ELSEVIER

Signal Processing: *Image Communication* 16 (2001) 763–784

SIGNAL PROCESSING:
IMAGE
COMMUNICATION

www.elsevier.nl/locate/image

User-aware object-based video transmission over the next generation Internet

Huai-Rong Shao*, Wenwu Zhu, Ya-Qin Zhang

Microsoft Research, China, 3F, Beijing Sigma Center, No. 49 Zhichun Road, Haidian District, Beijing, 100080, China

Abstract

In this paper, we present a new user-aware adaptive object-based video transmission approach to heterogeneous users over the next generation Internet. Firstly, we describe a new transport framework for complex multimedia applications over the next generation Internet, which provides differentiation functionality within one IP session as well as among different IP sessions. It includes application-aware intelligent resource control at the edge of the network, fast transcoding and signaling in the network. Secondly, we propose a new bitstream classification, prioritization and packetization scheme in which different types of data such as shape, motion and texture are reassembled, assigned to different priority classes, and packetized separately based on their priorities. Thirdly, we present a simple but effective mechanism of object-based dynamic rate control and adaptation by selectively dropping packets in conjunction with differentiated services (Diffserv) to minimize the end-to-end quality distortion. Finally, we perform the queuing analysis for our mechanism and explore how to extend our approach to the multicast case. Experimental results demonstrate effectiveness of our proposed approach. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Object-based video; Quality of service; Video transport; Differentiated services; Multicast; Next generation Internet; User interactivity

1. Introduction

Over the past several years, as the speed of computer and network increased dramatically, more and more networked multimedia applications proliferated rapidly. Among all kinds of multimedia techniques, multimedia streaming, as it can be used in many important areas such as distance learning and video on demand, has attracted enormous interests from both industry and academia. With the success of the Internet and the emerging multimedia communication era, the new international standard, MPEG-4 [20], is posed to address the new demands that arise in which more and more audiovisual material is exchanged in digital format. The key innovation in MPEG-4 is the introduction of *objects* as the smallest accessible units compared to the traditional frame-based approach. These *objects* can be auditive or visual, static or dynamic, natural or synthetic. Thus, MPEG-4 can offer new interactivities for end users of multimedia streaming applications. For instance, besides VCR functionalities that are usually provided by frame-based video, object-based video can allow users to

*Corresponding author. Tel.: + 86-10-6261-7711/3138; fax: + 86-10-6255-5337.
E-mail address: hrshao@microsoft.com (H.-R. Shao), wwzhu@microsoft.com (W. Zhu).

PII: S0923-5965(01)00006-6

interact with video contents (video objects) dynamically, such as object moving, object zooming/out, object adding/deleting, and object quality enhancing/degrading. However, the interactivities may bring out the new network control issues: how to adapt the bit-rate of each object in the same scene and how to assign different transmission priorities to different objects according to user's dynamic interactions.

Current IP networks offer the so-called best effort (BE) services, which do not make any service quality commitment. However, most multimedia applications are delay/loss sensitive, thereby requiring quality of service (QoS) support from the network. As a result, the current Internet is becoming increasingly inadequate to support the service quality of multimedia streaming applications. To support QoS in the Internet, the IETF has defined two architectures: the integrated services (Intserv) [21], and the differentiated services (Diffserv) [2,4–6]. The integrated services model provides per-flow QoS guarantee and resource reservation protocol (RSVP) was suggested for resource admission control and resource allocation. However, it is very complicated for backbone routers to maintain states of thousands of classes. On the other hand, differentiated services model gives a class-based solution to support relative QoS. In differentiated services, packets are divided into different QoS classes and forwarded as different priorities. The QoS class of each packet is specified by IPv4 type of service (TOS) byte or IPv6 Traffic Class byte. Since it is highly scalable and relatively simple, differentiated services model may be promising to dominate the backbone of the next generation Internet in the near future.

Compressed video over IP networks roughly consists of packetization, rate adaptation, and transport. Previous efforts on packetization for video applications over the Internet include [25] for H.261, [29] for H.263, [3] for H.263+, [9] for MPEG1/2, and [27] for MPEG4. However, all these works do not differentiate information types within a frame or an object. On rate control, traditional functionality of source rate control is mainly implemented at the encoder by means of quantization step-size adjustment [26]. Obviously, it is not suitable for pre-stored video distribution if transcoding is not introduced. Layered or scalable coding and multicast [10,14,19,20,23] alleviate this problem since scalable or layer coding is capable of gracefully coping with the bandwidth fluctuations in the Internet. But multiple network sessions are needed for one video program and it is complicated to maintain the synchronization among layers. In addition, no packet-level transmission rate control scheme at edge of network and end system is addressed. On transport of prioritized video over the next generation Internet, Shin et al. proposed a mechanism between video applications and Diffserv network to achieve enhanced end-to-end video quality [22]. They discussed the mapping mechanism between application packets and Diffserv classes. However, their target is not object-based application with user interactivity. Most of the previous work on Diffserv [2,4–6,7,11] only considered how to differentiate media streams, i.e., different packets in the same IP session have the same communication characteristics. Information with different transmission requirements is transported by different IP sessions. For example, in layered coding and transmission [14,19,20], multiple independent communication channels are required for different layers. Two problems arise due to multiple channel approach: the synchronization between channels and the network state maintenance.

To address the above problems, in this paper we propose a new transport framework for complex multimedia applications such as MPEG-4 with multiple objects and user interactivity over the next generation Internet. It includes both the end systems of the hosts and the communication network. This framework combines differentiated services and integrated services [1]. That means per-conversion admission control using RSVP is supported at the edge of the network but aggregated traffic handling is implemented within the backbone. Our proposed framework has three new features: (1) differentiation functionality within one IP session as well as among different IP sessions; (2) application-aware intelligent resource control and management at the edge of the network; (3) multimedia processing agent (MPA) on the bottlenecks within domains and at the edge between domains. Because of the above new features, adaptive video applications can take full advantage of the Diffserv to achieve optimal end-to-end unicast and multicast quality. More specifically, we have the following:

- At the end system, we present a new bitstream classification, prioritization and packetization scheme in which different types of data such as shape, motion and texture are re-assembled, assigned to different priority classes, and packetized into different classes of network packets provided by Diffserv. Our scheme distinguishes not only different kinds of frames, but also different types of information within the same frame. In addition, different types of information are packetized into different classes of network packets with various transmission priorities. That is, more important compressed information is put into higher priority packets and less important information into lower priority ones.
- We address a dynamic object-based rate control and adaptation in conjunction with Diffserv to minimize the end-to-end distortion. By taking advantage of Diffserv, dynamic transmission rate control can be easily implemented by selectively dropping lower priority packets at the sender or intermediate network nodes.
- We introduce the user awareness in MPEG-4 video streaming to provide object-based user interactivity. A user can interact with the video player or the server [12] in several ways such as mouse clicking, mouse moving, fast forwarding/backward, object zoom-in/zoom-out, object adding or deleting.

The rest of this paper is organized as follows. Section 2 describes our new user-aware adaptive object-based video transmission framework for the next generation Internet. Section 3 presents our new information re-organization, prioritization and packetization scheme. In Section 4 we discuss the object-based rate control and interactivity issue with Diffserv. Section 5 shows the performance evaluation results. In Section 6, the extension of our approach to multicast case is introduced. Finally, Section 7 concludes the paper.

2. Framework of user-aware object-based video transmission over the next generation Internet

Fig. 1 depicts a general framework for complex multimedia applications such as MPEG-4 with multiple objects over the next generation Internet. It includes both the end systems of the hosts and the communication network. For the end-system part, there are two enhancements in our framework: differentiation functionality within one IP session as well as application-aware intelligent resource control and management. The communication network combines differentiated services and integrated services. In this framework, although signaling messages traverse the network from end to end, they are processed only in the hosts and in

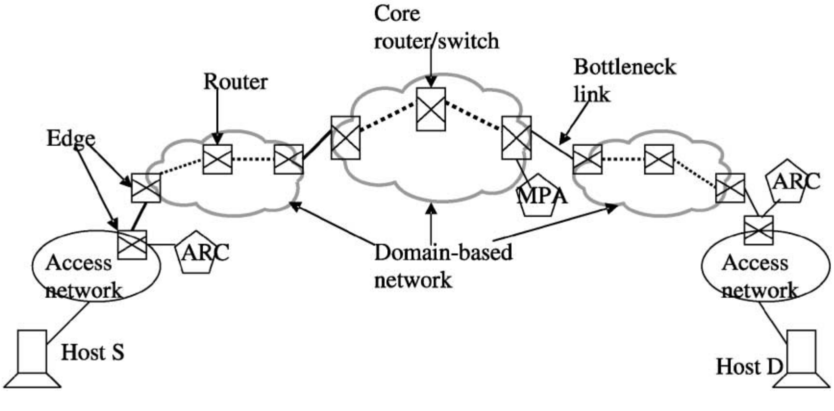


Fig. 1. A general framework for adaptive multimedia applications.

the router that is appointed as admission control agent for the routing network. Most routers in the core of the routing network apply aggregate traffic handling and do not process signaling messages. This model of per-conversation signaling at the edge of the network and aggregate traffic handling in the core yields a good tradeoff between complexity and efficiency. In addition, we introduce two new components within the communication network: *multimedia processing agent* (MPA) and *application-aware resource controller* (ARC). The MPA is a component in a router or gateway, or a server attached to a router, who is responsible for adapting the stream rate to network state. In our framework, ARC has the responsibility of handling application-based signaling according to the application-aware resource control, rather than per-flow signaling. Therefore, the traffic overload of signaling is greatly reduced in our framework.

The ARC is a logical concept. It can be implemented in a proxy server or an edge router, or an independent server between the access network and the backbone network. It can provide application-aware intelligent resource coordination among different users, applications, and traffic classes. Compared to the traditional admission control mechanism that does not consider the semantic relationship among different priority classes within one application, ARCs use control messages to interact with the server and the receivers, dynamically allocate resources to different applications and different classes according to network state and user's preferences. Under the condition of lacking network bandwidth, ARC can allocate more desired bandwidth to the object where better perceptual visual quality is desired by a receiver than other objects. The ARC has the following responsibilities:

- Receive application admission request from sending users.
- Interact with remote ARC(s) on the access networks of the receivers or MPA(s) on the forwarding path to exchange RSVP signaling messages.
- Give feedback to senders whether to admit the application.
- Aggregate traffic from local users and dynamically map them to Diffserv classes.
- Coordinate bandwidth allocation among multiple applications and video object flows, and dynamically map different priorities information to different network classes.

The ARC uses the adaptive transmission scheme proposed in Section 4 to adapt the flow rate to network state and users' requirements.

The MPA is responsible for adapting the stream rate to network state and user requirements by means of transcoding or other mechanisms such as selective packet dropping. This solution allows us to place the MPA on the nodes that connect to network bottlenecks. There can be multiple MPAs along the path from a server to a client. The MPA has the following responsibilities:

- Receive video object streams from the server or the previous MPA.
- Filter received video object streams by selectively discarding packets with lower priorities.
- Send filtered video object streams to clients or next MPA(s).
- Receive requests from clients or next MPA(s).
- Act upon requests or generate a combined request from multiple clients or next MPA(s) and forward it to previous MPA(s).
- Coordinate bandwidth allocation among multiple video object streams, perform application-aware admission control, and dynamic bandwidth re-allocation according to users' dynamic interactions.

2.1. *Differentiation within one IP session*

For complex multimedia applications such as MPEG4 programs, usually there are multiple objects of the same media or different media. If these objects have different QoS requirements, generally each object is served by an individual IP session or even several sessions in the case of multiple layers. In general, it is difficult for the end-system and network to maintain so many sessions for one application. However, if differential marking within a flow is supported, layers belonging to the same object or different objects can be multiplexed into one IP session. A possible problem resulting from differentiation within one flow is disorder of packets within the same flow. However, this problem exists even without the function of differentiation within one flow because of the connectionless characteristic of IP protocol. To support the new

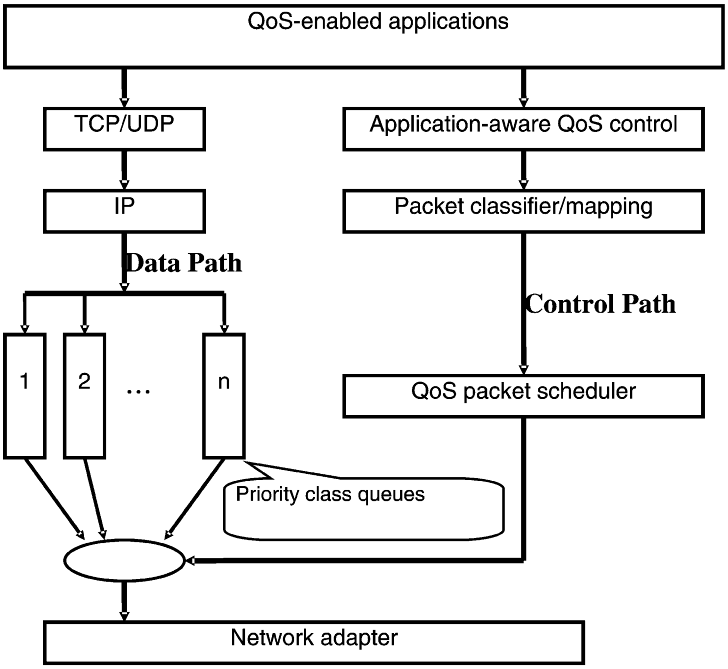


Fig. 2. Enhanced communication protocol stack for Diffserv.

differentiation functionality, an extension needs to be made on the protocol stack of the end-system and ARC. Fig. 2 shows the diagram of the communication protocol stack. By default, packets are marked based on a mapping from the service type associated with a flow. All packets within one flow have the same marking value. We propose a new marker mapping mechanism in the host protocol stack to support differentiation within one flow. We introduce the multiple queue mechanism at the end-system and each queue buffers packets with a particular priority. When the IP header is added at the IP layer, the priority is mapped to the Diffserv Code Point (DSCP) byte.

2.2. Network-aware end-system

Fig. 3 shows the architecture of the streaming server with intelligent resource control and management for multimedia applications. This architecture considers the transmission of multiple-object video programs and other types of media such as audio and data. Each video object is compressed first and the corresponding elementary stream is generated. Then information within each elementary stream is classified based on importance and assembled into packets with different Diffserv classes. Network Monitor is responsible for estimating the available network bandwidth dynamically through probing or feedback-based approach. Packet Forwarder forwards the packets to the network. We do not discuss these two blocks in detail in this work. The other functional components are described as follows.

- Priority Mapping and Marking Agent: This component is responsible for the interaction between applications and the Diffserv networks. It assigns DSCP marks to packets and maps them to the corresponding Diffserv classes.
- Application Collaborator: The *Application Collaborator* is responsible for resource coordination among multiple objects within one application and among multiple applications. It receives information from Application Profiles, Remote

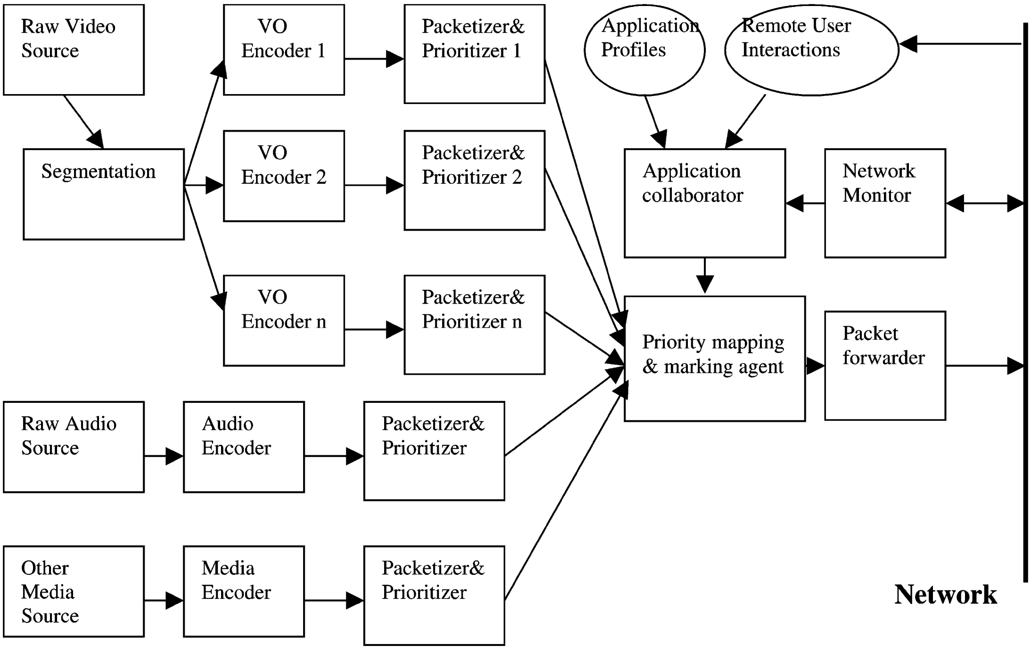


Fig. 3. Architecture in the end-system.

Users Interactions, and Network Monitor to make the decision. In addition, the Application Collaborator tells us how to map packet priorities from individual encoders into network classes. The receivers can interact with the server through user-level signaling.

- Application Profiles: This component records the semantic information of the applications such as which media and flows are included in an application and their relative importance levels.
- Remote User Interactions: A user can interact with the video player or the server [12] in several ways such as mouse clicking, mouse moving, fast forward, fast backward, object zoom-in, object zoom-out, add or delete. Some of these interactivity behaviors require dynamic adaptation of the bit-rate of each video object and dynamic resource allocation coordination among multiple video objects. In object-based video multicast applications, different clients can have different views and interactions for the same video.

3. Bitstream re-organization, prioritization and packetization

In this section, we present a new bitstream classification, prioritization and packetization scheme in which different types of data such as shape, motion and texture are re-assembled, assigned to different priority classes, and packetized into different classes of network packets provided by differentiated services. It can be seen that this packetization scheme can improve the error resilience capability and flexibility of bit-rate control.

3.1. Bitstream classification

After compression, encoded video data are placed in the bitstream according to the temporal and spatial position of its content, i.e., frame by frame, macroblock by macroblock, and block by block. Information within the compressed bitstream can be divided into several semantic types such as control header, shape, motion and texture information. These different types of information

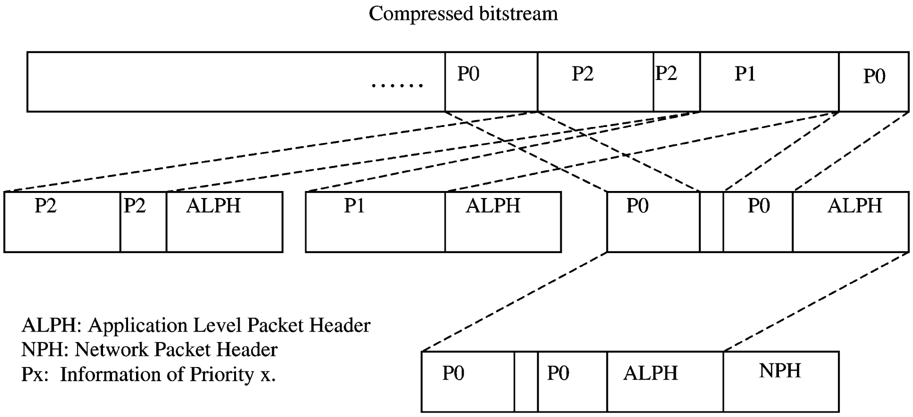

Compressed bitstream
......
P0
P2
P2
P1
P0
P2
P2
ALPH
P1
ALPH
P0
P0
ALPH
ALPH: Application Level Packet Header
NPH: Network Packet Header
Px: Information of Priority x.
P0
P0
ALPH
NPH

Fig. 4. Intelligent packetization.

are interleaved together, although intrinsically they have different importance levels for decoding and transmission. For example, shape and motion information is more important than texture for a P frame in MPEG4. If the shape and motion information is lost during transmission, the decoder cannot reconstruct the P frame successfully. However, if partial texture information is lost without the loss of shape and motion, it is still possible to reconstruct the P frame with acceptable quality. Existing Internet video streaming and networking schemes usually do not consider this kind of differentiation within the bitstream [3,9,15,18,19,24,27], or just distinguish different types of frames (I, P or B) or different layers (base layer and enhancement layers) [8,26]. When network congestion occurs, packets are discarded with no distinction in most previous works on video transmission. Some important information may be dropped together with some less important information and the decoder cannot produce proper video sequences.

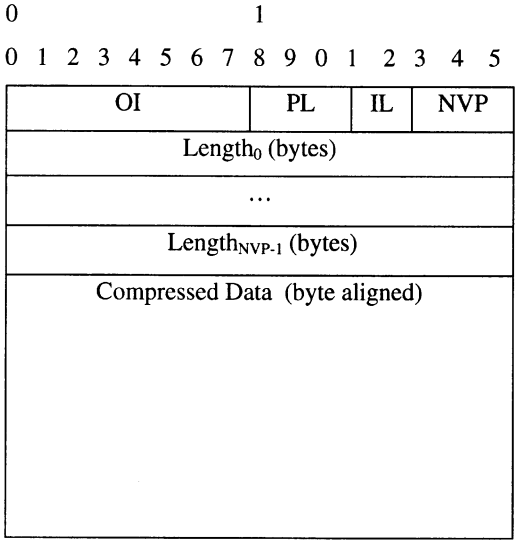
The information in the bistream of object-based video coding such as MPEG4 can be classified into the following categories without considering the enhancement layers:

- Control information, such as Video Object Header, Video Object Layer Header, and Video Object Plane Header.
- Shape information of I frame.
- Texture DC information of I frame.
- Texture AC information of I frame.
- Shape information of P frame.
- Motion information of P frame.
- Texture information of P frame.
- Shape information of B frame.
- Motion information of B frame.
- Texture information of B frame.

3.2. Prioritization and packetization

Different types of information can be assigned to different importance levels. Besides information type, the importance level of the information is decided by the following two factors: the distortion reduction of the information and the dependency relationship between different parts of information. When transmitted, information with the same priority is aggregated and different priorities of information are packetized into different classes of packets (Fig. 4). For example, we classify the single layer or base layer of multi-layer compressed information into the following classes:

- Priority 0 class:
  - ◦ control information;
  - ◦ shape information of I frame (base layer);
  - ◦ texture DC information of I frame (base layer).
- Priority 1 class: texture AC information of I frame (base layer).
- Priority 2 class:
  - ◦ shape information of P frame (base layer);
  - ◦ motion information of P frame (base layer).



OI: Object Identification

PL: Priority Level

IL: 1, multiplexed; 0, otherwise

NVP: Number of VideoPackets in the ALP

Lengh: Length of each compressed data (full or part of the information of a video packet)

Fig. 5. Format of the application level packet.

- Priority 3 class: texture information of P frame (base layer).
- Priority 4 class:
  - ○ shape information of B frame (base layer);
  - ○ motion information of B frame (base layer);
  - ○ texture information of B frame (base layer).

In case network resources cannot satisfy the rate requirement of a video object flow, the packets with lower priorities will be discarded by the sender or intermediate nodes earlier than those with higher priorities if needed. In addition, different error control mechanisms can be implemented on different priority packets to enhance the error resilience capability. To maintain compliance with the MPEG4 syntax, we use an index table for each video object rather than define a new syntax in the video bitstream. The index table includes several items such as index number, information category, priority level, starting position (relative), and length. This table is used to index different types of information in the compressed bitstream and is generated as an individual file along with the compressed bitstream when encoding a video object. So the index table actually is a virtual table which acts only as a reference for extracting different parts of information and does not constitute part of the bitstream.

The data partitioning mode of MPEG-4 video encoding [20] is adopted in our approach. Under data partitioning mode, in each video packet (VP), for I frame shape and texture DC information is separated from texture AC information by DC marker, and for P frame shape and motion information is separated from texture information by motion marker. In MPEG-4 Standard [20], VP is designed mainly for resynchronization purpose to enhance error resilience capability, rather than packetization for transmission. Generally, the size of VP is much smaller than the maximum transmission unit (MTU) of the Internet physical networks. If mapping each VP directly into a network packet, usually the overhead is large and the transmission efficiency is low.

We define a new application level packet (ALP) format for object-based video. The format of application level packet is illustrated in Fig. 5. We limit ALP size to no larger than the MTU of the network. We also avoid using small ALPs in order to achieve high transmission efficiency. As shown in Fig. 5, several parts of the information with the same priority will be multiplexed at the ALP level, though probably they are from different VPs. However, multiplexing will give rise to a new problem. If one packet that contains a lot of consecutive motion information is lost, severe quality degradation may be brought to video playback. Hence, we use two methods to reduce this kind of impairment. One is to limit the number of video packets in an ALP. The other is to place video packets interleavingly. For example, shape and motion information of video packet with 0, 2, 4 are placed into ALP0, and information of video packet 1, 3, 5 are placed into ALP1, respectively. Our rule for multiplexing of the shape and motion information or texture information of VPs into one ALP is that we interleave until one of the following constraints is met: the number of VPs in the current ALP reaches a certain threshold; the size of one ALP reaches a certain maximum size; or a different vop_coding_type is found.

4. Adaptive video transmission

After packetization, video packets are forwarded to the network. Because the aggregated traffic overload of the network fluctuates with time, the video application should accordingly adapt its transmission rate to the available network bandwidth to achieve fairness among different users. We can use the network bandwidth estimation algorithm approaches [26,28] to calculate the available network bandwidth for a video application. The sender can adjust its transmission rate to the estimated network bandwidth by rate control at sender or intermedia node by selectively dropping lower priority packets. The packets chosen to enter the network are mapped to various different Diffserv classes according to their priority and the network resource state. Network can also drop packets to avoid or alleviate network congestion if it occurs.

4.1. Dynamic packet-level rate control at the sender side

Our scheme can support adaptive rate control by discarding some packets with lower priorities at the sender or/and intermediate nodes according to the network available bandwidth. We assume that there are N priority levels, P_i $(0 \leqslant i < N)$, for a video object, and each level has the original bit-rate r_i $(0 \leqslant i < N)$, and the original rate of the video flow is R. Obviously, we have

$$R = \sum_{i=0}^{N-1} r_i. \tag{1}$$

During transmission, if the network congestion occurs or the receivers require lower bit-rates through some user-interactivity behaviors, the bit-rate of the video object needs to be reduced to R'. In order to do that, we first need to find the index k $(0 \leqslant k < N)$ satisfying

$$\sum_{i=0}^{k-1} r_i \leqslant R' < \sum_{i=0}^{k} r_i. \tag{2}$$

All packets with priority P_j $(k < j < N)$ will be discarded at the sender level. Then, if

$$R' = \sum_{i=0}^{k-1} r_i \tag{3}$$

all packets with priority P_k will also be discarded at the sender level. Otherwise, some fine bit-rate adjustment will be implemented, meaning some packets with priority P_k will be selectively discarded at the sender and/or the intermediate nodes. For example, if some B frames need to be dropped, we can employ the scheme proposed in [8] to discard B frames selectively. For another example, if texture information within the P frame needs to be partially discarded, we will adopt the progressive fine granularity scalability (PFGS) [16] and discard $A\%$ texture information in P frame wherein

$$A = \frac{\sum_{i=0}^{k} r_i - R'}{r_k} \times 100\%. \tag{4}$$

It can be seen that our rate control scheme supports both temporal and quality scalability.

4.2. Forwarding mechanism to support Diffserv classes at routers

To assign the prioritized video packets to several network Diffserv levels, the priority of each packet is classified and conditioned. With priority of each video packet, routers can do re-mapping under constraints such as loss-rate differentiation and pricing.

The random early detection (RED) queue management and the weighted fair queuing (WFQ) scheduling provide the differentiated forwarding in our proposed scheme [31]. RED is a congestion avoidance mechanism that takes advantage of TCP congestion control mechanism. By randomly dropping packets prior to periods of high congestion, RED tells the packet source to decrease its transmission rate. Assuming the packet source is using TCP, it will decrease its transmission rate until all the packets reach their destination, indicating that the congestion is cleared. When RED is not configured, output buffers fill during periods of congestion. When the buffers are full, tail drop occurs, and all additional packets are dropped. RED reduces the chances of tail drop by selectively dropping packets when the output interface begins to show signs of congestion. By dropping some packets early rather than waiting until the buffer is full, RED avoids dropping large numbers of packets at once and minimizes the chances of global synchronization. Thus, RED allows the transmission line to be used fully at all times.

The WFQ scheduling is currently implemented in many advanced routers since it guarantees each queue to be allocated a fair share of bandwidth irrespective of the behavior of other queues in the same router. We consider the multiple queue case in which each queue combines with one network class. These queues are served by WFQ scheduler.

4.3. Statistical analysis for packet forwarding at the routers

In this section we give the performance evaluation of forwarding mechanism in which RED and WFQ are combined. We derive the statistical performance parameters such as packet loss and delay as well as the relation among the parameters. These parameters can be used in implementation mechanisms such as dynamic buffer allocation at the router, routing path selection and call admission control. A WFQ scheduler sends out packets just before reaching the maximum delay variation for packets in the higher-priority queues. This decreases delay variation in the lower-priority queues. Our analysis is based on WFQ and the analytical model shown in Fig. 6. Totally, there are N classes, and the overall service rate is μ. Each class (e.g., class i) has four parameters: the arrival rate λ_i, the buffer length B_i, the weight w_i, and the threshold H_i. We aim to calculate the statistical packet loss and delay using probability theory. However, in normal situations only the upper bound of the packet loss and delay is given.

To simplify the analysis, we make the following assumptions: (1) the source is Poisson stream; (2) the service time is negative exponential distribution; (3) the process of the transition of the queue length is a birth–death process; and (4) RED is applied for packet dropping. Notice that we assume every class will be regulated (e.g., through shaper) before it enters the core network so that the Poisson assumption is relatively reasonable. Based on the above assumptions, the model of each class is an M/M/1 queue, whose arrival rate is λ_m and service rate is $w_m\mu$. The state transition diagram is illustrated in Fig. 7.

The state i $(0 \leqslant i \leqslant Bm + 1)$ of class m has probability $\pi_{m,i}$. Using stochastic balance, we can

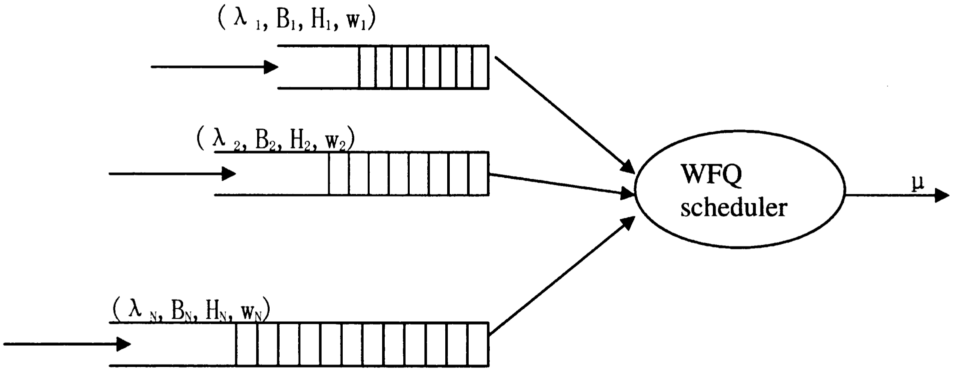


Fig. 6. Analytical model for packet forwarding.

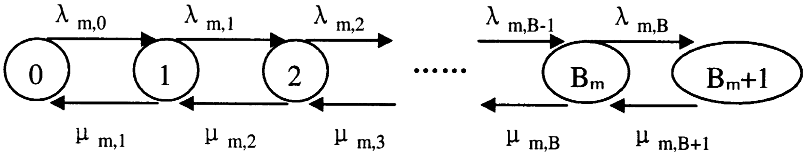


Fig. 7. Transition diagram of the queue state.

get

$$\pi_{m,k} = \pi_{m,0} \prod_{i=1}^{k} \frac{\lambda_{m,i-1}}{\mu_{m,i}}, \tag{5}$$

where

$$\pi_{m,0} = \left[1 + \sum_{k=1}^{B_m+1} \prod_{i=1}^{k} \frac{\lambda_{m,i-1}}{\mu_{m,i}} \right]^{-1}. \tag{6}$$

We perform analysis under the following cases:

(1) We consider the non-work-conserve case without using the RED schedule.

For class i, $\lambda_{m,i} = \lambda_m$ is a constant, and $\mu_{m,i} = w_m \times \mu$ is also a constant.

Substituting $\lambda_{m,i} = \lambda_m$ and $\mu_{m,i} = w_m \times \mu$ into Eqs. (5) and (6), we obtain

$$\pi_{m,k} = \pi_{m,0} \left(\frac{\lambda_m}{w_m \times \mu} \right)^k, \tag{7}$$

where

$$\pi_{m,0} = \left[1 + \sum_{k=1}^{B_m+1} \left(\frac{\lambda_m}{w_m \times \mu} \right)^k \right]^{-1} = \frac{1 - (\lambda_m/(w_m \times \mu))}{1 - (\lambda_m/(w_m \times \mu))^{B_m+2}}. \tag{8}$$

For M/M/1 model, the probability of the queue length that one observes at any time is equal to that someone who just joins the queue observes. That is, $P - \ = P$. Hence, it can be seen that the packet loss probability is the probability that the queue buffer is full, which is given by

$$P_{m,\text{loss}} = \pi_{m,B_m+1} = \left(\frac{\lambda_m}{w_m \times \mu} \right)^{B_m+1} \frac{1 - (\lambda_m/(w_m \times \mu))}{1 - (\lambda_m/(w_m \times \mu))^{B_m+2}}. \tag{9}$$

Meanwhile, the delay probability is calculated as follows:

$$P_{m,\text{delay}} = \sum_{i=0}^{B_m} \pi_{m,i} \times \mathrm{e}^{-\lambda i} \lambda \frac{(\lambda i)^{m-1}}{(m-1)!}. \tag{10}$$

(2) Now we take RED into account. Note that the arrival rate is not a constant any more in this case. When the length of queue m reaches H_m, the arriving packets will be dropped with a probability D_m, which is determined by the following RED equations:

$$D_{m,k} = \begin{cases} 0 & \text{when } 0 \leqslant £e < £e_{\min}, \\ \dfrac{k - k_{\min}}{k_{\max} - k_{\min}} D_{m,\max}£e & \text{when } k_{\min} \leqslant k \leqslant k_{\max}, \\ 1 & \text{when } k_{\max} < k. \end{cases} \tag{11}$$

When the arrival rate decreases to $D_m \times \lambda_m$, $\pi_{m,k}$ changes to

$$\pi_{m,k} = \pi_{m,0} \prod_{i=1}^{k} \frac{\lambda_m \times (1 - D_{m,k})}{w_m \times \mu}. \tag{12}$$

From this equation, we can compute $\pi_{m,k}$ using $\sum_{k=0}^{B_m+1} \pi_{m,k} = 1$. Thus, the loss probability is equal to π_{m,B_m+1}. Substituting Eq. (12) into Eq. (10), we get the updated delay probability.

(3) Finally, we consider the work-conserve case with RED. That is, when queue i is empty, the weight of other queues is changed by the following equation:

$$w'_j = \frac{w_j}{\sum_{k \neq i} w_k}. \tag{13}$$

In this case, the service rate is not simply $w_m \times \mu$. It is confined by the queue state. In such a situation, it is rather difficult to directly calculate the probability of the state. Here, we give an iterate equation to solve this problem as follows:

$$w'_j = w_j \times \prod_{i \neq j} (1 - \pi_{i,0}) + \sum_{A \subset U} \left(\frac{w_j}{\sum_{k \in A} w_k + w_j} \prod_{i \in A} (1 - \pi_{i,0}) \prod_{i \in U - A} \pi_{i,0} \right), \tag{14}$$

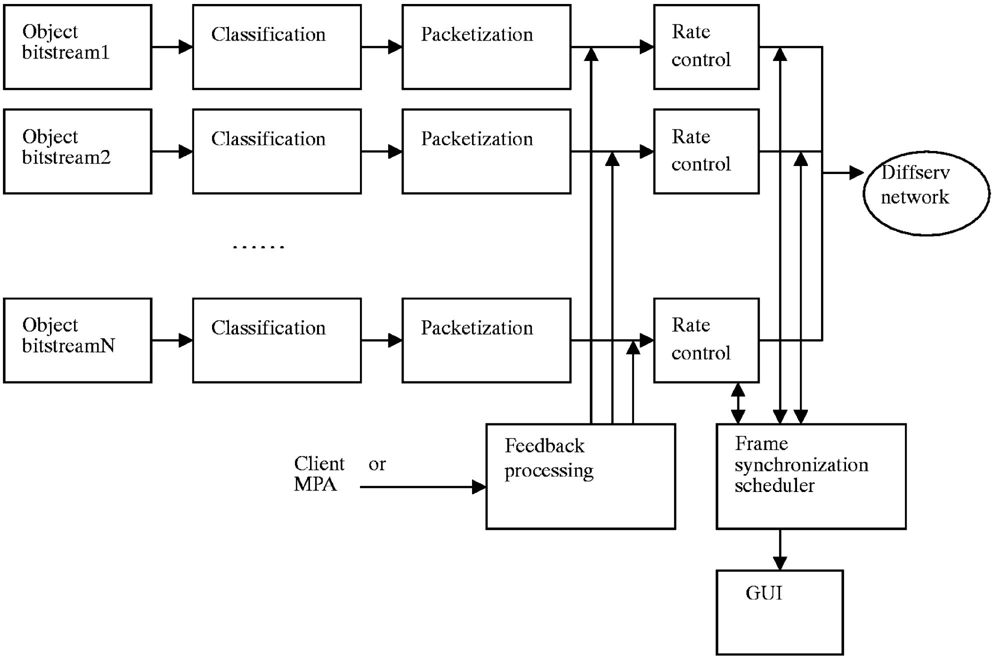


Fig. 8. System diagram of the video server.

where $U = \{1,2,\dots,j-1,j+1,\dots,N\}$ and A represents all the subsets of U.

From Eq. (14), we can get the service rate $\mu_m = w'_m \times \mu$ and then get $\pi_{m,k}$.

5. Experimental results and performance evaluation

We implemented a simple MPEG4 video streaming system and run it on a simulated Diffserv network to test our proposed user-aware object-based video transmission scheme. Our streaming system is based on the Microsoft MPEG4 video encoding/decoding source codes and Microsoft research's IPv6 protocol stack source codes. The system diagrams of the server and the receiver are depicted in Figs. 8 and 9. Besides our scalable transmission approach, for the sake of comparison we also implemented the traditional approach in which the bitstream is packetized with no information re-organization/prioritization and all packets have a fixed size (600 bytes). In addition, we implemented the differentiation functionality within one flow based on the Microsoft IPv6 communication protocol stack.

We used the chain network configuration in Fig. 10 to test our approach. In the chain configuration, G1 consists of one MPEG-4 source, three TCP connections and three UDP connections, while G2, G3 and G4 all consist of three TCP connections and three UDP connections, respectively. The link capacity between the routers is 1 Mbps on Links 12, 23 and 34. Simulation parameters are illustrated in Table 1. Under different network conditions, we compared our scalable transmission scheme with the traditional approach without distinguishing different kinds of information under the different network conditions (Table 2) simulated in [17].

Experiment 1. This experiment aims to investigate the quality improvement for object-based video transmission using our new scheme. Two standard MPEG4 video testing sequences with CIF format (900 frames), a typical head-and-shoulder video sequence, "Akiyo", and an active sequence,

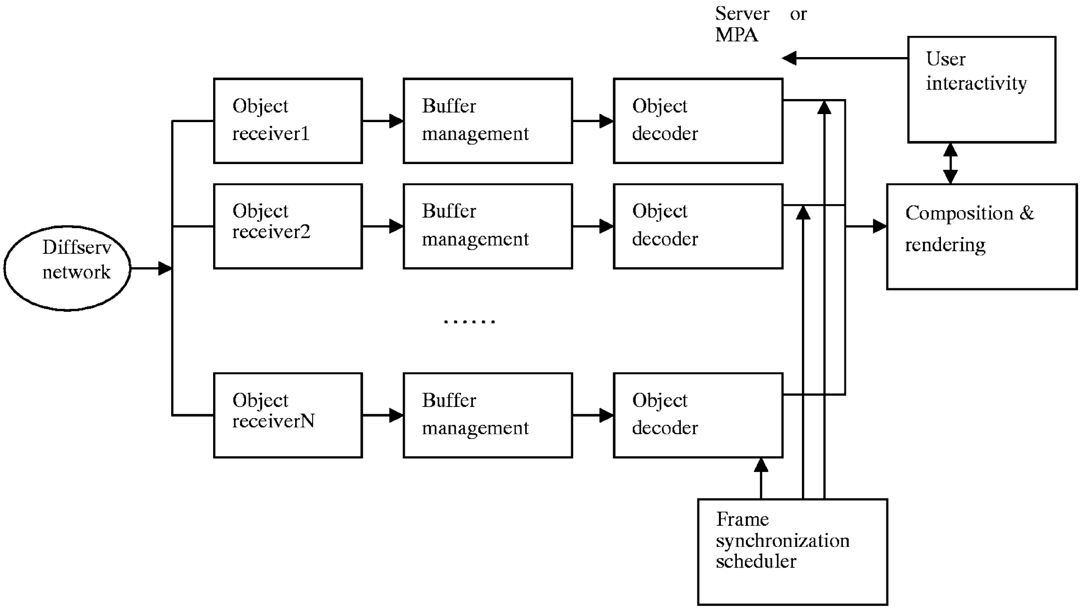


Fig. 9. System diagram of the receiver.

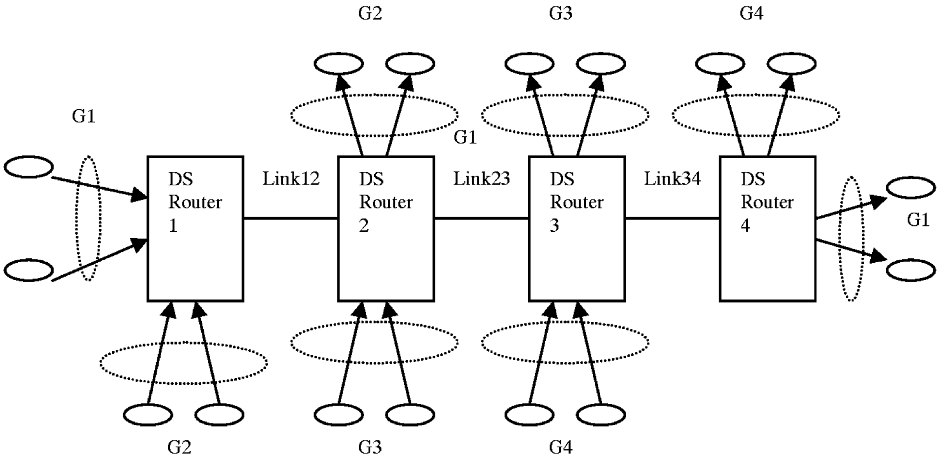


Fig. 10. A chain network model.

"Bream", were used in this experiment. We performed experiments under both low and high bit-rates using Akiyo and Bream sequences at 30 fps. In the encoded bitstreams, there is an I frame for every 99 P frames (B frames are not included in the bitstream). Figs. 11–18 show the PSNR curves of Y parameter under various conditions. From these figures, it can be seen that our proposed approach is much better than the traditional one, particularly for high bit-rate video. Figs. 19 and 20 show examples of the reconstructed video frame using our approach and the traditional one.

Experiment 2. This experiment is to test dynamic bit-rate coordination among video objects

Table 1
Network simulation settings

End system	TCP	Mean packet processing delay	300 μs
		Packet processing delay variation	10 μs
		Packet size	1000 bytes
		Maximum receiver window size	150 kbytes
		Default timeout	500 ms
		Timer granularity	500 ms
		TCP version	Reno
	UDP	*E*(ton)	100 ms
		E(toff)	150 ms
		rp	200 kbps
		Packet size	1000 bytes
Diffserv router		Buffer size	10 kbps
		Packet processing delay	4 μs
Link	End system to router	Link speed	10 mbps
		Distance	1 km
	Router to router	Distance	100 km

Table 2
Packet loss rate parameters

	Class 0	Class 1	Class 2	Class 3
Case 1	0.2%	0.5%	1%	5%
Case 2	0.5%	1%	4%	25%

according to user's interactive behaviors. We used the standard "News" video sequence with CIF format. But the video object composed of the two anchors was split into two objects. Experimental results of dynamic bit-rate coordination among multiple video objects caused by remote users' dynamic interactions are illustrated in Figs. 21–23. It can be seen from them that objects' qualities can be changed dynamically according to user's behaviors. Fig. 21 shows that due to user's preference, the object of dancers was blurred because of decrease of the bandwidth allocated to it, but the quality of the male anchor and the female anchor did not change. Fig. 22 demonstrates that due to the serious deficiency of the bandwidth, the object of dancers was deleted by the user and the object of the male anchor was blurred in order to guarantee that enough bandwidth was allocated to the object of the female anchor based on user's preference. Fig. 23 shows that video qualities of the male

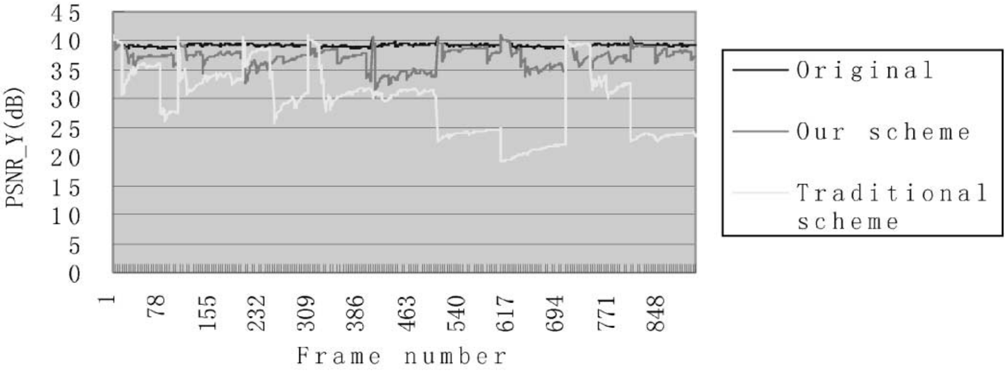


Fig. 11. Video quality comparisons for Akiyo at 3.9% packet loss rate (actual bit-rate = 254 kbps; original bit-rate = 260 kbps).

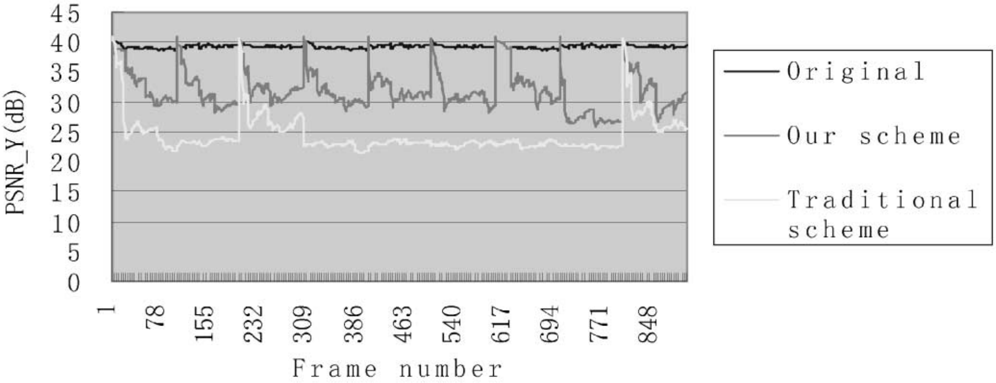


Fig. 12. Video quality comparisons for Akiyo at 19.2% packet loss rate (actual bit-rate = 213 kbps; original bit-rate = 260 kbps).

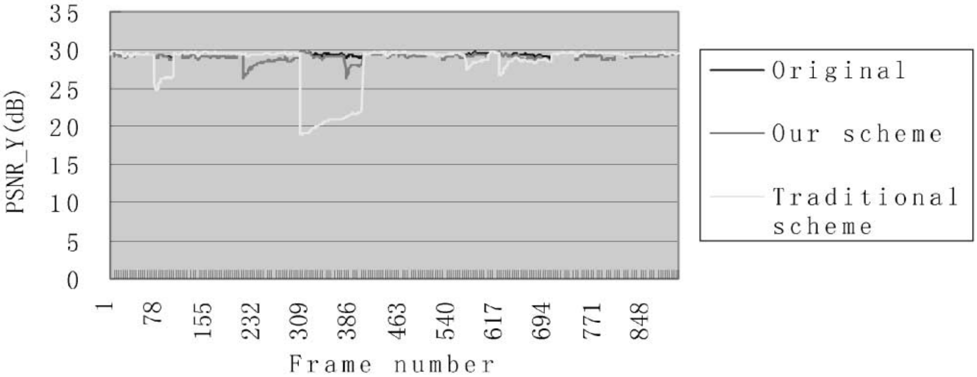


Fig. 13. Video quality comparisons for Akiyo at 1.4% packet loss rate (actual bit-rate = 56 kbps; original bit-rate = 57 kbps).

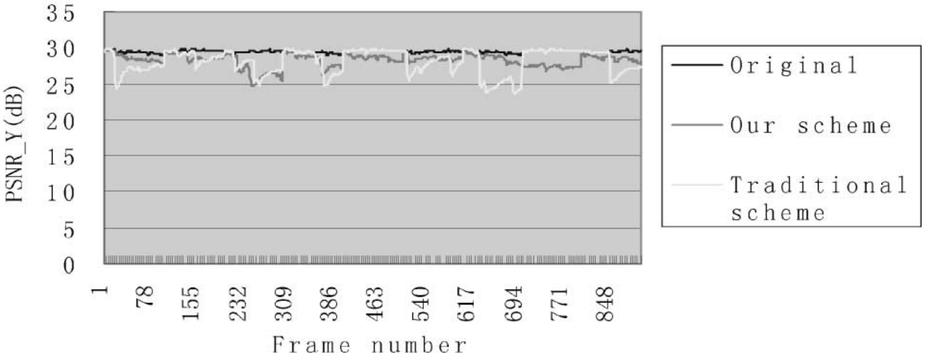


Fig. 14. Video quality comparisons for Akiyo at 6.3% packet loss rate (actual bit-rate = 54 kbps; original bit-rate = 57 kbps).

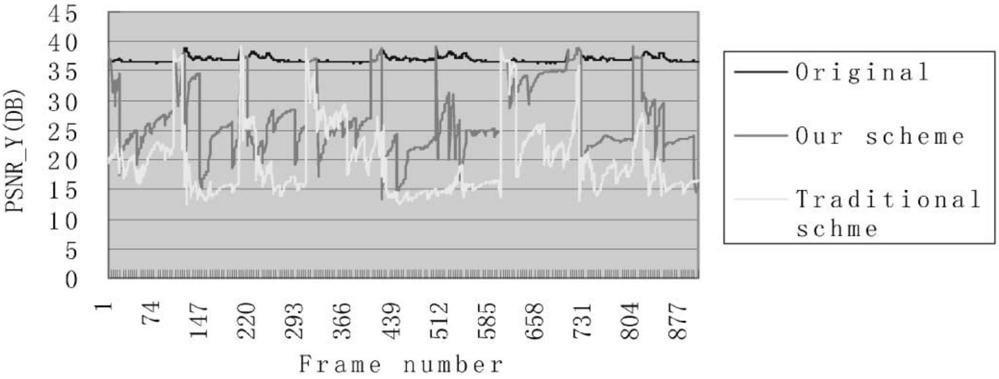


Fig. 15. Video quality comparisons for Bream at 4.4% packet loss rate (actual bit-rate = 987 kbps; original bit-rate = 1030 kbps).

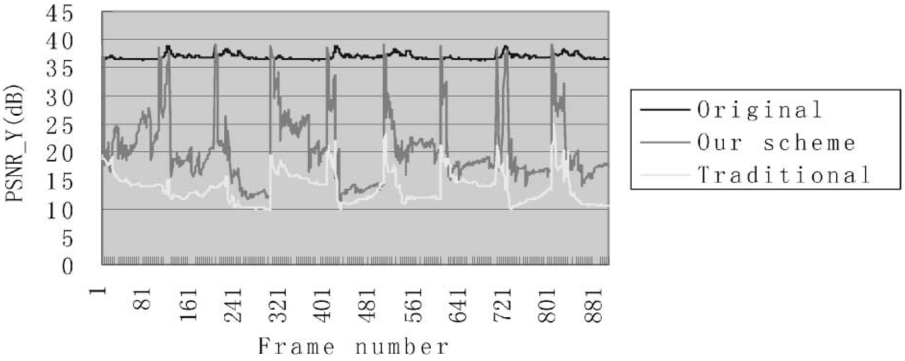


Fig. 16. Video quality comparisons for Bream at 22.0% packet loss rate (actual bit-rate = 796 kbps; original bit-rate = 1030 kbps).

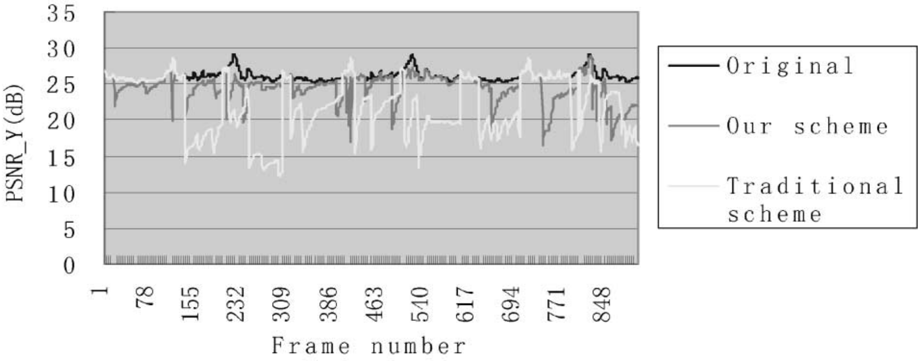


Fig. 17. Video quality comparisons for Bream at 2.5% packet loss rate (actual bit-rate = 182 kbps; original bit-rate = 187 kbps).

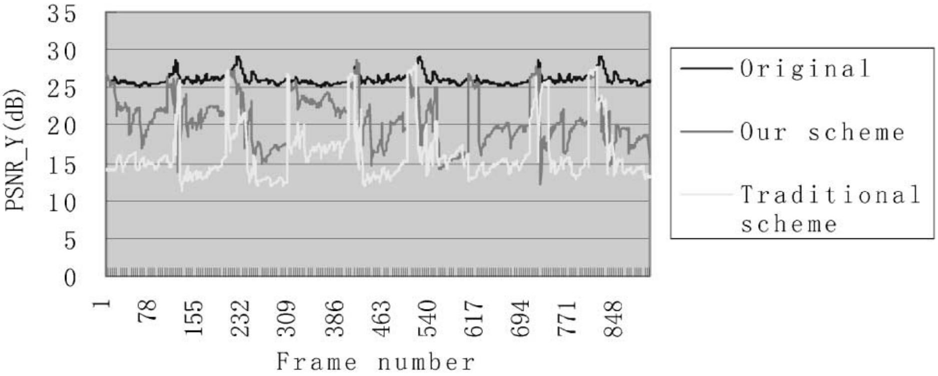


Fig. 18. Video quality comparisons for Bream at 11.7% packet loss rate (actual bit-rate = 168 kbps; original bit-rate = 187 kbps).

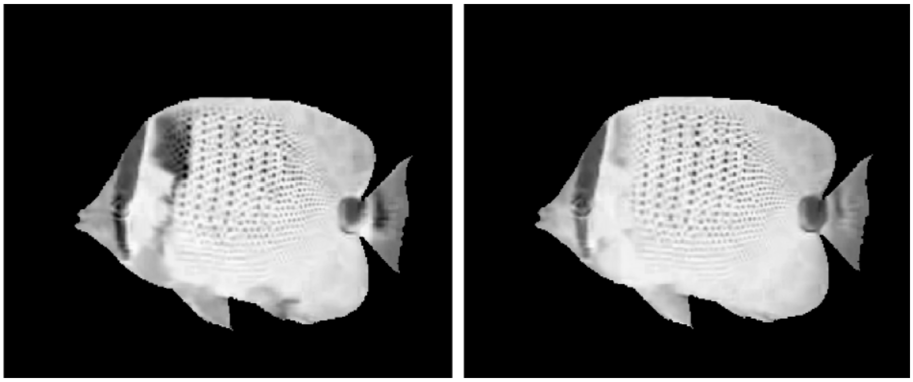


Fig. 19. Reconstructed video frame (Number: 150) for Bream at 2.5% packet loss rate (actual bit-rate = 183 kbps; original bit-rate = 187 kbps). Left: standard approach, PSNR = 16.5 dB. Right: the proposed scheme, PSNR = 23.7 dB.

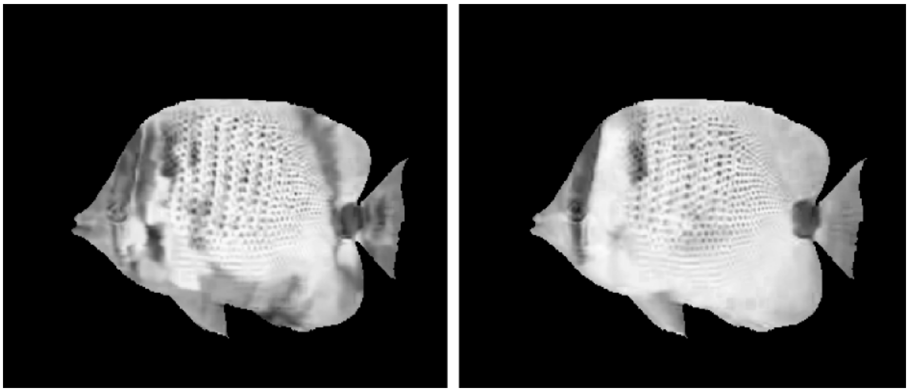


Fig. 20. Reconstructed video frame (Number: 150) for Bream at 11.7% packet loss rate (actual bit-rate = 168 kbps; original bit-rate = 187 kbps). Left: standard approach, PSNR = 13.8 dB. Right: the proposed scheme, PSNR = 19.3 dB.

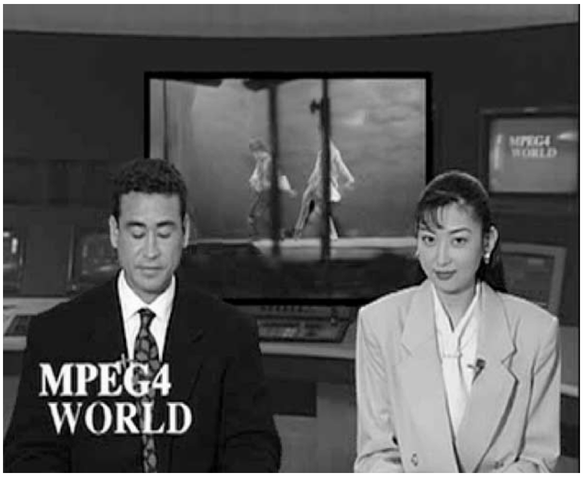


Fig. 21. The reconstructed video frame with object quality control. Frame Number: 45. Note that the object of *Dancers* has been blurred due to the limited bandwidth.

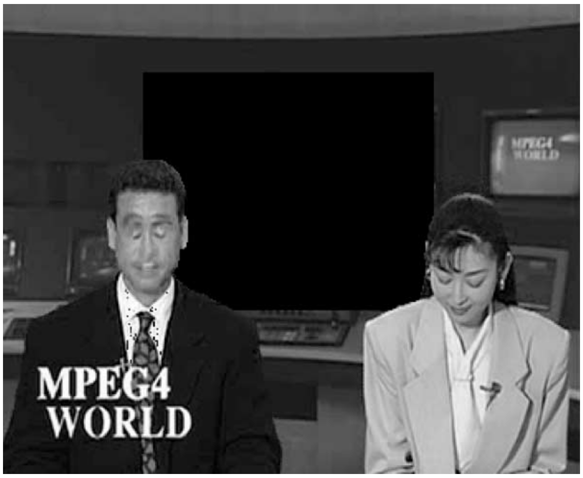


Fig. 22. The reconstructed video frame with object quality control. Frame Number: 140. Note that the object of *Dancers* has been deleted and the object of the male anchor has been blurred because of serious deficiency of network bandwidth and the user's preference.

anchor and the female anchor were switched with each other based on the change of the user's preferences. Fig. 24 illustrates the bitate traces of the objects during one streaming instance.

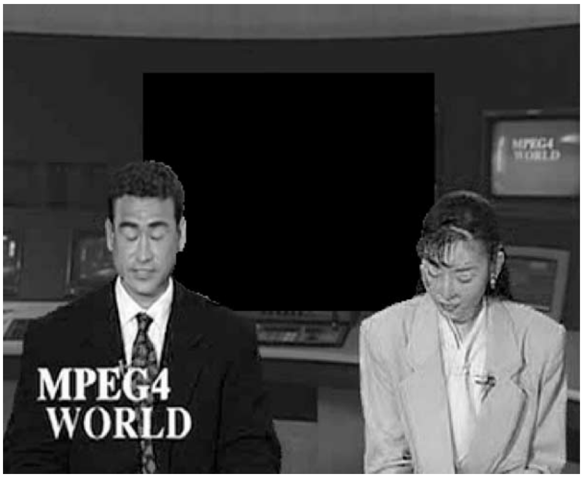


Fig. 23. The reconstructed video frame. Frame Number: 280. Note that because the user's interest in switching quality from the object of female anchor to the male anchor, the bandwidth allocated to the two anchors is also switched. This frame shows that the video quality of the female anchor was decayed but the video quality of the male was increased according to user's selection.

6. Extension to multicast

Since multicast can greatly save network bandwidth, it is considered as an effective communication support for multi-party multimedia applications such as distance learning and video broadcasting. However, due to the heterogeneity of the network and user's capabilities, a single-sender transmission rate cannot satisfy different bandwidth requirements of different receivers. In general, the sender rate is adapted to the requirement of the worst-positioned receiver. The disadvantage of this approach is that the quality of the received video is degraded, except for the worst-positioned one. This limitation can be overcome by using layered multicast mechanisms that have received a lot of attention recently [14,15,18,19,23]. McCanne et al. [19] described a receiver-driven layered multicast protocol for rate-adaptive video transmission. In their work, the source transmits each layer of its signal on a separate multicast group. Each receiver specifies its level of subscription by joining a subset of multicast groups of layered video. Li et al. [14] also proposed and evaluated a layered video multicast with

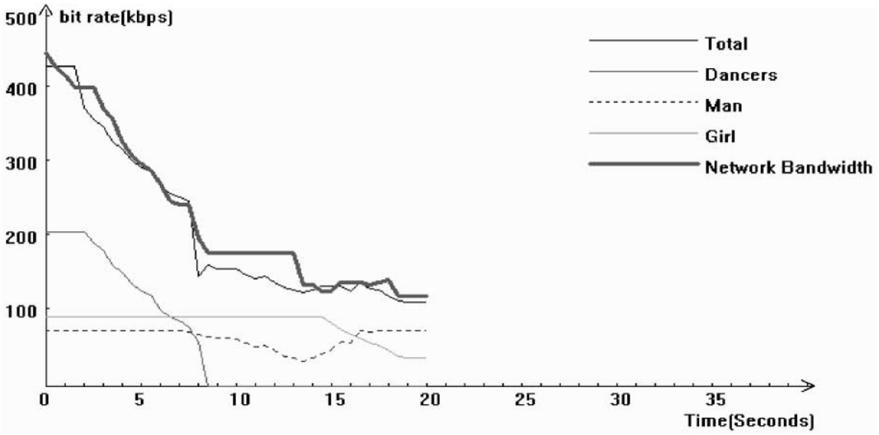


Fig. 24. The transmission rates for objects within the "News" sequence during a streaming process.

retransmission (LVMR) scheme for distributing video using layered coding over the Internet. Conceptually, they improved the quality of reception within each video layer by retransmitting lost packets that are given an upper bound on recovery time. In addition, they apply an adaptive playback scheme to help achieve more successful retransmission. They also use hierarchical rate control (HEC) mechanisms to adapt rate to network congestion and heterogeneity. However, the layered approach usually requires multiple network sessions for one video stream. It is complicated for the network and the end-system to control and manage many network sessions for an MPEG4 video program consisting of multiple video objects, and the synchronization among multiple layers belonging to the same video object is difficult to maintain. Moreover, the transmission rate cannot be adjusted in finer granularity than the difference between layers. Therefore, the layered multicast mechanism may not be suitable for object-based video consisting of multiple video objects. Our intelligent packetization and bit-rate control approach described above can be easily extended to the multicast case. By taking advantage of MPAs, it provides each video object with only one single network session, but different users can obtain the reconstructed video object with different perceptual qualities according to their receiving capabilities and interactivity behaviors.

Fig. 25 gives an example to illustrate our multicast scheme. It can be seen from Fig. 25 that the control (signaling) paths are required by clients, MPAs and the server, so are data paths. Control paths are used to set up multicast sessions, and negotiate or renegotiate what and how much information within each object needs to be transmitted. The control paths also coordinate bit-rate allocation among multiple video objects according to network state and user interactions. It should be noted that Fig. 25 only presents the logical concepts of the data path and control path, not the physical connections. Among the server, MPAs and clients, there are probably one or more routers or switches. In Fig. 25, we assume client 1 access to the network through 10 M Ethernet, client 2 through 2 M ISDN, client 3 through 56k modem, and client 4 through 34k modem. We further assume that the server is multicasting object-based MPEG4 video to the four clients. The server packetizes the bitstream of each video object and sends packets with different QoS requirements. When congestion occurs, routers discard packets with lower priorities. The MPAs filter the video information received from the upstream, and selectively send packets according to the capabilities of the network links and clients.

A new resource allocation policy is proposed for our multicast scheme. In a multicast session, each MPA/user is associated with a two-element set

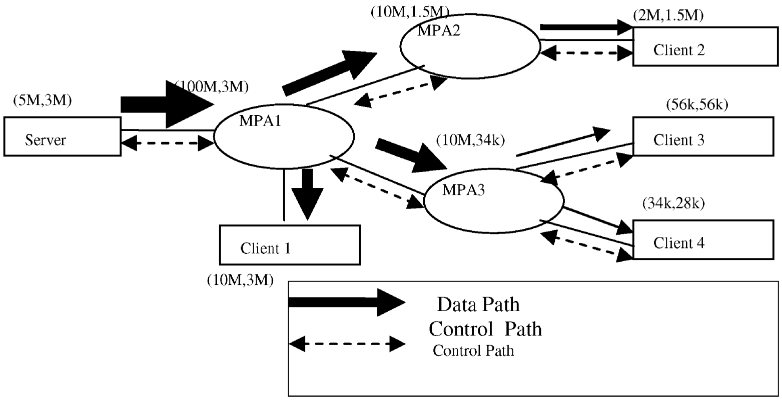


Fig. 25. Data path and control path in the network (logical path).

(Capacity, Requirement). The *Capacity* represents the amount of resource available for a video object in the MPA or the end system of the user. The *Requirement* represents the amount of resource for a video object required by the MPA or the user. Considering the network bandwidth, the *Capacity* of an MPA is determined by the link state between the MPA and its neighboring upstream MPA or the sender if it has no upstream MPA. We assume that the *Capacity* of the sender is the bit-rate of the original bitstream, denoted as *RATEs*. However, the *Requirement* of an MPA is determined by *(Capacity, Requirement)* of all its neighboring downstream MPAs and the users connecting directly to this MPA within the multicast session. For example, we assume that an MPA, named V_i, has M neighboring downstream MPAs and users. The two element sets of MPA are (Cap$_{ij,}$ Req$_{ij}$) ($0 \leqslant j < M$), then the *Requirement* of V_i is calculated as follows:

$$\mathrm{Req}_i = \min\{\min(\mathrm{Cap}_{ij})\},\quad \max(\mathrm{Req}_{ij}), \mathrm{RATEs}\},$$
$$0 \leqslant j < M.$$

For a receiver, the *Requirement* is determined by the user's preference and interactivity behavior. In a multicast session, the *(Capacity, Requirement)*s of its MPAs and users can be calculated from bottom to top in the multicast tree. At the ith MPA$_i$, the rate of the video object bitstream can be adapted to Req$_i$ by using the scalable transmission scheme proposed in Section 4. It can be seen that the MPAs have the functionality of filtering. However, traditional video filters need to implement decoding and re-encoding to complete the filtering. The MPAs can perform filtering by only discarding some less important packets and is much simpler and faster than the traditional filters. The *(Capacity, Requirement)*s of a single video object are depicted in Fig. 25. It can be seen that all clients can obtain their bit-rates except Client 3.

7. Conclusion and discussions

In this paper, a novel object-based quality-adaptive video multicast approach is proposed. Our approach can provide both object-based interactive functionalities and flexible quality adaptation for heterogenous users. The main contributions of this paper are as follows.

1. A new general transport framework for complex multimedia applications over the next generation network. Three new features are described in this framework: differentiation functionality within one IP session as well as among different IP sessions; application-aware intelligent resource control and management both at the end system and the edge of the network; multimedia processing agent (MPA) on the bottlenecks within domains and at the edge between domains. Based on these new

features and the new capabilities of the next generation Internet, multimedia transmission with QoS provision are achieved effectively.
2. A new bitstream classification, prioritization and packetization scheme in which different types of data such as shape, motion and texture are re-assembled, assigned to different priority classes, and packetized separately. This scheme improves the capability of both error resilience and flexible transmission rate control.
3. A simple but effective object-based dynamic rate control and adaptation mechanism by selectively dropping packets in conjunction with differentiated services to minimize the end-to-end video quality distortion.
4. A new adaptive multicast resource allocation policy for heterogeneous users. Our proposed approach provides each video object with only one single network session but has the advantage of the layered multicast approach.

We implemented a real user-aware object-based video transmission system and ran it on a simulated Diffserv network. Experimental results show that our proposed transmission approach can provide improved QoS performance and flexible user interactivities under the same network resource status and user requirements.

Scalable coding is capable of gracefully coping with the bandwidth fluctuations in the Internet. Extending this framework to scalable video codec such as FGS [13] or PFGS [16] is one of the future research efforts. Another interesting future work is to add unequal error (loss) protection to our scheme for error control [30]. Moreover, multicast to heterogeneous users within the Diffserv networks, particularly signaling mechanism and packet multicast forwarding, need to be further explored.

Acknowledgements

The authors would like to thank Jiangchuan Liu and Bo Li from Hong Kong University of Science and Technology for their partial contribution to packetization part. The authors also thank Daming Hang from Tsinghua University for his contribution to Section 4.3.

References

[1] Y. Benet, The complementary roles of RSVP and Differentiated serveices in the full-service QoS network, IEEE Commun. Mag. 38 (2) (February 2000) 154–162.

[2] Y. Benet et al., A framework for differentiated services, November, Internet Draft, 1998, draft-ietf-Diffserv-framework-01.txt.

[3] C. Bormann, L. Cline et al., RTP payload format for the 1998 version of ITU-T Rec. H.263 video (H.263+), Internet Engineering Task Force, RFC 2429, October 1998.

[4] M. Carlson, W. Weiss, S. Blake, Z. Wang, D. Black, E. Davies, An architecture for differentiated services, RFC 2475, December.

[5] B. Carpenter, D. Kandlur, Diversifying Internet delivery, IEEE Spectrum, November 1999.

[6] D. Clark, W. Fang, Explicit allocation of best effort packet delivery service, IEEE/ACM Trans. Networking 6 (August 1998) 362–373.

[7] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, Assured forwarding PHB group, RFC 2597, June 1999.

[8] M. Hemy, U. Hengartner, P. Steenkiste, T. Gross, MPEG system streams in best-effort networks, in: Packet Video'99, New York.

[9] D. Hoffman, G. Fernando, V. Goyal, RTP payload format for MPEG1/MPEG2 video, Internet Engineering Task Force, RFC 2429, October 1996.

[10] D. Hoffman, M. Speer, Hierarchical video distribution over internet-style networks, in: ICIP'96, Lausanne, Swizerland, September 1996.

[11] V. Jacobson, K. Nichols, K. Poduri, An expedited forwarding PHB, RFC 2598, June 1999.

[12] H. Kalva, L. Tang, J.-F. Huard et al., Implementing multiplexing, streaming, and server interaction for MPEG-4, IEEE Trans Circuits Systems Video Technol. 9 (8) (December 1999) 1299–1311.

[13] W. Li. Streaming video profile in MPEG-4, IEEE Trans. Circuits Systems Video Technol., March 2001.

[14] X. Li, M.H. Ammar, S. Paul, Layered video multicast with retransmission (LVMR): Evaluation of error recovery, in: Proceedings of NOSSDAV, May 1997.

[15] X. Li, M.H. Ammar, S. Paul, Video multicasting over the Internet, IEEE Network Mag. 13 (2) (March/April 1999) 46–60.

[16] S. Li, F. Wu, Y. Zhang, Study of a new approach to improve FGS video coding efficiency, ISO/IEC JTC1/SC29/WG11/m5583, December 1999, Maui.

[17] J. Liu, Congestion Control and feedback algorithms for MPEG4 video, Technical Report, Microsoft Research, China.

[18] S.R. McCanne, Scalable compression and transmission of Internet multicast video, Ph.D Thesis, The University of California, Berkeley, CA, December 1996.

[19] S. McCanne, Receiver-driven layered multicast, in: SIGCOMM Symposium on Communications Architectures and Protocols, Palo Alto, California, August 1996.

[20] MPEG-4 video verification model version 13.0. ISO/IEC JTC1/SC29/WG11/N2687, March 1999.

[21] S. Shenker, R. Braden, D. Clark, Integrated services in the internet architecture: an overview, June 1994, Internet RFC 1633.

[22] J. Shin, J.W. Kim, C.-C. Jay Kuo, Content-based packet video forwarding mechanism in differentiated service networks, in: 10th International Workshop for Packet Video, Sadinia, Italy, May 2000.

[23] D. Sisalem, QoS control using adaptive layered data transmission, in: IEEE International Conference on Multimedia Computing and Systems, 1998.

[24] T. Turletti, C. Huitema, Video conferencing on the Internet, IEEE/ACM Trans. on Networking 4 (3) (June 1996) 340–351.

[25] T. Turletti, C. Huitema, RTP payload format for H.261 video streams, Internet Engineering Task Force, RFC 2032, October 1996.

[26] D. Wu, Y.T. Hou, Y.-Q. Zhang, Transporting real-time video over the Internet: challenges and approaches, in: Proceedings of the IEEE, Vol. 88, No.12, December 2000.

[27] D. Wu, Y.T. Hou, W. Zhu, H.J. Lee, T. Chiang, Y.-Q. Zhang, On end-to-end transport architecture for MPEG-4 video streaming over the Internet, IEEE Trans. Circuits Systems Video Technol., 2000.

[28] Q. Zhang, Y. Zhang, W. Zhu, Resource allocation for audio and video streaming over the Internet, in: IEEE International Symposium on Circuits and Systems (ISCAS) 2000, Geneva, Switzerland, May 2000.

[29] C. Zhu, RTP payload for H.263 video streams, Internet Engineering Task Force, RFC 2190, September 1997.

[30] W. Zhu, Q. Zhang, Y. Zhang, Network-adaptive rate control with unequal loss protection for scalable video over Internet, in: Internat. Symposium on Circuits and Systems, Sydney, 6–9 May 2001.

[31] http://www.cisco.com/warp/public/732.