Resource Allocation for Multimedia Streaming Over the Internet

Qian Zhang, *Member, IEEE*, Wenwu Zhu, *Senior Member, IEEE*, and Ya-Qin Zhang, *Fellow, IEEE*

***Abstract*—This paper addresses the resource allocation problem for multiple media streaming over the Internet. First, we present an end-to-end transport architecture for multimedia streaming over the Internet. Second, we propose a new multimedia streaming TCP-friendly protocol (MSTFP), which combines forward estimation of network conditions with information feedback control to optimally track the network conditions. Third, we propose a novel resource allocation scheme to adapt media rate to the estimated network bandwidth using each media's rate-distortion function under various network conditions. By dynamically allocating resources according to network status and media characteristics, we improve the end-to-end quality of services (QoS). Simulation results demonstrate the effectiveness of our proposed schemes.**

***Index Terms*—Flow control, multimedia streaming, QoS, rate control, resource allocation, TCP-friendly.**

I. Introduction

THERE has been tremendous interest in audiovisual streaming over the Internet recently. However, efficient delivery of streaming media over the Internet presents many challenges. On one hand, the current Internet only provides best-effort service and it does not provide quality of service (QoS) guarantee or provision for multimedia services. Specifically, network conditions and characteristics such as bandwidth, packet loss ratio, delay, and delay jitter vary from time to time. On the other hand, many media encoders generally do not take the network conditions into account. In general, different kinds of media have different characteristics. Real-time media such as video or audio is delay sensitive but capable of tolerating certain degree of errors. Nonreal-time media such as Web data is less delay sensitive but requires reliable transmission. Consequently, different types of media may have different quality impairments under the same network condition. Therefore, designing a high-quality multiple media streaming system that can cope with varying Internet conditions becomes important.

To date several schemes have been developed for QoS management, including resource reservation, priority mechanism, and application control. Resource reservation (e.g., RSVP) is the most straightforward approach [1]. However, RSVP requires that all routers have QoS supports. In addition, it may tend to over-allocate resources for QoS guarantee, thus leading to network under-utilization. In priority-based mechanisms, different data packets or streams are labeled with different priorities and treated differently in the network routers. This is also called differentiated service (DiffServ). However, the exact mechanism for setting the priority levels and mapping from the application priority levels to the router priority levels, the router mechanism for controlling these levels and the performance gains for defining priority classes are under investigation [2], [3]. In application control, the QoS is enforced by congestion control and transmission-rate adaptation [4]–[6]. The advantage of it is that there is almost no need to change the router or network itself, the main challenge, however, is to design efficient congestion and flow control.

To efficiently transport media over the Internet, both real-time and nonreal-time systems are expected to react to congestion by adapting their transmission rates and maintain the inter-protocol's fairness. Since a dominant portion of today's Internet traffic is TCP-based, it is very important for multimedia streams to be "TCP-friendly," by which we mean a media flow generates similar throughput as a typical TCP flow along the same path under the same condition with lower latency. There are two existing groups of TCP-friendly flow-control protocols for multimedia streaming applications: 1) sender-based rate adjustment and 2) model-based flow control. Sender-based rate adjustment [4], [5], [7] performs additive increase and multiplicative decrease (AIMD) rate control in the sender as in TCP. The transmission rate is increased in a step-like fashion in the absence of packet loss and reduced multiplicatively when congestion is detected. This approach usually requires the receiver to send an acknowledgment for every received packet to detect congestion indications, such as packet loss and timeouts. The drawbacks of this approach are as follows:

1) Network congestion could severely degrade the performance since frequent feedback packets are needed for flow control.
2) The time-varying network status cannot be reflected since the control scheme is independent of packet loss ratio, bandwidth variation, and adjusting interval.

Model-based flow control [8]–[10], on the other hand, uses a stochastic TCP model [11], which represents the throughput of a TCP sender as a function of packet loss ratio and round trip time (RTT). Since this protocol can run in the receiver, the congestion problem in the reverse path can be avoided. However, this approach also has its shortcomings. First, the available bandwidth may be over-estimated or under-estimated for high packet loss ratio. Second, the estimated packet loss ratio is not for the next time interval so as to affect the accuracy of the throughput calculation. Third, sending rate is reassigned to meet the calculated bandwidth and its fluctuation is not suitable for continuous media.

Manuscript received March 26, 2001. The associate editor coordinating the review of this paper and approving it for publication was Dr. Jie Chen.

The authors are with Microsoft Research, Beijing, 100080, China (e-mail: wwzhu@microsoft.com).

Publisher Item Identifier S 1520-9210(01)07886-5.

To address the above issues, we propose a new multimedia streaming TCP-friendly protocol (MSTFP) to iteratively combine forward estimation of network condition with information feedback control to optimally track network status. Our proposed MSTFP is well suited for continuous media streaming since it integrates accurate throughput calculation with history-related rate adjustment.

In some applications where audio, video and data, or a set of visual elements are delivered simultaneously over the Internet, the media rates are usually aggregated. To make the aggregated bit-rate equal to or less than the Internet available bandwidth, independent control for each media is usually employed by allocating a fixed rate to each media. However, this may lead to large variations in quality among different media and thus cause inefficient utilization of the Internet resource.

Unlike independent control, joint control only needs to maintain a constant aggregate bit-rate while allowing bit-rate of each media to vary. Recent studies have shown that joint control is more efficient than independent control for multiple media coding [12]–[14]. However, none of these approaches takes the time-varying network conditions into account. Since different media may have different quality degradations under various network situations, it is intuitive to move bits from less active and lightly degraded media to more active and heavily degraded ones.

In order to alleviate the aforementioned problems, we further propose a novel resource allocation scheme for multiple media streams to achieve end-to-end optimal quality according to the estimated network bandwidth and media rate-distortion functions.

The rest of this paper is organized as follows. In Section II, we present our end-to-end architecture for multiple media streaming over the Internet. In Section III, we first describe an Internet packet loss ratio model for forward estimation, followed by the description of available bandwidth estimation algorithm and sending rate smoothing scheme. Then, a new TCP-friendly protocol is proposed with combination of forward modeling and feedback control. In Section IV, we present our novel resource allocation algorithm. Section V gives simulation results and Section VI provides conclusions and discussions.

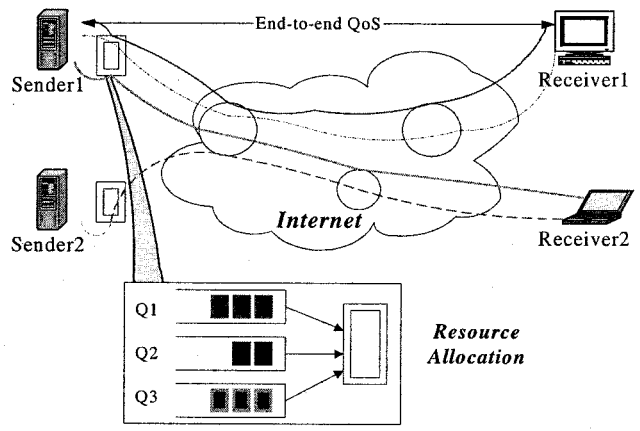

Fig. 1. General architecture consisting of multiple servers and multiple clients for media streaming.

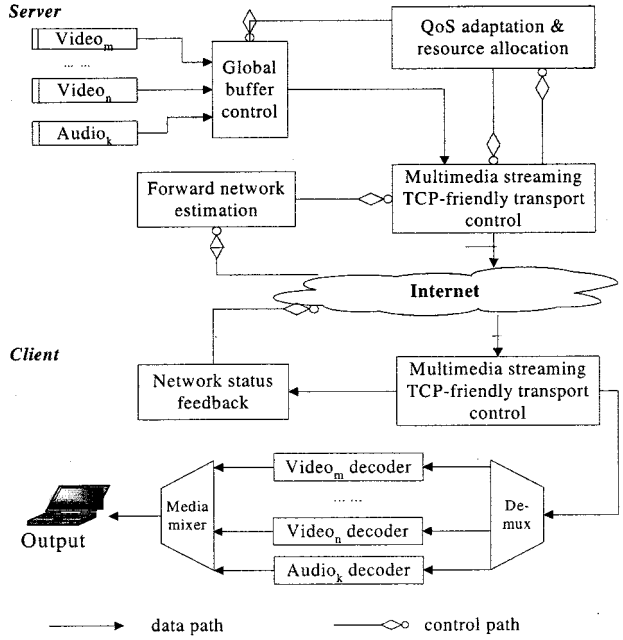

Fig. 2. End-to-end transport architecture for multiple media streaming.

II. An End-To-End Transport Architecture for Multimedia Streaming

In media streaming, multiple servers and multiple clients are usually deployed in the same session. Fig. 1 depicts a general architecture where several continuous media servers play back multimedia streams for heterogeneous clients over the Internet. Each server is able to support a large number of requests simultaneously. Each client, on the other hand, is able to request services from different servers. Since each client has its own bandwidth requirement, usually it is difficult to achieve the optimal end-to-end QoS. In this paper, resource allocation is performed in the sender to achieve such a goal.

Fig. 2 depicts our proposed end-to-end transport architecture for multimedia streaming over the Internet. The media data is first controlled by the global buffer in the sender and then transmitted using our proposed TCP-friendly protocol. On the receiver side, the control data such as packet loss ratio and packet transmission time are fed back to the encoder while received data are de-multiplexed and decoded. Consequently, the encoder uses the feedback information to forwardly estimate the channel status and control the media quality. As a result, by combining the estimated bandwidth with media characteristics, this architecture can deliver multiple media streams with good visual quality across the congested network.

The key components in this framework consist of *MSTFP, global buffer control,* and *quality adaptation and resource allocation. MSTFP* periodically monitors network status and regulates the server's transmission rate. *Forward channel estimation* and *network status feedback* are iteratively integrated to optimally track network condition.

Global buffer control module controls synchronization of various media. Different media is jointly controlled so that the buffer space can be optimally allocated to each media. As men-

tioned above, bits are moved from less active and lightly degraded media to more active and heavily degraded ones.

Quality adaptation and resource allocation adjusts the quality of the transmitted streams. First, this module periodically estimates the available bandwidth from MSTFP. Second, network-adaptive rate-distortion functions for each media are modeled. Third, bit allocation among each media is performed to adapt each media's rate to its available resource.

We will present details for those modules in the following sections.

III. Multimedia Streaming TCP-Friendly Protocol

A. Packet Loss Model

In the Internet environment, data are transmitted on a packet-by-packet basis. When delivered over the Internet, usually a packet is either received correctly or lost. These losses are mainly caused by network congestion and queueing delay. A packet loss model determines the probability of the event that a packet is lost.

We tend to model packet loss in the Internet over a period of time. It is useful to know how much past information is necessary to calculate the parameters of a loss model. Assume the packets can be represented as a binary time series $\{x_i\}_{i=1}^{n}$, where x_i takes 1 if the ith packet has arrived successfully and 0 if it is lost.

To precisely represent the loss status of the Internet, we use two kinds of information. The first one is the number of loss packets and the number of received packets; while the second one is the length of the good runs and the length of loss runs. The length of good runs is the length of the portions of the trace that consist solely of consecutive 1s. Similarly, the length of loss runs is the length of the portions of the trace that consist solely of consecutive 0s.

The typically used packet loss models are the Bernoulli model and the two-state Markov model. In the Bernoulli model [15], packets are assumed to be independent and identically distributed. Under this assumption, this model can be characterized by a single parameter, packet loss ratio r. It can be estimated from a sample trace, i.e., $\hat{r} = n_0/n$, where n_0 is the number of times the value 0 occurs (packet is lost) in the observed time series and n is the total number of samples in the time series. The length distributions of good runs and loss runs are $f(j) = \hat{r}(1-\hat{r})^{j-1}$, $j = 1, 2, \ldots, \infty$ and $f(j) = (1-\hat{r})\hat{r}^{j-1}$, $j = 1, 2, \ldots, \infty$, respectively.

In the two-state Markov model (see Fig. 3) [15], the loss process is modeled as a discrete-time Markov chain with two states. The current state x_i of the stochastic process, depends only on the previous value x_{i-1}. Unlike the Bernoulli model, this model is able to capture the dependence between consecutive losses with an additional parameter.

The transition probabilities between the two states are calculated as $p = P[x_i = 1 | x_{i-1} = 0]$ and $q = P[x_i = 0 | x_{i-1} = 1]$. The maximum likelihood estimators for p and q are $\hat{p} = n_{01}/n_0$ and $\hat{q} = n_{10}/n_1$, respectively, where n_{01} is the number of times in the observed time series when 1 follows 0 and n_{10} is the number of times when 0 follows 1. n_0 is the number of 0s and n_1 is the number of 1s in the trace. It can be derived that the length

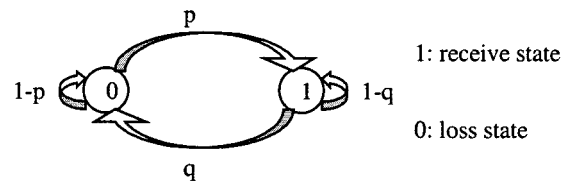

Fig. 3. Two-state Markov model.

distributions of good runs and loss runs are $f(j) = \hat{p}(1-\hat{q})^{j-1}$, $j = 1, 2, \ldots, \infty$ and $f(j) = \hat{q}(1-\hat{p})^{j-1}$, $j = 1, 2, \ldots, \infty$, respectively. This model can be characterized by the following matrix that includes two parameters, p and q:

$$\begin{pmatrix} p & 1-p \\ 1-q & q \end{pmatrix}. \tag{1}$$

Note that if q equals $1-p$, the two-state Markov model is equivalent to the Bernoulli model, meaning the probability of loss is independent of the previous state. In addition, the relative values of q and $1-p$ provide us information about how burst the losses are. Specifically, if $(1-p) > q$, the loss of current packet is more likely provided that the previous packet was lost rather than received successfully. If $q > (1-p)$, on the other hand, the loss of current packet is more likely if the previous packet was not lost.

In this paper, we use this two-state Markov model (Gilbert model) to model packet loss in the Internet.

B. TCP-Friendly Protocol for Multimedia Streaming

For audio/video streaming application, it is desirable to adjust its transmission rate according to the perceived congestion level in the network. Through this adjustment, a suitable loss level can be maintained and bandwidth can be shared fairly between connections.

To reach such a goal, we take the characteristics of packet loss and bandwidth fluctuation into account. We propose a new MSTFP that minimizes the number of future packets likely to be dropped and smoothes the sending rate. Our MSTFP works as follows. The receiver monitors the network condition and gathers related information; while the sender changes its sending rate according to the available network bandwidth estimated from the packet loss ratio, RTT, and retransmission timeout (RTO) values.

The MSTFP protocol is composed of a sender part and a receiver part. The sender transmits data packets to the receiver at a certain rate. The header of the sender-side packet includes the packet sequence number, time stamp indicating the time when the packet is sent ($ST1$), and the size of the sending packet. The receiver sends feedback to the sender at regular intervals. The receiver-side packet consists of the time interval of a packet spent in the receiver side (ΔRT), the time stamp of the packet sent from the sender ($ST1$), the estimated packet loss ratio, and the rate at which data is received. Based on the receiver's feedback, the sender adjusts its sending rate in a TCP-friendly manner.

The process of optimally tracking of network conditions using MSTFP consists of four stages:

1) estimating packet loss ratio;
2) estimating RTT and RTO;

3) estimating available network bandwidth;
4) adjusting sending rate.

1) Packet-Loss-Ratio Estimation: The receiver monitors the network condition and records the states of lost or received packets. The state transition probability is first calculated using the formula introduced in the previous section. Then the probability of the event that the trace is in the loss state is given by

$$P_L = \frac{\hat{q}}{\hat{p}+\hat{q}}. \tag{2}$$

2) RTT and RTO Estimation: Based on the receiver's feedback packet, the sender adjusts the current RTT using the following equation:

$$\text{RTT} = \alpha \times \overline{\text{RTT}} + (1-\alpha) \times (\text{now} - ST1 - \Delta RT) \tag{3}$$

where

$\overline{\text{RTT}}$	current round trip time;
RTT	estimated round trip time;
now	time stamp indicating the time at which the packet was received in the sender;
α	weighting parameter that is set to 0.75 in our work.

Note that this weighting parameter α is set to 0.875 in the traditional TCP congestion control algorithm. Considering the real-time requirement, we choose a smaller weighting value such that the recent RTT value has a higher impact on the RTT estimation.

After calculating the RTT, the RTO can be defined as

$$\text{RTO} = \text{RTT} + k \times \text{RTTVAR} \tag{4}$$

where k is recommended to be 4; RTT is the estimated round trip time; and RTTVAR is a smoothed estimate of the variation of RTT which can be represented as

$$\text{RTTVAR} = \alpha_2 \times \overline{\text{RTTVAR}} + (1-\alpha_2) \times |\text{RTT} - (\text{now} - ST1 - \Delta RT)| \tag{5}$$

where $\overline{\text{RTTVAR}}$ is the current RTT variation, and the weighting parameter α_2 is set to 0.25, as discussed in [16].

3) Available Bandwidth Estimation: After the above procedures, the sender estimates current available network bandwidth. The typical formula used to estimate the network throughput is [4], [5], [9]

$$\text{rcvrate} = \frac{C}{\text{RTT} \times \sqrt{p_L}} \tag{6}$$

where C is a constant that is usually set to either 1.22 or 1.31, depending on whether the receiver uses delay acknowledgment. As discussed in Section I, this formula is not suitable for estimating bandwidth in some cases. In other words, since (6) does not take timeouts into account, it usually overestimates the connection throughput when loss ratio is higher than 5%.

Padhye *et al.* [11] proposed a formula to calculate the network throughput. Afterward, several people adopted this formula in their studies [6], [8]. This formula is given by (7), shown at the bottom of the page, where

PacketSize	size of the sending packet;
RTT	estimated round trip time;
RTO	retransmission timeout;
P_L	packet loss ratio.

It can be seen that RTO has been considered in this formula. Note that in [6], [8], [11], and [17], the current packet loss ratio and RTT are used, which is not suitable for the future bandwidth estimation at the next time interval. In our approach, the calculated RTT and P_L are all estimated for the next time interval, which can be used to estimate the future available bandwidth using (7).

4) Network-Based Sending Rate Adjustment: Having learned the available bandwidth, the sender can adjust its sending rate based on the estimated value. Note that the TCP-throughput model is based on the previous loss and delay observed during the lifetime of a connection. However, rate adjustment decisions need to be made based on the current loss and delay values. As the observed losses and round trip delays vary dynamically, using TCP-throughput model as the only criterion for rate adjustment results in a rather fluctuant sending rate that might lead to annoying perceived quality for multimedia. Usually TCP congestion control is based on AIMD, which increases its transmission rate by an additive increase rate (AIR) without facing losses and decreases its transmission rate by a multiplicative decrease rate (MDR) otherwise. In our scheme, we use the estimated network bandwidth to dynamically adjust the sending rate. To smooth the sending rate, we alter the transmission rate according to some network-related information, such as network congestion degree (i.e., packet loss ratio), bandwidth variation, and adjusting interval, as follows.

- Increment in transmission rate: appropriate if the sender experiences loss when transmission rate is less than the available connection capacity.
- Slight reduction in transmission rate: appropriate if the sender experiences loss when transmission rate is at or around the available connection capacity.
- Aggressive reduction in transmission rate: appropriate if the sender experiences congestion loss when transmission rate is higher than the available connection capacity.

Quantitatively, we use the following procedure to control the sending rate related to the current packet loss ratio, bandwidth

$$\text{rcvrate} = \frac{\text{PacketSize}}{\text{RTT} \times \sqrt{\frac{2P_L}{3}} + 3 \times \text{RTO} \times P_L \times \sqrt{\frac{3P_L}{8}} \times (1+32P_L^2)} \tag{7}$$

variation, and adjusting interval (see (8) at the bottom of the page)

where

$\overline{\text{now}}$	time stamp indicating the time at which the current adjustment happened;
lastchange	time stamp indicating the time when last adjustment occurred;
$\overline{\text{currate}}$	current sending rate;
currate	updated sending rate;
β	weighting parameter for rate smoothing that is set to 0.75 in our work;
R_f	reduction factor that determines the degree of the reaction of the sender's losses.

A higher value results in a faster reduction of the transmission rate but a more oscillatory behavior. A lower value, on the other hand, leads to more stable rate value but results in longer convergence period. Note that R_f is related to the adjustment interval. In our scheme, this reduction factor is set based on the adjustment interval. Considering the tradeoff between convergence time and stability, R_f is constrained between 1 and 2.

The advantage of our scheme is that the sending rate can be increased or decreased very smoothly according to the network-related information. In other words, MSTFP has less variation in the transmission rate and is less sensitive to packet loss. In summary, our proposed MSTFP has two good features: "TCP-friendliness" and "rate smoothness."

IV. Resource Allocation for Multimedia Streaming

Dynamic resource allocation is essential for distributed multimedia systems that support application-level control. As described in Section II, it can be implemented by integrating global buffer control, media rate control, and dynamic network bandwidth estimation. One of the challenges of resource reallocation is global coordination of the feedback information from multiple streams.

Since bandwidth resource is scarce in today's Internet, it is important to manage available bandwidth resource in an optimal way, e.g., send the most relevant contents to the receiver according to the available bandwidth. If the essential (e.g., higher priority) contents are available with acceptable quality, the final quality may be higher than that if all the contents are available with unacceptable quality.

Different applications such as file-transfer, web browsing, and audio/video streaming have different tolerances to mismatches between sending rate and network bandwidth. For example, file-transfer does not have real-time constraint. Audio/video streaming, on the other hand, has real-time constraint. The difference between the sensitivities to human aura and visual systems indicates that the audio and video should be handled differently when adverse conditions arise, thereby affecting the playback of media streams. It is known that the aural sense is more sensitive to disturbances than the visual one. Therefore, it is appropriate to assign higher priority to audio than video. If data needs to be discarded when congestion occurs in the network, it is preferable to discard the video data first. For the video data, in some applications it may also be possible to use receiver's information, e.g., delete backgrounds or transmit high-priority objects when there is not enough bandwidth available, to improve the subjective quality. In some other applications, we may just transmit the important layers such as base layer and lower enhancement layers when bandwidth is limited. Therefore, selectively protecting the scene content according to the priority of its relevance and its application is very useful and important for the final subjective impact.

Our objective of resource allocation is to minimize the overall distortion under the total bit-rate constraint. We classify media streams into two types: one is continuous media (CM); while the other is noncontinuous media (NCM). MSTFP is used for delivering CM, such as audio and video, and TCP is adopted for transmitting NCM, such as emails and web traffic. We denote the sending rate of ith media stream for the CM as r_i, the distortion obtained in this stream as d_i, and the quality-impact degree of this stream as α_i. We further denote the sending rate of jth media stream for the NCM as r_j, which is determined by the maximum tolerant latency of the media. Then our resource allocation problem is formulated as

$$\begin{aligned} &\text{Minimize } D = \sum_{i \in CM} \alpha_i \times d_i \\ &\text{subject to } R = \left(\sum_{i \in \text{CM}} r_i + \sum_{j \in \text{NCM}} r_j \right) \le R_T \end{aligned} \tag{9}$$

where R_T is the total bit budget for the current time instant obtained from MSTFP, i.e.,

$$R_T = \sum_i \text{currate}_i. \tag{10}$$

$$\begin{aligned} &R_f = \frac{(\overline{\text{now}} - \text{lastchange})}{\text{RTT}} \\ &1 \le R_f \le 2 \\ &\text{if } (\text{rcvrate} > \overline{\text{currate}}) \\ &\qquad \text{currate} = \overline{\text{currate}} + \left(\frac{\text{PacketSize}}{\text{RTT}} \right) \times R_f \times (1 - P_L) \\ &\text{else} \\ &\qquad \text{currate} = (\beta \times \text{rcvrate} + (1 - \beta) \times \overline{\text{currate}}) \times R_f \times (1 - P_L) \end{aligned} \tag{8}$$

Note that each CM media has its own rate and distortion relation given by $R_i = F(D_i)$. The above optimization problem heavily depends on this R-D function. In streaming media over the Internet, the distortion is composed of the source distortion and the network distortion. The former is caused by media rate control and the latter is caused by Internet transmission. Considering the similar characteristic of audio and video in rate control, we will use video as an example of CM to demonstrate our resource allocation scheme in this paper.

It is well known that packet loss ratio greatly affects media quality. Frossard analyzed the packet-loss effect on MPEG-2 video quality [18]. Moving picture quality metric (MPQM) is used to evaluate the subjective video quality. Video quality rating is scaled from 1 to 5. It is shown that the video quality remains constant as packet loss ratio is within a certain range. When packet loss ratio is beyond a certain value (e.g., about 10^{-4}), the video quality drops sharply. It is also shown that, the higher the bit-rate (the smaller the quantization parameter), the sharper dropping of video quality after that value. We apply this approach to MPEG-4 video and obtain similar results, as shown in Fig. 4. Fig. 5 shows the relation between MPEG-4 video quality and encoding bit-rate under packet loss ratio of 10^{-2}. From Fig. 5, it can be seen that there is an optimal point in which best quality is achieved. We observe that joint analysis of rate control with packet loss can obtain good visual quality. Such an observation is important for developing our network-adaptive resource allocation schemes.

In the following, we will present our new network-adaptive rate control scheme for MPEG-4 MoMuSys codec consisting of multiple video objects (MVOs) and network-adaptive bit allocation scheme for MPEG-4 progressive fine granularity scalable (PFGS) multilayer scalable codec [19].

A. Network-Adaptive Rate Control for Multiple Video Objects in MPEG-4

MPEG-4 is an object-based video coding standard in which a visual scene is typically composed of video objects (VOs). Each VO is individually coded to give rise to an elementary stream, which can be individually accessed and manipulated. The composition information is sent in a separate stream. For MVO streams, foreground objects receive most attention from the viewer, while background objects is less important. In addition, foreground objects usually change rapidly, while background objects change slowly. Therefore, they may have different impact on overall video quality.

Similar to other video coding standards such as MPEG-1, MPEG-2, and H.26x, the bit-rate of video encoder can be affected by choosing a quantizer of each DCT transformed macroblock. Since the network bandwidth used for transmission of the coded video sequence is finite, a rate-control system is needed for achieving the best buffer occupancy related to the available network bandwidth. Since the MPEG-4 coder is designed for coding each video object plane (VOP) independently, each VOP can be assigned different quantizer to regulate its bit-rate according to the global rate target that should be network and buffer controlled.

In this section, we describe a rate control scheme for MVOs that achieves minimal distortion in a global scene. Fig. 6 illustrates the block diagram of the proposed rate control scheme for MVOs. Compared to the standard rate control schemes in MPEG-4 [13], [14], our proposed approach has the following new features:

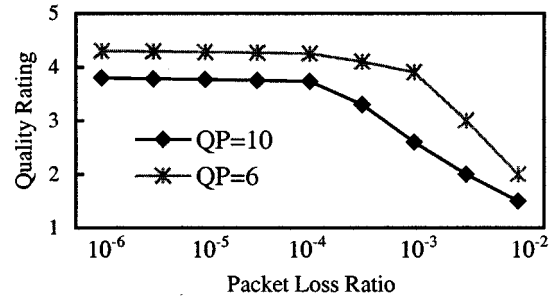

Fig. 4. Packet loss effect on video quality.

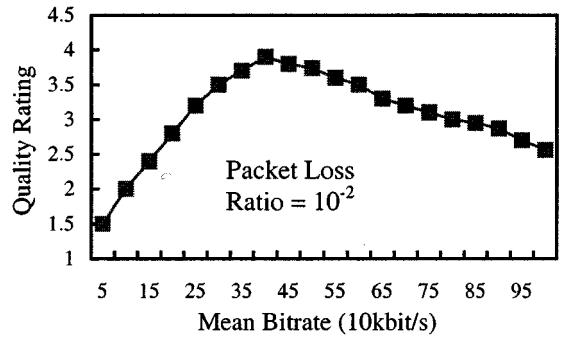

Fig. 5. Relation between video quality and encoding bit-rate under a certain packet loss ratio.

- new rate-distortion model that adapts to the varying network condition;
- global buffer updating to control frame skipping and synchronization among various objects;
- dynamically allocate target bit-rate among different objects according to the estimated network bandwidth.

1) Network-Related R-D Model: In the current MPEG-4 rate control scheme, the number of bits corresponding to shape, motion, and texture information can be directly predicted by taking the observed value from the previous VOP of the same type and the same object. The employed stochastic model of the texture coding process consists of rate prediction function and distortion prediction function. Chiang *et al.* [20] proposed the following two quadratic functions to predict the number of bits that used to encode the ith VOP and estimate the corresponding distortion:

$$r_i = \frac{(p_1)_i \times \mathrm{MAD}_i}{QP_i} + \frac{(p_2)_i \times \mathrm{MAD}_i}{QP_i^2} \tag{11}$$

$$d_i = (q_1)_i \times QP_i + (q_2)_i \times QP_i^2 \tag{12}$$

where MAD_i is the mean absolute difference of the ith VOP, and p_1, p_2, q_1, and q_2 are control parameters. Appropriate updating of control parameters has been developed. However, packet loss occurred during video transmission is not considered in the above functions.

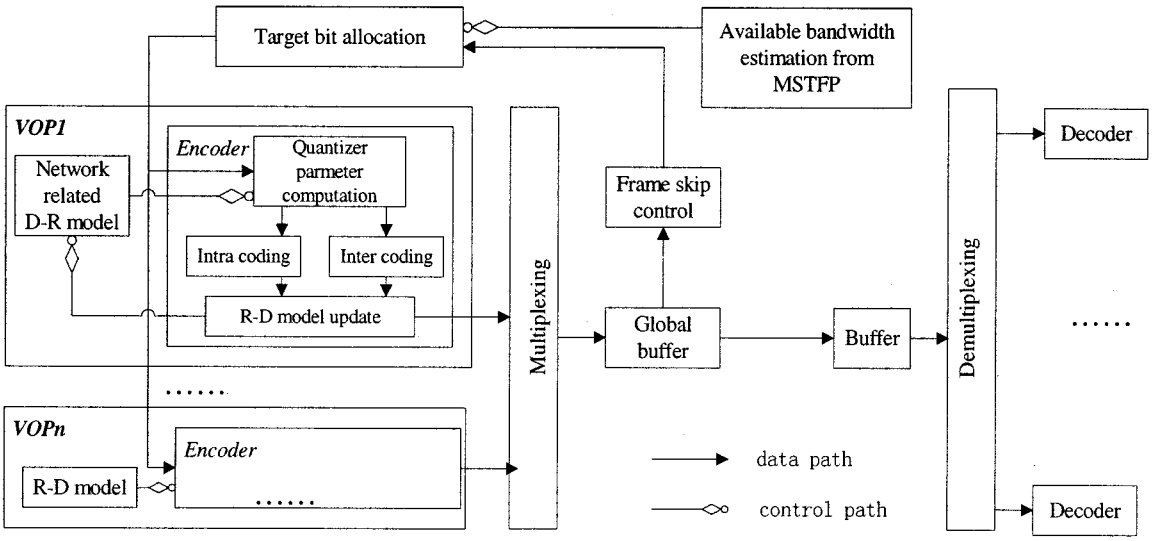

Fig. 6. Block diagram of our network-adaptive rate control scheme.

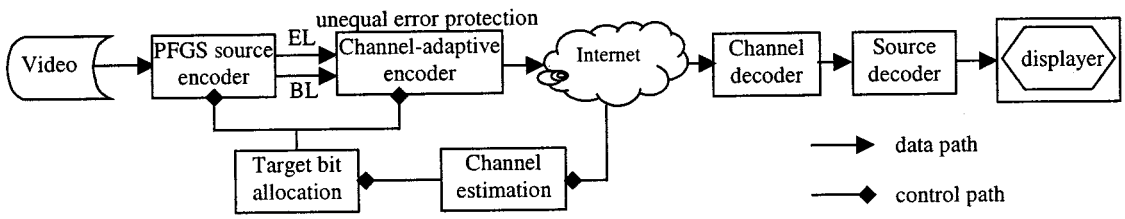

Fig. 7. Block diagram of our network-adaptive bit allocation scheme for PFGS streaming with UEP.

As stated earlier in this section, the varying packet loss ratio and the sending rate have impact on video quality. Video quality deteriorates quickly as the packet loss ratio and the sending rate beyond a certain point. To take the packet loss ratio into account, we modify (12) as follows:

$$d_i = (q_1)_i \times QP_i + (q_2)_i \times QP_i^2 + (q_3)_i \times r_i \times (P_L)_i \quad (13)$$

where q_3 is an additional control parameter.

2) R-D Model Updating: In (11)–(13), all the control parameters vary with time and depend on the coding types of a VOP. The rate-distortion model should be updated based on the encoding results of the current frame as well as those from a specified number of previous frames. Assuming that the sequence properties vary slowly, the estimation of the model parameters $\{p_i, q_i\}$ can be done by the least mean squares (LMS) approach as in [14].

There are different strategies for model estimation and updating. In our approach, we try to ensure that the derived functions have realistic dependence on the quantization parameter (QP). A different set of model parameters are kept for the VOP and are updated after encoding each VOP by applying the LMS adjustment on a data set consisting of n most recent observations. Each resultant function is then checked for monotonicity in the interval $[QP_{\min}, QP_{\max}]$. If the monotonicity test fails, the control parameters are calculated by a convex linear combination of those control parameters. Mathematically, if the initial coefficients are q_1^*, q_2^*, q_3^*, and the coefficients from the previous model are q_1^{n-1}, q_2^{n-1}, and q_3^{n-1}, then the control parameters can be updated as follows:

$$q_1^n = (1-\delta_{\min})q_1^* + \delta_{\min} q_1^{n-1} \quad (14)$$

$$q_2^n = (1-\delta_{\min})q_2^* + \delta_{\min} q_2^{n-1} \quad (15)$$

where $\delta_{\min}$ is obtained so that the polynomial

$$[(1-\delta)q_1^* + \delta\, q_1^{n-1}]x + [(1-\delta)q_2^* + \delta\, q_2^{n-1}]x^2 \quad (16)$$

is nondecreasing in the interval $[QP_{\min}, QP_{\max}]$. Finally, q_3^n is updated by the LMS approach.

3) Target Bit Allocation and Quantization Adjustment: The objective of rate control is to allocate the target bits properly so as to minimize the overall distortion to achieve the optimal quality for all the video frames. Considering the quality impact of different VOs, a natural choice for the objective of rate control is to minimize the weighted average of the distortion of the different VOs. Such a problem can be represented as

$$\text{minimize } D = \sum_i w_i \times d_i$$

$$\text{subject to } R = \sum_i r_i \le R_T \quad (17)$$

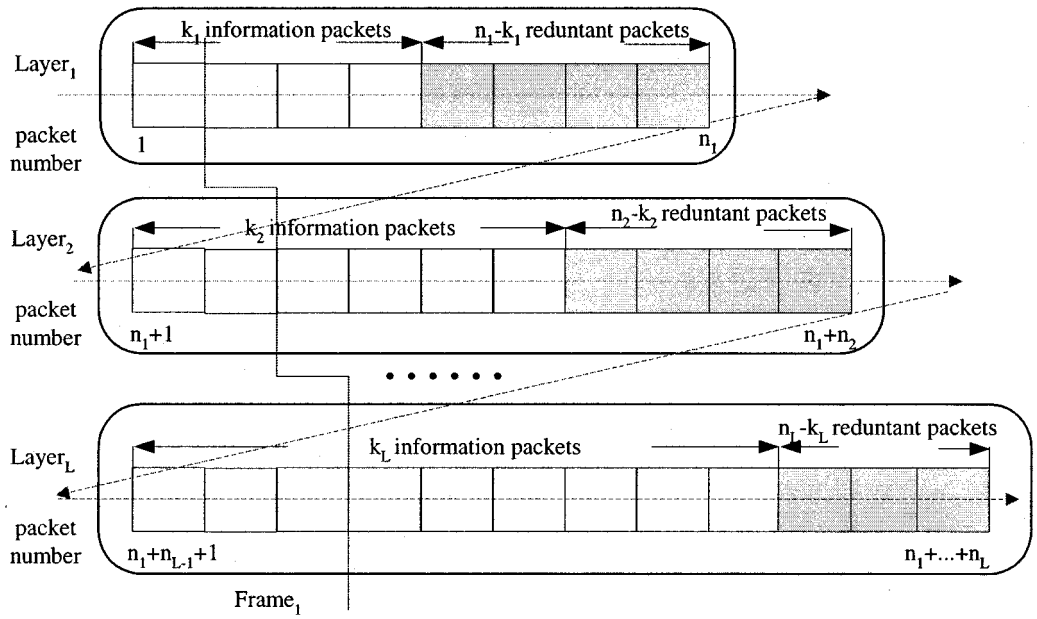

Fig. 8. Packetization scheme for PFGS with UEP.

where R_T is the total bit-rate budget for the current time instant obtained from the buffer control algorithm and the MSTFP protocol, and w_i is the weighting factor for the ith VO. By assigning larger weighting factors to the semantically more important VOs, the encoder is indirectly instructed to be more careful with them than with other less important objects. Setting all weighting factors to be equal, i.e., $w_i = 1$, results in a dynamic allocation of the rate aiming at minimizing the average distortion over the global scene.

In our scheme, two different approaches can be used to determine the weighting factor for each video object. One is to allow users to assign these factors, w_i, for each VOP to represent the priority of the VOP since each object may have different impact. The other is to calculate the factors based on the characteristics of each object, such as size, motion, and distortion. In this way, the weighting factor for object i is given by

$$w_i = \alpha_s \mathrm{SIZE}_i + \alpha_m \mathrm{MOT}_i + \alpha_d \mathrm{MAD}_i \tag{18}$$

where SIZE_i, MOT_i, and MAD_i are the size (number of macroblocks), motion (value of motion vector component), and MAD of object i, normalized by the total SIZE, MOT, and MAD of all the objects, respectively. The weights satisfy $\{\alpha_s, \alpha_m, \alpha_d\} \in [0, 1]$ and $\alpha_s + \alpha_m + \alpha_d = 1$, whose values can be assigned based on users' interest.

In order to maintain constant quality for whole video sequence, the QP of each VOP is limited to the range from 1 to 31. It is allowed to change within the pre-defined range (ΔQP), i.e., $(QP_{n-1} - \Delta QP) \leq QP_n \leq (QP_{n-1} + \Delta QP)$, where subscript n represents the current number and $n - 1$ represents the previous one.

4) Joint Global Buffer Control: Joint global buffer control is used to control frame skipping and synchronization among various objects. Using the proposed MSTFP protocol, we first estimate the current available network bandwidth (R_T). Together with the number of bits spent in the previous time instant (B_{prev}), the frame rate (F), the size ($R_{\mathrm{old}}/2$), and the occupancy (W_{prev}) of the encoder buffer, the target rate and global buffer are then updated for each frame. The total target bits from the joint buffer control are allocated for each VOP. The buffer size $R_{\mathrm{old}}/2$ is changed to $R_T/2$. The occupancy of the buffer W_{cur} is changed as follows:

$$W_{\mathrm{cur}} = \max\left(\left((W_{\mathrm{prev}} + B_{\mathrm{prev}}) \times \frac{R_T}{R_{\mathrm{old}}} - \frac{R_T}{F}\right), 0\right). \tag{19}$$

If W_{cur} is larger than a given buffer margin M, the encoder skips the encoded frames until the buffer occupancy lower than M. The buffer occupancy is reduced by R_T/F bits when one frame is skipped. In our scheme, frame skipping occurs after all the VOs in the scene have been encoded. This is because when the global scene is considered, all the VOs composed of the scene should be encoded at the same frame rate.

B. Network-Adaptive Bit Allocation for Scalable Video

In this section, we present a network-adaptive bit allocation scheme for multilayer scalable video codec. We use PFGS video codec as an example although our approach can be applied to any scalable codec such as fine granularity scalable (FGS) [21]. In our scheme, we combine PFGS codec with network-adaptive unequal error protection (UEP) across packets. We strongly protect the base layer of PFGS against packet loss so as to be decodable even if no enhancement layers are available by employing UEP based on Reed-Solomon (RS) forward error correction (FEC) code.

The difficulty encountered in joint bit allocation between source and Internet channel is how to add FEC so that the decoder can still recover the lost frames correctly. Obviously, it can be observed that under a given channel rate, the additional FEC packets reduce the available rate for source coding, thus

resulting in a tradeoff between source coding and FEC. In this paper, we address how to optimally allocate bits between source and FEC based on the R-D function such that the decoder can still successfully recover the lost packets. Specifically, in our scheme the optimal bit allocation is dynamically adjusted according to varying video characteristic and network condition. We formulate this problem as follows. Let $R(t)$ denote the network bandwidth available for transmission at time t. Let $R_S(t)$ and $R_{\mathrm{FEC}}(t)$ denote PFGS source rate and rate of FEC packets, respectively. Furthermore, let $D_S(t)$ and $D_{\mathrm{FEC}}(t)$ represent PFGS source distortion and distortion from FEC packets, respectively. Then the problem becomes how to allocate the available bit-rate at time t so that the optimal $R_S(t)$ and $R_{\mathrm{FEC}}(t)$ are obtained by minimizing end-to-end video distortion under the following constraint: $R_s(t) + R_{\mathrm{FEC}}(t) \leq R(t)$, i.e.,

$$\begin{aligned} &\text{minimize } D = D_S(t) + D_{\mathrm{FEC}}(t) \\ &\text{subject to } R_S(t) + R_{\mathrm{FEC}}(t) \leq R(t). \end{aligned} \tag{20}$$

The block diagram of our bit allocation scheme for the PFGS source and UEP is illustrated in Fig. 7. PFGS source coder encodes input video into two layers: one is the base layer (BL) that carries the most important information; the other is the enhancement layer (EL) that carries less important information. The EL bit stream can be truncated anywhere. These layers are packetized and protected against packet loss according to their importance and network status using different FEC. The *channel estimation* module adaptively updates the network status as discussed before. On the receiver side, channel decoder reconstructs packets for each layer and display video after source decoding. To efficiently deliver video over Internet, several error resilience mechanisms have been adopted in the video coder, such as error localization, data partition, error concealment, etc.

The idea of FEC across packets is to transmit additional packets that can be used in the receiver to reconstruct lost packets. Our FEC scheme uses RS codes across packets. RS codes are perfectly suitable for error protection against packet loss, because they are the only known nontrivial maximum distance separable codes, i.e., there are no other existing codes that can reconstruct erased symbols from a smaller fraction of received code symbols [22]. An RS (n, k) code with length n and dimension k encodes k information symbols containing m bits per symbol into a codeword of n symbols. With the knowledge of error position, RS (n, k) can generally correct up to $t = n - k$ symbol errors.

To evaluate the performance of an RS (n, k) code, we need to know the probability that more than $n - k$ packets are lost. We can compute this probability if we know the probability of which m packets are lost within n packets.

As stated above, we use the two-state Markov model to estimate network status. This model is determined by the distribution of error-free intervals (gap). Let gap length ν be the event that after a lost packet, $\nu - 1$ packets are received and then another packet is lost. The gap density function $g(\nu)$ gives the probability of gap length ν, i.e., $g(\nu) = \Pr(1^{\nu-1}0|0)$. The gap distribution function $G(\nu)$ is the probability of gap length greater than $\nu - 1$, i.e., $G(\nu) = \Pr(1^{\nu-1}|0)$. They can be derived as

$$g(\nu) = \begin{cases} 1 - p & \text{for } \nu = 1 \\ p(1-q)^{\nu-2}q & \text{for } \nu \geq 1 \end{cases} \tag{21}$$

$$G(\nu) = \begin{cases} 1 & \text{for } \nu = 1 \\ p(1-q)^{\nu-2} & \text{for } \nu \geq 1. \end{cases} \tag{22}$$

Let $R(m, n)$ be the probability of $m-1$ packet losses within the next $n-1$ packets followed by a lost packet. It can be calculated using recurrence, as shown in (23) at the bottom of the page. Then, the probability of m lost packets within n packets is

$$P(m,n) = \sum_{\nu=1}^{n-m+1} P_B G(\nu) R(m, n-\nu+1) \quad \text{for } 1 \leq m \leq n \tag{24}$$

where P_B is the average of packet-loss probability.

Now, the probability that more than $n - k$ packets are lost within the n packets can be represented as $\sum_{m=n-k+1}^{n} P(m, n)$. This probability is the residual loss probability experienced by a video decoder after RS decoding, which can be used to design the overall system if how many losses are acceptable for a video decoder is known.

In the multilayer scalable video coder such as PFGS, the impact of the residual loss probabilities of different layers on the video quality is not equal. This layered coding framework is well suited for prioritized transmission. Base layer can be assigned to a high-priority class while enhancement layers can be assigned to lower priority classes. Since the current Internet only provide best-effort service, prioritized transmission can be achieved by applying unequal loss protection scheme to different layers. In our work, unequal loss protection is achieved by protecting different layers with different FEC codes. More specifically, strong channel-coding protection is applied to the base-layer data stream to produce a higher-priority data class while weaker channel-coding protection is applied to the subsequent enhancement layers to produce low-priority classes. This will result in that the base-layer data stream will experience a lower packet-loss probability while delivered over Internet. The packetization of PFGS with UEP is depicted in Fig. 8. The transmission order for the packets is marked as dash line.

It is well known that efficient FEC codes are desirable to enable error recovery with as little overhead as possible. For RS

$$R(m,n) = \begin{cases} G(n) & \text{for } m = 1 \\ \sum_{\nu=1}^{n-m+1} g(\nu) R(m-1, n-\nu) & \text{for } 2 \leq m \leq n. \end{cases} \tag{23}$$

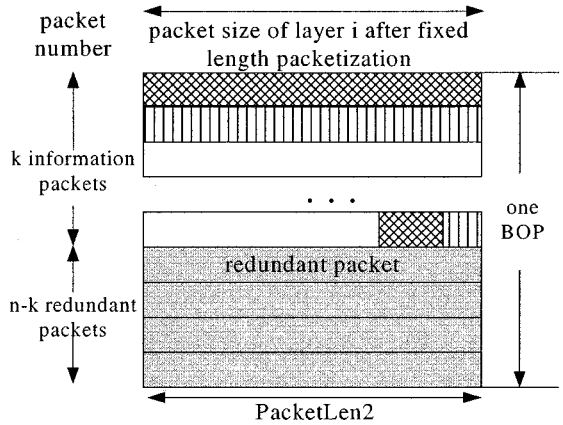

Fig. 9. Generation of FEC packets using fixed-length packetization scheme.

code used in our work, maximizing the FEC code rate k/n for specific network condition is quite important to improve the protection efficiency. We should point out that, since the data packet sizes are not fixed, for a block of k data packets, the resulting $n-k$ FEC packets are all of the maximal size (denoted as PacketLen1).Meanwhile, stuffing is needed for the k data packets. This will decrease the utilization efficiency of the available bandwidth.

To increase the bandwidth utilization, the *error resilient entropy code (EREC)* approach [23] is applied in this work (see Fig. 9) to reassemble different packets of data into k packets to form a block of packets (BOP). The basic idea of EREC is to reorganize the variable-length blocks into the EREC frame structure such that each block (slice in PFGS case) starts at a known position within the code. In this way, the decoder can independently find the start of each block. The EREC frame structure is composed of k slots of length d_i (equal to PacketLen2 in this case) bytes to yield total length of $T = \sum_{i=1}^{k} d_i$ bytes to transmit. The k slots of data can be transmitted consecutively without risk of any loss of synchronization. Each EREC frame can be used to transmit up to k variable-length blocks of data, provided that the total data to be coded does not exceed the total available T bits. The EREC places the variable-length blocks of data into the EREC code structure using a bit-reorganization algorithm that relies only on the ability to determine the end of each variable-length block. The details of the bit-reorganization algorithm can be found in [23].

By using the EREC approach, fixed length packetization is achieved. Small stuffing is needed in this fixed-length packetization scheme. The packet size changes from PacketLen1 to PacketLen2. The bandwidth utilization is improved approximately by $((\text{PacketLen2} - \text{PacketLen1})/\text{PacketLen2}) \times 100\%$.

Since some error resilience mechanisms have been used in PFGS, the distortion for packet loss may just affect the slice. On the encoder side, distortion for each slice can be measured independently in advance. Let $Ds(Rs)$ stand for the source perceptual distortion-rate function. Our problem is finding the optimal FEC scheme (k_i, n_i) for different layers to minimize the end-to-end distortion D

$$\text{Minimize } D = D_S(R_S) \times P\left(0, \frac{R_S}{S_p}\right) + \sum_{i=1}^{m}\left(w_i \times \sum_{j=k_i}^{n_i}\left(D(i,j) \times \left(\sum_{l=n_i-k_i+1}^{n_i}\left(P(l,n_i)\left|\prod_{x=1}^{i-1}\sum_{y=0}^{k_x} P(y,n_x)\right.\right)\right)\right)\right) \quad (25)$$

where

$D(i,j)$ distortion that jth packet at ith layer is lost;
w_i distortion weight for the ith layer;
m number of layers to be transmitted.

Based on the decoder performance of PFGS, if the corresponding packet at any lower layers is lost, the packet of this layer is treated as lost whether or not it is received.

V. Simulation Results

In this section, we implement our proposed architecture and algorithms. The purpose of this section is to demonstrate the following:

1) MSTFP can track varying bandwidth well, be TCP-friendly, and have a smooth sending rate; and
2) our resource allocation scheme can adapt each media rate to the available estimated bandwidth with good perceptual media quality under varying network conditions.

We used the network simulator (NS) Version 2 [24] to study the performance of the MSTFP protocol and the resource allocation schemes for the MPEG-4 MVOs and multilayer scalable video. The network topology in the simulation consists of a single shared bottleneck link, as shown in Fig. 10. The senders reside on one side of the link and the receivers are on the other side. All links except the bottleneck link are sufficiently provisioned to ensure that any drops/delays occurred are only caused by congestion at the bottleneck link. All links are drop-tail links. In the simulations, the background traffic consists of infinite-duration TCP-based connections, e.g., TCP1, TCP2, and infinite-duration real-time-adaptive protocol (RAP) connections, e.g., RAP1, RAP2, proposed by Rejaie *et al.* that properly model the real-time Internet traffic [4].

A. Performance of MSTFP

We define the "friendliness" metrics as follows. Let k_m denote the total number of monitored MSTFP connections and k_t denote the total number of monitored TCP connections. We further denote the throughputs of MSTFP connections as $T_1^m, T_2^m, \ldots, T_{k_m}^m$ and those of TCP connections as $T_1^t, T_2^t, \ldots, T_{k_1}^t$. Then the average throughputs of MSTFP and TCP connections are, respectively, defined as

$$T_M = \frac{\sum_{i=1}^{k_m} T_i^m}{k_m}$$

and

$$T_T = \frac{\sum_{i=1}^{k_t} T_i^t}{k_t}. \quad (26)$$

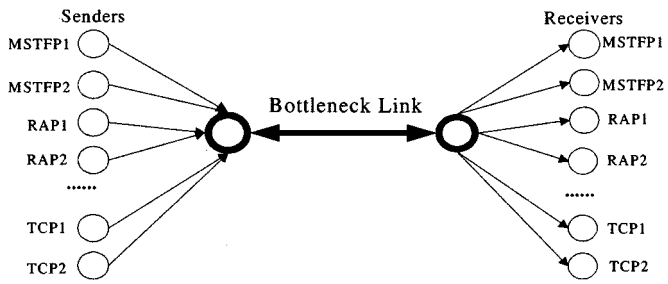

Fig. 10. Simulation topology.

The friendliness measure is defined as

$$F = \frac{T_M}{T_T}. \tag{27}$$

To evaluate "rate smoothness" of MSTFP connections, let $R^1_{m_i}, R^2_{m_i}, \ldots, R^{S_m}_{m_i}$, respectively, represent the sending rates at different time instances $1, 2, \ldots, s_m$ of the ith MSTFP connection and $R^1_{t_k}, R^2_{t_k}, \ldots, R^{S_t}_{t_k}$, respectively, represent the sending rates at different time instances $1, 2, \ldots, s_t$ of the kth TCP connection, then sending-rate variation of MSTFP and TCP connections are, respectively, defined as

$$\Delta_{M_i} = \sum_{j=1}^{S_m} |R^j_{m_i} - R^{j-1}_{m_i}|$$

and

$$\Delta_{T_k} = \sum_{j=1}^{S_t} |R^j_{t_k} - R^{j-1}_{t_k}|. \tag{28}$$

The smoothness measure is defined as

$$S = \frac{\Delta_{M_i}}{\Delta_{T_k}}. \tag{29}$$

Note that $S \leq 1$ means the ith MSTFP connection is smoother than the kth TCP connection.

Fig. 11 depicts the simulation results of throughput for different connections. The friendliness measures, calculated using (27), are 1.04 between MSTFP and TCP and 2.01 between RAP and TCP. It can be easily seen that our proposed MSTFP is friendlier to TCP than RAP.

The sending rates for different connections are illustrated in Fig. 12. The smoothness measures, calculated using (29), are 0.29 between MSTFP and TCP and 0.31 between MSTFP and RAP, respectively. From Figs. 11 and 12, we can see that our proposed MSTFP renders smoother sending rate than TCP and RAP.

B. Performance of Resource Allocation Scheme

To demonstrate the effectiveness of network bandwidth adaptation to varying network conditions, we use our proposed protocol, MSTFP, to track varying network bandwidth. Fig. 13 shows the tracking results using MSTFP. The upper figure shows bandwidth variation from 320 kb/s to 480 kb/s and the lower figure shows bandwidth variation from 80 kb/s 140 kb/s. It can be seen from Fig. 13 that given varying network bandwidth, MSTFP can track the available network bandwidth well.

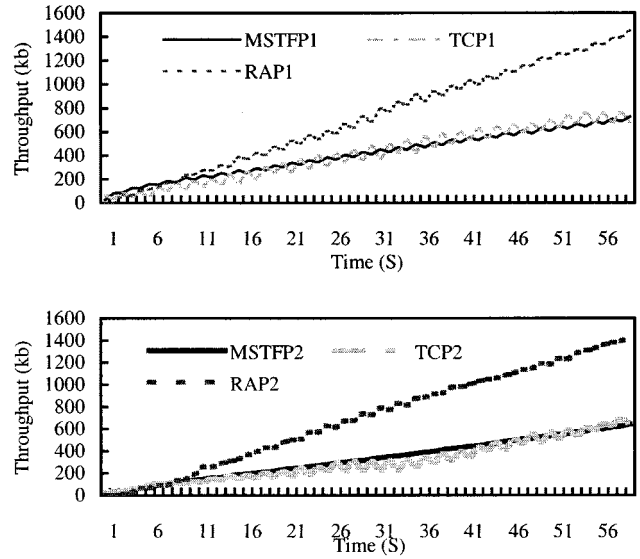

Fig. 11. Comparisons of throughput for different connections.

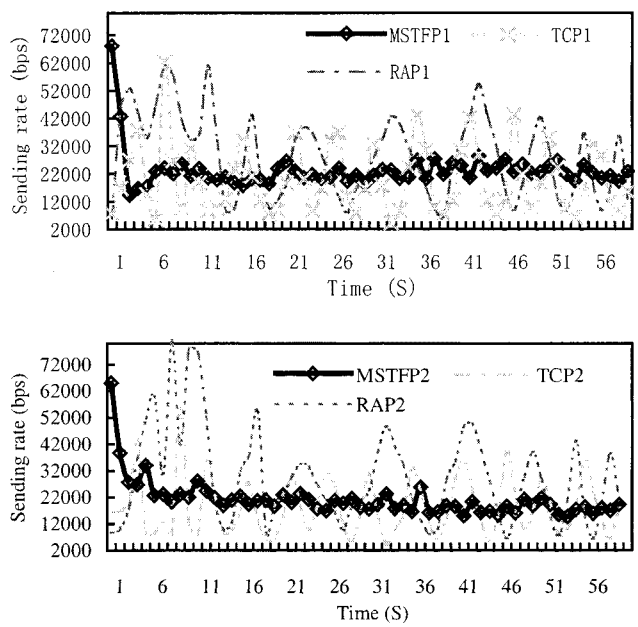

Fig. 12. Comparisons of sending rate for different connections.

1) Performance of the Network-Adaptive Rate Control Scheme: This simulation is to demonstrate the effectiveness of our proposed network-adaptive rate control scheme. In this simulation, we used a standard MPEG-4 codec with MVOs and the unrestricted motion vector modes to test two rate control schemes: 1) our network-adaptive MVOs rate control and 2) MPEG-4 standard MVOs rate control (Q2) without the knowledge of network bandwidth and packet loss ratio. We made a minor change on the frame-skipping in the MPEG-4 Q2 scheme to maintain all the objects of the same scene to a constant frame rate. In both cases, the first frame was intra-coded and the remaining frames were inter-coded. The testing video

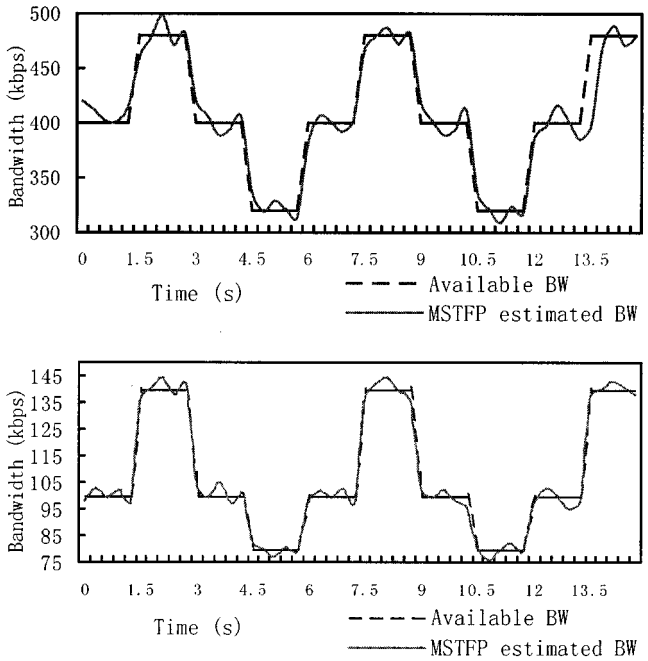

Fig. 13. Available network bandwidth and its estimation using MSTFP.

sequence is *News*. Fig. 14 illustrates the VOs for sequence *News*. The sequence consists of four objects: *1) background (obj0), 2) monitor (obj1), 3) newscasters (obj2), and 4) subtitle (obj3)*. Among these four objects, *background* and *subtitle* are less important than the other two. In our simulation, the weighting factors for these objects are, respectively, selected as 0.7, 1.2, 1.2, and 0.2 according to its own quality impact.

We use PSNR as a metric to measure video quality. For an 8-bit image with intensity values between 0 and 255, the PSNR is defined as PSNR $= 20\log_{10}(255/\text{RMSE})$, where RMSE stands for root mean squared error. Given an original $N \times M$ image f and the compressed or degraded image f', the RMSE can be calculated as

$$\text{RMSE} = \sqrt{\frac{1}{N \times M}\sum_{x=0}^{N-1}\sum_{y=0}^{M-1}[f(x,y) - f'(x,y)]^2}. \quad (30)$$

A fair comparison of PSNR between the two rate-control methods with different frame skipping is not trivial. A rate-control technique that skips more frames would typically spend more bits per coded frame and could easily have a very high average PSNR per coded frame. In the rate-control testing in MPEG-4, it was decided that, when a frame was skipped, the previous encoded frame should be used in the PSNR calculation because the decoder displays the previous encoded frame instead of the skipped one. We calculate the average PSNR using this approach in our simulations.

To demonstrate the effectiveness of our network-adaptive rate control scheme for different applications, we conducted simulations at two different bit-rates: high bit-rate and low bit-rate. At high bit-rate, *News*, was coded in CIF at a temporal resolution of 15 fps (frame per second) under network condition from 320 kb/s to 480 kb/s. At low bit-rate, it was coded in CIF at a temporal resolution of 10 fps under network condition from 80 kb/s to 140 kb/s. In both cases, the corresponding packet loss ratio varies from 0.5% to 5%.

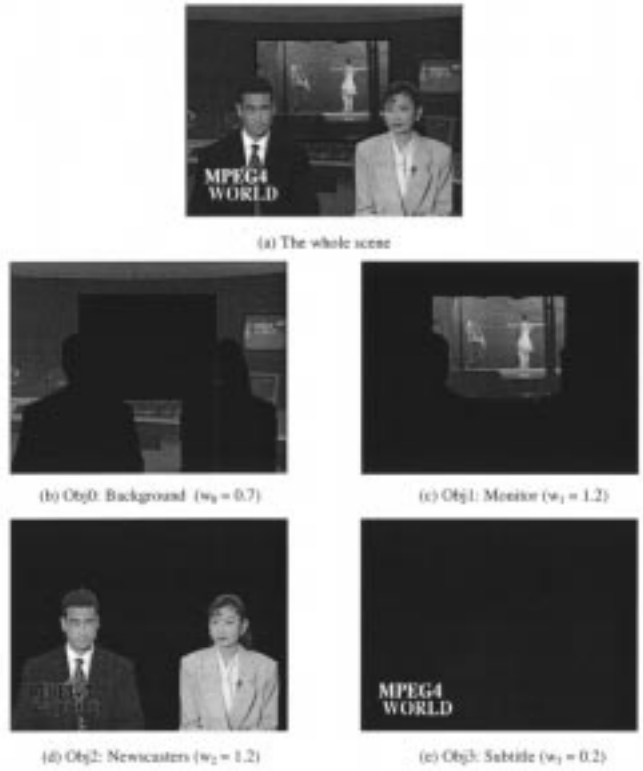

Fig. 14. Video objects 0, 1, 2, and 3 from sequence *News*.

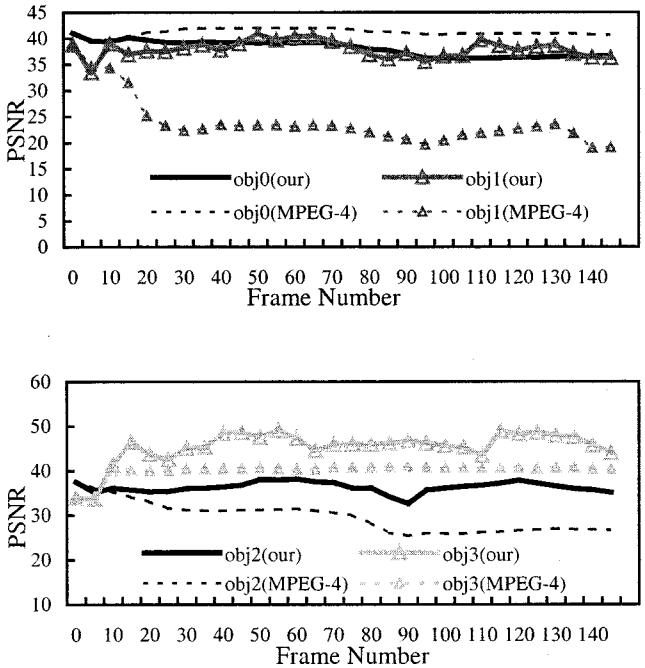

Fig. 15. PSNR comparisons of different objects using our approach and the MPEG-4 scheme.

Case 1: High Bit-Rate: We perform simulation using our network-adaptive rate control scheme at a high bit-rate for *News*. Fig. 15 shows comparison results of PSNR using our proposed rate control scheme and MPEG-4 Q2 scheme. In order to show

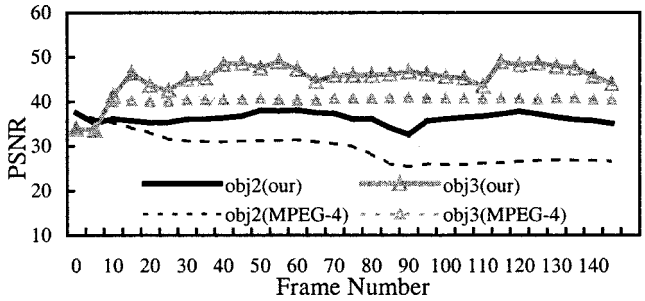

Fig. 16. Comparisons of PSNR for whole scene of *News* using our approach and the MPEG-4 Q2 scheme.

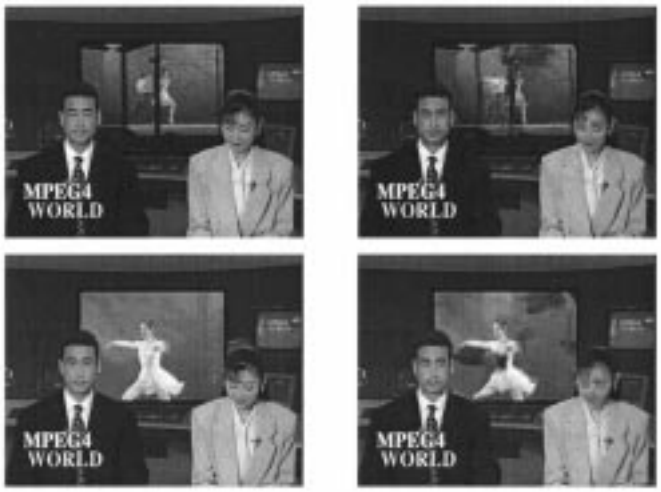

Fig. 17. Comparisons of the reconstructed 80th frame (top) and 99th frame (bottom) of sequence *News*. The images on the left are reconstructed by our scheme and those on the right are obtained by the MPEG-4 Q2 scheme.

the results clearly, we only show the simulation results of two objects in one figure. This figure shows that we obtain higher PSNR values except for *background*.

Fig. 16 shows comparison results of PSNR using our proposed rate control scheme (solid line) and the MPEG-4 Q2 scheme (dashed line). Having considered the varying network condition and different quality impact of each video object, we distributed available bits to more important objects such that the global quality is improved.

Fig. 17 shows comparisons of the reconstructed frames of sequence *News* using our rate control scheme and the MPEG-4 Q2 scheme. In this figure, the upper left image is the reconstructed 80th frame using our rate control scheme; while the upper right image shows the reconstructed 80th frame using the MPEG-4 Q2 scheme. The lower left image of the figure shows the reconstructed 90th frame using our proposed rate-control approach and the lower right image is the reconstructed frame using MPEG-4 Q2 scheme.

Table I depicts the comparison results of average PSNR for each object and whole sequence using our proposed network-adaptive rate control scheme and the MPEG-4 Q2 scheme.

From Figs. 15–17 and Table I, it can be seen that our proposed network-adaptive rate control scheme outperforms the MPEG-4 Q2 scheme at high bit-rate both subjectively and objectively. Note that we get lower PSNR for *object 0* in Fig. 15 and Table I, because *object 0* is background and we assign fewer bits to it in our proposed rate control scheme to achieve better quality for the foreground objects.

Case 2: Low Bit-Rate: We perform simulation using our network-adaptive rate control scheme at low bit-rate. Fig. 18 shows the comparison results of PSNR for *News* sequence at low bit-rate using our proposed rate control scheme and the MPEG-4 Q2 scheme. In order to show the results clearly, we only show the simulation results of two objects in one figure. It can be seen from Fig. 18 that we obtain higher PSNR values for *monitor* and *newscasters*; while obtaining lower PSNR values for *subtitle* and *background*. Since *monitor* and *newscasters* are foreground objects, which are more sensitive to users perceptual quality, we achieve higher PSNR in the global scene, as depicted in Fig. 19.

Fig. 19 shows comparison results of PSNR using our proposed rate control scheme (solid line) and the MPEG-4 Q2 scheme (dashed line). In both cases, the PSNR values of the tested video drops sharply due to frame skipping. We can see that the number of frames skipped is reduced using our scheme. Notice that at the initialization stage, our scheme gets poorer quality than the MPEG-4 Q2 scheme. Actually, there are several frames skipped in that stage. This is because the joint global buffer control is adopted for frame skipping in our rate control scheme. When the buffer occupancy is over a certain margin, frame is skipped and buffer occupancy is decreased by the frame size rather than set to 0 in the MPEG-4 Q2 scheme. This frame-skipping policy will increase the stability of network overload, but results in a longer convergence period. One possible approach for solving this problem is to use a pre-fetching buffer.

Table II depicts comparison results of the average PSNR for each object and whole sequence using our proposed network-adaptive rate control scheme and the MPEG-4 Q2 scheme.

From Figs. 18 and 19 and Table II, it can be seen that our proposed network-adaptive rate control obtains better results than the MPEG-4 Q2 scheme at low bit-rate. Note that we get lower average PSNR for *object 0* and *object 3*, which are background and text, respectively. This is because we assign fewer bits to them in our rate control scheme to achieve better quality for the foreground objects.

2) Performance of the Network-Adaptive Bit Allocation Scheme: This simulation is to demonstrate effectiveness of our proposed network-adaptive bit allocation scheme for PFGS. In this simulation we tested: 1) our network-adaptive bit allocation scheme and 2) PFGS without knowledge of network bandwidth and packet loss ratio. In both cases, the first frame was intra-coded and the remaining frames were inter-coded. The testing video sequence is *Foreman*, which is coded in CIF at a temporal resolution of 15 fps. We conducted simulations under network condition from 320 kb/s to 480 kb/s. The corresponding packet loss ratio varies from 0.5% to 5%.

Fig. 20 shows comparison results of PSNR for *Foreman* sequence using our proposed bit allocation scheme and the PFGS scheme. It can be seen that, overall, our scheme outperforms the PFGS scheme, especially in the packet loss cases. It can also be seen that the video quality achieved by our approach changes

TABLE I
AVERAGE PSNR COMPARISONS FOR OBJECTS AND WHOLE SEQUENCE OF *NEWS* AT HIGH BIT-RATE

Rate control Scheme	Average PSNR				
	Obj0	Obj1	Obj2	Obj3	Sequence
MSTFP + adaptive rate allocation	38.1	36.9	36.2	43.2	37.3
MPEG-4 rate control	41.2	32.2	32.6	40.7	34.2

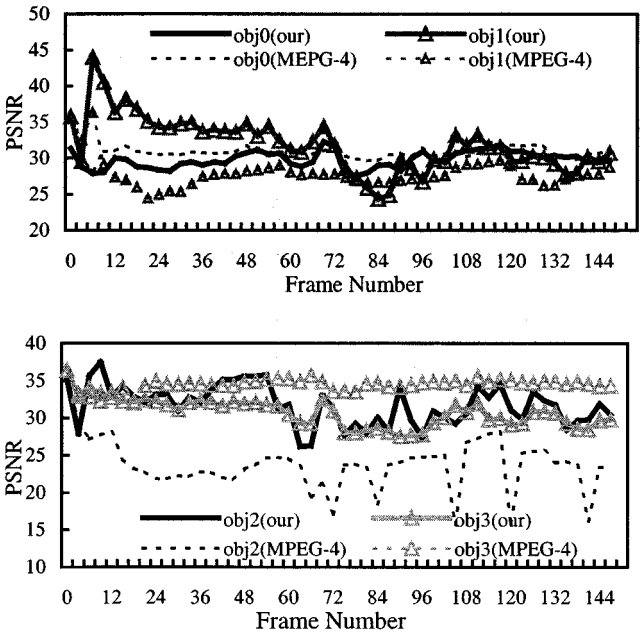

Fig. 18. Comparisons of PSNR for different objects using our approach and the MPEG-4 scheme.

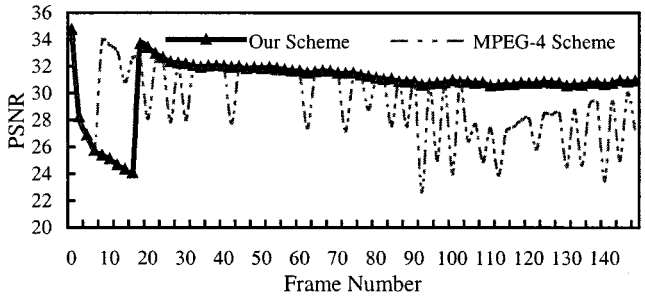

Fig. 19. Comparison of PSNR for whole scene of *News* using our approach and the MPEG-4 Q2 scheme.

more smoothly. In contrast, the quality of the reconstructed sequences using the PFGS scheme is more fluctuant. Note that in Fig. 20 for few frames, the PSNR values for the PFGS scheme are higher than ours. This is because we spend some bits for error protection from the source in the PFGS scheme according to its layer priority and estimated network bandwidth, while for the PFGS scheme, all the bits are allocated to the source. It can be seen from Fig. 20 that when packet loss does occur, we achieve significantly better PSNR compared to the PFGS scheme. In addition, since errors due to network congestion occurred randomly and are reflected at different frame locations, we get slightly different performance in terms of PSNR in the same simulation condition.

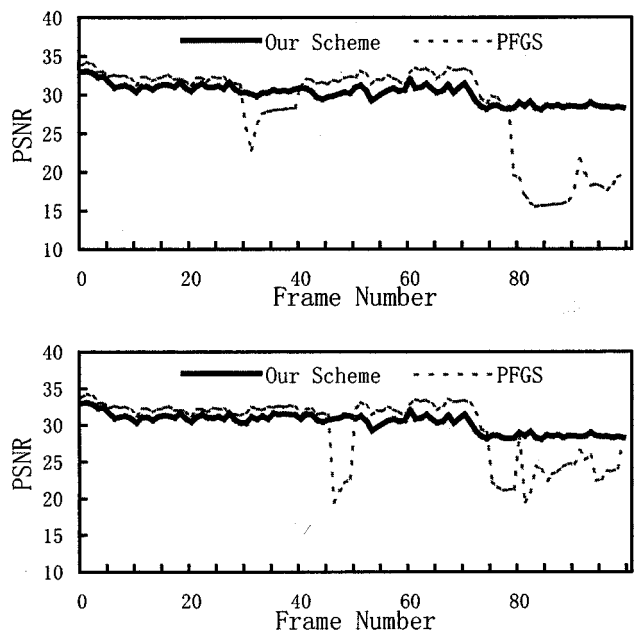

Fig. 20. Comparisons of PSNR using our approach and the PFGS scheme. Packet loss occurs at the 29th and 76th frames (top). Packet loss occurs at the 46th and 74th frames (bottom).

Table III depicts comparison results of average PSNR for whole sequence and average overhead using our proposed network-adaptive bit allocation scheme and the PFGS scheme.

Fig. 21 shows two reconstructed frames of sequence *Foreman* using our bit allocation scheme and the PFGS scheme. In this figure, the upper left image is the reconstructed 29th frame using our bit allocation scheme; while the upper right image shows the reconstructed 29th frame using the PFGS scheme. The lower left image shows the reconstructed 46th frame using our proposed bit allocation approach and the lower right image is the reconstructed frame using the PFGS scheme.

From Figs. 20 and 21 and Table III, it can be seen that our proposed network-adaptive bit allocation approach obtains better results than the PFGS scheme under packet loss network both subjectively and objectively.

In summary, the simulation results presented in this section demonstrate that:

1) our MSTFP can keep good track of network bandwidth through forward estimation and information feedback control. Moreover, it is very TCP-friendly and smoothes sending rate;
2) our resource allocation approach can achieve better overall quality than the MPEG4 Q2 scheme for MVO at low and high bit-rates in packet-loss environment;

TABLE II
AVERAGE PSNR COMPARISONS FOR OBJECTS AND WHOLE SEQUENCE OF *NEWS* AT LOW BIT-RATE

Rate control Scheme	Skipped frame	Average PSNR				
		Obj0	Obj1	Obj2	Obj3	Sequence
MSTFP + adaptive Rate allocation	8	29.8	30.1	32.2	31.2	30.8
MPEG-4 rate control	25	30.9	28.6	24.6	34.7	29.4

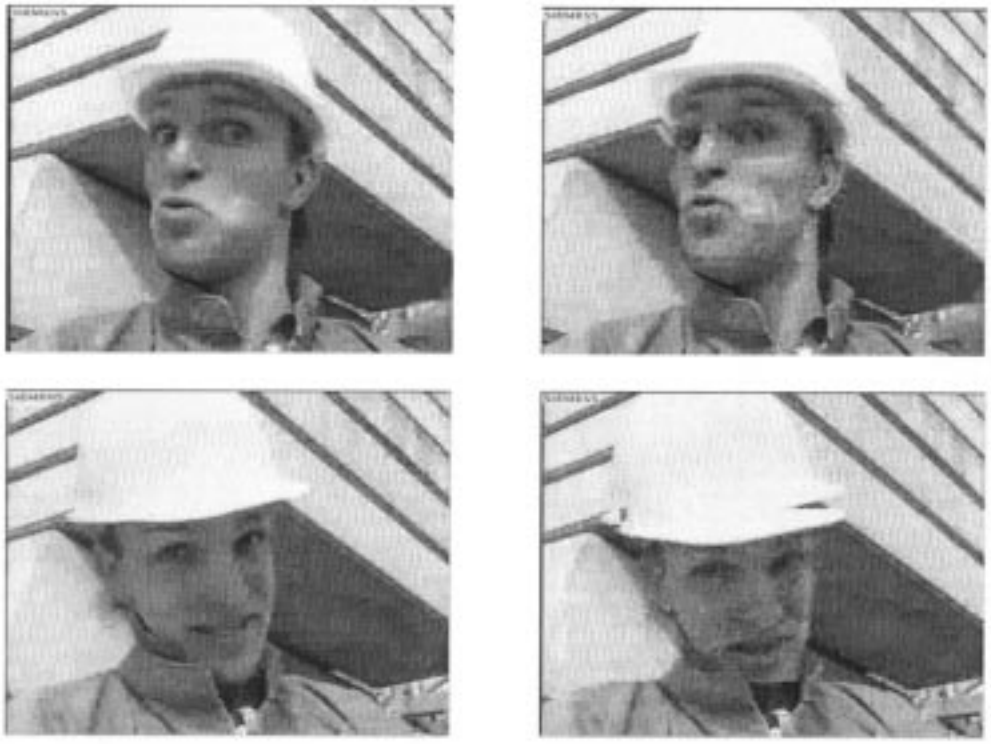

Fig. 21. Comparisons of the reconstructed 29th frame (top) and 46th frame (bottom) of sequence *Foreman*. The images on the left are reconstructed by our scheme and those on the right are reconstructed by the PFGS scheme.

