ongoing research on international and national fisheries, water pricing, and numerous projects for managing marine resources and the protection of endangered species.

Prof. Groves is a Fellow of the Econometric Society and the American Academy of Arts and Sciences.

Pamela Cosman (S'88–M'93–SM'00–F'08) received the B.S. degree (with honors) from the California Institute of Technology, Pasadena, in 1987, and the M.S. and Ph.D. degrees from Stanford University, Stanford, CA, in 1989 and 1993, respectively, all in electrical engineering.

She was a National Science Foundation (NSF) Postdoctoral Fellow at Stanford University and a Visiting Professor with the University of Minnesota from 1993 to 1995. In 1995, she joined the faculty of the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, where she is currently a Professor. She was the Director of the Center for Wireless Communications from 2006 to 2008. Her research interests are in the areas of image and video compression and processing, and wireless communications.

Dr. Cosman is a member of Tau Beta Pi and Sigma Xi. She was the recipient of the Electrical and Computer Engineering Departmental Graduate Teaching Award, an NSF Career Award, a Powell Faculty Fellowship, and a Globecom 2008 Best Paper Award. She was a guest editor of the June 2000 special issue of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS on "Error-resilient image and video coding," and was the Technical Program Chair of the 1998 Information Theory Workshop in San Diego. She was an associate editor of the IEEE COMMUNICATIONS LETTERS (1998–2001) and an associate editor of the IEEE SIGNAL PROCESSING LETTERS (2001–2005). She was the Editor-in-Chief (2006–2009) as well as a Senior Editor (2003–2005, 2010–present) of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS.