Joint Rate Allocation and Buffer Management for Robust Transmission of VBR Video

ZHANG Yuan-Hai[1, 2] LI Kai-Hui[1, 2] XU Chang-Qiao[1, 2] SUN Li-Min[1]

Abstract In this paper we present an adaptive video transmission framework that integrates rate allocation and buffer control at the source with the playback adjustment mechanism at the receiver. A transmission rate is determined by a rate allocation algorithm which uses the program clock reference (PCR) embedded in the video streams to regulate the transmission rate in a refined way. The server side also maintains multiple buffers for packets of different importance levels to trade off random loss for controlled loss according to the source buffer size, the visual impact, and the playback deadline. An over-boundary playback adjustment mechanism based on proportional-integra (PI) controller is adopted at the receiver to maximize the visual quality of the displayed video according to the overall loss and the receiver buffer occupancy. The performance of our proposed framework is evaluated in terms of peak signal-to-noise ratio (PSNR) in the simulations, and the simulation results demonstrate the improvement of the average PSNR values as well as the better quality of the decoded frames.

Key words Variable-bit-rate (VBR), video streaming, program clock reference (PCR), buffer management, proportional-integra controller

In order to obtain better visual quality, videos are required to use variable-bit-rate (VBR) encoding. However, it is more difficult to manage the VBR video traffic because of its significant bit-rate burstiness[1]. Normally, transmission of video requires high bandwidth and low delay. Many researches have been done on VBR compressed video transmission[2−9]. In [2], the problem of streaming packetized media in a rate-distortion optimized way was addressed. An interactive descent algorithm was used to minimize the average end-to-end distortion. However, the high computational complexity of this approach made it less appealing during real-time streaming, where the server must adapt to bandwidth variations very quickly. Adaptive media playout (AMP) was proposed from the receiver point of view in [3] to vary the playout rate of media frames according to the buffer occupancy as soon as the target buffer level is reached, which may cause jitter at the critical point of two adjacent buffer levels. A multi-buffer scheduling scheme was proposed in [4] to schedule the transmission based on the source buffer priority. A proportional-integral-derivative (PID) controller was adopted in [5] to have better tradeoff between spatial and temporal qualities. The above two schemes belong to server-side technologies, which only consider the sender buffer state without taking into account the end-to-end delay constraint of multimedia applications. [6] addressed the problem of optimizing the playback delay experienced by a population of heterogeneous clients and proposed a server-based scheduling strategy that targets a fair distribution of the playback delays. [7] modeled the streaming system as a queuing system. An optimal substream was selected based on the decoding failure probability of the frame and the effective network bandwidth. [8] proposed a reverse frame selection (RFS) scheme based on dynamic programming to solve the problem of video streaming over VBR channels. [9] presented a streaming framework centered around the concept of priority drop. It combined the scalable compression and adaptive streaming to provide a graceful degradation of the quality.

Most of the previous approaches focused on regulating transmission rate through the observation of network status, client capabilities, playback rate, and so on, but they ignored the effect of VBR stream structure on source rate control, which may make the sending rate too large and cause the receiver buffer overflow under the situation of high bandwidth. In order to adapt to the timing constraints of streaming media, source rate should be adjusted according to the decoding rate[10]. In [11], two transmission schemes, called PCR-assist constant bit rate (CBR) (PCBR) and PCR-assist dual-rate CBR (PDCBR), were given, which used PCR to control the transmission rate. They reduced the client buffer requirement at the cost of higher transmission rate because of the use of coarse regulation time scale[12].

Besides the problem of transmission rate allocation, the buffer control mechanism is also very important. The VBR encoded video consists of packets with different levels of importance and the packets have different impacts on the presentation quality of the decoded videos. Treating all of the packets with equal importance usually results in severe quality degradation during packet losses in heavy congestion. However, by using intelligent transmission buffer management, random loss can be traded off for controlled loss, which may significantly improve the quality of the received video. In addition, with the use of receiver buffer, playback adjustment can be achieved to smooth the received video stream and reduce the jitter introduced by the changing network delays and the variable transmission rates. The retransmission of the lost frame packets before the decoding deadline is also available.

In this paper, we propose a refined PCR level scheduling strategy that dynamically changes the transmission rate using more detailed time scale. The buffer management mechanism is introduced for both server and client sides to smooth the variation of bit streams further. The integrated scheme takes into account the network status, the client capabilities, and the video stream characteristics to optimize an average quality of service for all the clients. Compared with traditional solutions, our approach is unique in that it provides a more flexible framework to allow a joint decision of the sender and receiver rates to meet the QoS requirements of multimedia applications.