TABLE III
COMPARISONS OF AVERAGE PSNR AND OVERHEAD FOR *FOREMAN*

Sequence	Tested schemes	Average PSNR	Average overhead (%)
Foreman (320kb/s ~ 480kb/s)	Our approach (1)	30.16	8.25
	PFGS (1)	28.52	0
	Our approach (2)	30.33	8.2
	PFGS (2)	29.25	0

3) our resource allocation approach can achieve better overall quality than that in multilayer scalable codec (PFGS) under packet-loss environment.

VI. CONCLUSIONS AND DISCUSSIONS

This paper addresses how to allocate bits among different media types/objects based on the estimated network available bandwidth so as to obtain good perceptual quality for multimedia streaming over the Internet from an end-to-end perspective. The main contributions of this paper are summarized as follows.

- We proposed a new multimedia streaming TCP-friendly protocol (MSTFP) that combines forward estimation of network condition with feedback control for optimal network status tracking. MSTFP is able to adaptively estimate the network bandwidth and reduce sending-rate variation.
- We designed a global buffer control algorithm to synchronize various media types and achieve R-D-based optimal bit allocation according to network conditions.
- We proposed a novel quality-adaptation resource allocation scheme to periodically estimate the available bandwidth using MSTFP. Combining the estimation of available network bandwidth with the media characteristics, our proposed resource allocation scheme for multiple media streams achieves significant improvement in the end-to-end QoS.

Simulations using MPEG-4 MoMuSys codec with MVOs and PFGS codec with multilayer demonstrated that our proposed MSTFP adapts fairly well to network bandwidth variations and that our proposed resource allocation scheme achieves good

end-to-end visual quality across relatively congested connections at both high and low rates.

In this paper, we used the maximal transmission unit (MTU) of the network as the packet size to ensure the efficiency of network utilization. Studying the relation between packet loss ratio and packetization length is one of the future works. In addition, we studied the bit allocation between source coding and channel FEC based on R-D. A study of bit allocation among source coding, FEC, and ARQ for video streaming over the Internet is another interesting topic for future work [25].

ACKNOWLEDGMENT

The authors would like to thank A. Vetro from the Advanced Television Laboratory of Mitsubishi Electric Information Technology Center and Dr. S. Li and Dr. F. Wu from Microsoft Research for providing MPEG-4 MoMuSys codec and PFGS, respectively, for the simulations. The authors would also like to thank G. Wang from Tsinghua University for assistance on the PFGS simulation in Section V. Prof. Z. Xiong from Texas A&M University and Prof. S.-H. G. Chan from Hong Kong Univeristy of Science and Technology are acknowledged for reading parts of the draft. Finally, the authors wish to thank the anonymous reviewers for their comments and suggestions that helped to improve the presentation of this paper.

REFERENCES

[1] R. Braden, L. Zhang, and S. Berson *et al.*, "Resource reservation protocol (RSVP)—Ver. 1 functional specification,", RFC 2205, 1997.

[2] H. R. Shao, W. Zhu, and Y. Q. Zhang, "User-aware object-based video communication over next generation internet," *Signal Process. Image Commun.*, 2001, to be published.

[3] J. Shin, J. W. Kim, and C.-C. J. Kuo, "Content-based packet video forwarding mechanism in differentiated service networks," in *10th Int. Workshop Packet Video*, Sadinia, Italy, May 2000.

[4] R. Rejaie, M. Handley, and D. Estrin, "Quality adaptation for congestion controlled video playback over the internet," in *Proc. SIGCOMM'99.*, Aug. 1999, pp. 189–200.

[5] S. Jacobs and A. Eleftheriadis, "Streaming video using TCP flow control and dynamic rate shaping," *J. Vis. Commun. Image Represent.*, vol. 9, no. 3, pp. 211–222, Sept. 1998.

[6] I. Rhee, V. Ozdemir, and Y. Yi, "TEAR: TCP emulation at receivers—Flow control for multimedia streaming," Dept. Computer Science, North Carolina State Univ., Raleigh, 2000.

[7] S. Cen, C. Pu, and J. Walpole, "Flow and congestion control for internet streaming applications," *Proc. Multimedia Computing and Networking*, pp. 250–264, Jan. 1998.

[8] J. Padhye, J. Kurose, and D. Towsley *et al.*, "A model based TCP-friendly rate control protocol," in *Proc. 9th Int. Workshop NOSSDAV'99*, June 1999.

[9] W. Tan and A. Zakhor, "Real-time internet video using error resilient scalable compression and TCP-friendly transport protocol," *IEEE Trans. Multimedia*, vol. 1, pp. 172–186, June 1999.

[10] D. Disalem and H. Schulzrinne, "The loss-delay based adjustment algorithm: A TCP-friendly adaptation scheme," in *Proc. Int. Workshop NOSSDAV'98*, July 1998.

[11] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," in *Proc. SIGCOMM'98*, Aug. 1998, pp. 303–314.

[12] L. Wang and A. Vincent, "Joint coding for multi-programs transmission," in *Proc. IEEE Int. Conf. Image Proceedings*, Sept. 1996.

[13] A. Vetro, H. F. Sun, and Y. Wang, "MPEG-4 rate control for multiple video objects," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 186–199, Feb. 1999.

[14] M. Eckert and J. I. Ronda, "Bit-rate allocation in multi-object video coding,", Dublin, Ireland, ISO/IEC JTC1/SC29/WG11 MPEG98/m3757.

[15] M. Yajnik, S. Moon, and J. Kurose *et al.*, "Measurement and modeling of the temporal dependence in packet loss," UMASS CMPSCI, 98-78, 1998.

[16] S. Floyd, M. Handley, and J. Padhye *et al.*. Equation based congestion control for unicast applications. [Online]. Available: http://www.aciri.org/tfrc

[17] R. Rejaie, M. Handley, and D. Estrin, "An end-to-end rate-based congestion control mechanism for realtime streams in the internet," in *Proc. INFOCOM 99*, Mar. 1999.

[18] O. Verscheure, P. Frossard, and M. Hamdi, "MPEG-2 video services over packet networks: joint effect of encoding rate and data loss on user-oriented QoS," in *Proc. NOSSDAV'98*, July 1998, pp. 257–264.

[19] S. Li, F. Wu, and Y.-Q. Zhang, "Study of a new approach to improve FGS video coding efficiency,", Maui, HI, ISO/IEC JTC1/SC29/WG11, MPEG99/m5583, Dec. 1999.

[20] T. Chiang and Y. Q. Zhang, "A new rate control scheme using quadratic rate-distortion modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 246–250, Feb. 1997.

[21] "Text of ISO/IEC 14 496–2 MPEG-4 video FGS VM 2.0," MPEG Video Group , Melbourne, Doc. ISO/IEC JTC1/SC29/WG11 N2926, Oct. 1999.

[22] U. Horn, K. Stuhlmuller, and M. Link *et al.*, "Robust internet video transmission based on scalable coding and unequal error protection," *Image Commun., Special Issue on Real-Time Video Over the Internet*, vol. 15, pp. 77–94, Sept. 1999.

[23] D. W. Redmill and N. G. Kingsbury, "The EREC: an error-resilient technique for coding variable-length blocks of data," *IEEE Trans. Image Processing*, vol. 5, pp. 365–374, April 1996.

[24] The network simulator (NS) [Online]. Available: http://www-mash.cs.berkeley.edu/ns/ns.html

[25] Q. Zhang, W. Zhu, G. J. Wang, and Y.-Q. Zhang, "Resource allocation with adaptive QoS for multimedia transmission over W-CDMA channels," in *Proc. IEEE'WCNC*, Chicago, IL, Sept. 2000.

Qian Zhang (M'00) received the B.S., M.S., and Ph.D. degrees from Wu Han University, China, in 1994, 1996, and 1999, respectively, all in computer science.

In 1999, she joined Internet Media Group at Microscoft Research China, Beijing, where she is currently a Researcher in the Wireless and Networking Group. Her current research interests include wireless Internet multimedia, 3G wireless communications, and networking. Currently, she is actively participating in TCP/IP header compression over wireless in ROHC WG in IETF.

Wenwu Zhu (S'91–M'96–SM'01) received the B.E. and M.E. degrees from National University of Science and Technology, Changsha, China, in 1985 and 1988, respectively, the M.S. degree in electrical engineering from Illinois Institute of Technology, Chicago, and the Ph.D. degree in electrical engineering from Polytechnic University, Brooklyn, NY, in 1993 and 1996, respectively. From August 1988 to December 1990, he was with the Graduate School, University of Science and Technology of China (USTC) and the Institute of Electronics, Academia Sinica (Chinese Academy of Sciences), Beijing.

In October 1999, he joined Microsoft Research China, Beijing, as a Researcher. He was subsequently promoted to Project Leader and is currently Research Manager of the Wireless and Networking Group. Prior to joining Microsoft, he was with Bell Labs, Lucent Technology, where he was a Member of the Technical Staff from July 1996 to October 1999. While at Bell Labs, he performed research and development in the area of video conferencing, Internet video, and video over IP. He has published over 70 refereed papers. He is the inventor of more than a dozen pending patents. His current research interest is in the area of wireless communication/networking and wireless/Internet multimedia.

Dr. Zhu is a Member of Eta Kappa Nu. He is also a Member of the Visual Processing and Communication Technical Committee and Multimedia System and Application Technical Committee in IEEE Circuits and Systems Society. He is Guest Editor for Special Issues on Streaming Video and Wireless Video in IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.

Ya-Qin Zhang (S'87–M'90–SM'93–F'98) received the B.S. and M.S. degrees in electrical engineering from the University of Science and Technology of China (USTC), Beijing, in 1983 and 1985, respectively, and the Ph.D degree in electrical engineering from George Washington University, Washington, DC, in 1989. He graduated from the Executive Business Program at Harvard University, Cambridge, MA, in 1994.

Currently, he is the Managing Director of Microsoft Research China, Beijing, leaving his post as the Director of Multimedia Technology Laboratory at Sarnoff Corporation, Princeton, NJ (formerly David Sarnoff Research Center and RCA Laboratories). He has been engaged in research and commercialization of MPEG2/DTV, MPEG4/VLBR, and multimedia information technologies. He was with GTE Laboratories Inc., Waltham, MA, from 1989 to 1994. He has authored and coauthored over 200 refereed papers in leading international conferences and journals. He has been granted over 40 U.S. patents in digital video, Internet, multimedia, and wireless and satellite communications. Many of the technologies he developed with his team have become the basis for start-up ventures, commercial products, and international standards. He sits on the Board of Directors of five "high-tech" companies.

Dr. Zhang has received numerous awards, including several industry technical achievement awards and IEEE awards such as the CAS Jubilee Golden Medal, the Richard Merwin Award, and the Best Paper Award. He was named "Research Engineer of the Year" in 1997 by New Jersey Engineering Council. He received the prestigious national award as "The Outstanding Young Electrical Engineering of 1998," given annually to one electrical engineer in the U.S. He served as the Editor-In-Chief for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY from July 1997 to July 1999. He was the Chairman of the Visual Signal Processing and Communications Technical Committee of IEEE Circuits and Systems Society. He serves on the editorial boards of seven other professional journals and over a dozen conference committees. He has been a key contributor to the ISO/MPEG and ITU standardization efforts in digital video and multimedia.