The paper is organized as follows. Section 1 describes the components of our video transmission framework. In Section 2, we present simulation results. Conclusions are given in Section 3.

Received June 22, 2007; in revised form September 17, 2007
Supported by National Natural Science Foundation of China (60434030, 60673178, and 60472076) and National Basic Research Program of China (973 Program) (2007CB307106)
1. Institute of Software, Chinese Academy of Sciences, Beijing 100080, P. R. China 2. Gruaduate University of Chinese Academy of Sciences, Beijing 100039, P. R. China
DOI: 10.3724/SP.J.1004.2008.00337

1 Video transmission framework

Fig. 1 shows the block diagram of the video transmission framework. The rate allocation is performed by a PCR level rate controller (PLRC) to regulate the transmission rate according to the timing information calculated by PCRs. Multiple buffers for different importance levels are applied at the source buffer controller (SBC), via which the scheduling scheme differentiates packets with different priorities, and thus, the selective dropping scheme can be used to minimize the visual impact because of dropping in heavy congestion. At the receiver side, a playback buffer controller (PBC) smoothes the flow and reduces the jitter. Also, playback rate is adjusted appropriately to improve the quality of the displayed video. Individual components of the framework are described in the remainder of this section.

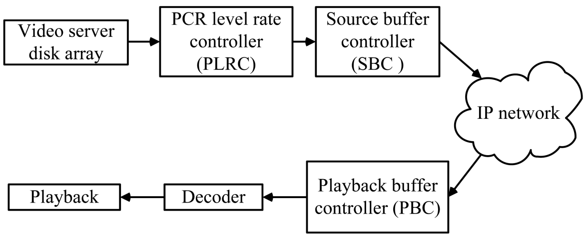


Fig. 1 The block diagram of video transmission framework

1.1 PCR level rate controller

PCR is the timing information embedded in the video stream by the encoder to keep clock synchronization. The traditional PCR-based schemes[11] use the PCR values to make sure that the PCR-containing packets are sent out at the correct time. For the sake of simplicity, they assume that there is only one PCR in one frame interval. However, it is often not the case for VBR video streams, which leads to unprecise sending time and bad quality of playback. In [12], we proposed a frame level scheme which considers the case of several PCRs in one frame interval and developed an analytical model. The model demonstrates that the more accurate the decided rate is, the smaller the required client buffer is, which can lighten the burden on the clients and improve the playback quality.

The above schemes only change the transmission rate when they observe a PCR or when the value of the time counter reaches the value of the most recently observed PCR. For a more refined rate allocation, we divide the PCR interval further and propose a PCR level scheme in this paper. Assume that in the i-th PCR interval (PCR_i, PCR_{i+1}) within a frame, the bit streams are divided into n_i individually decodable packets denoted as $P = \{P_{i,1}, \cdots, P_{i,j}, \cdots, P_{i,n_i}\}\,(1 \le j \le n_i)$. A header is appended to each packet which contains a timestamp indicating the correct transmission epoch denoted as $T = \{T_{i,1}, \cdots, T_{i,j}, \cdots, T_{i,n_i}\}\,(1 \le j \le n_i)$. The values of the timestamp can be calculated through two consecutive PCR values and the length of packets between them.

$$T_{i,j} = \begin{cases} PCR_i, & j = 1 \\ PCR_i + \dfrac{\sum\limits_{k=1}^{j-1} L(P_{i,k})}{\sum\limits_{m=1}^{n_i} L(P_{i,m})} \times (PCR_{i+1} - PCR_i), & 2 \le j \le n_i \end{cases} \tag{1}$$

where $L(P_{i,j})\,(1 \le j \le n_i)$ is the length of $P_{i,j}\,(1 \le j \le n_i)$. The value of $L(P_{i,j})$ here can be variable below 1 500 bytes, the ethernet max transport unit (MTU). During the period of high bit rate, $L(P_{i,j})$ is set as a smaller value to minimize the visual impact because of packet loss in network congestion. A larger value of $L(P_{i,j})$ is set when the bit rate is low. The sum of $L(P_{i,j})$ should be equal to the amount of data between PCR_i and PCR_{i+1}, as shown in the following equation.

$$\sum_{j=1}^{n_i} L(P_{i,j}) = b(PCR_{i+1}) - b(PCR_i) \tag{2}$$

where $b(PCR_i)$ and $b(PCR_{i+1})$ are the byte-orders of i-th and $(i+1)$-th PCR's in a video stream.

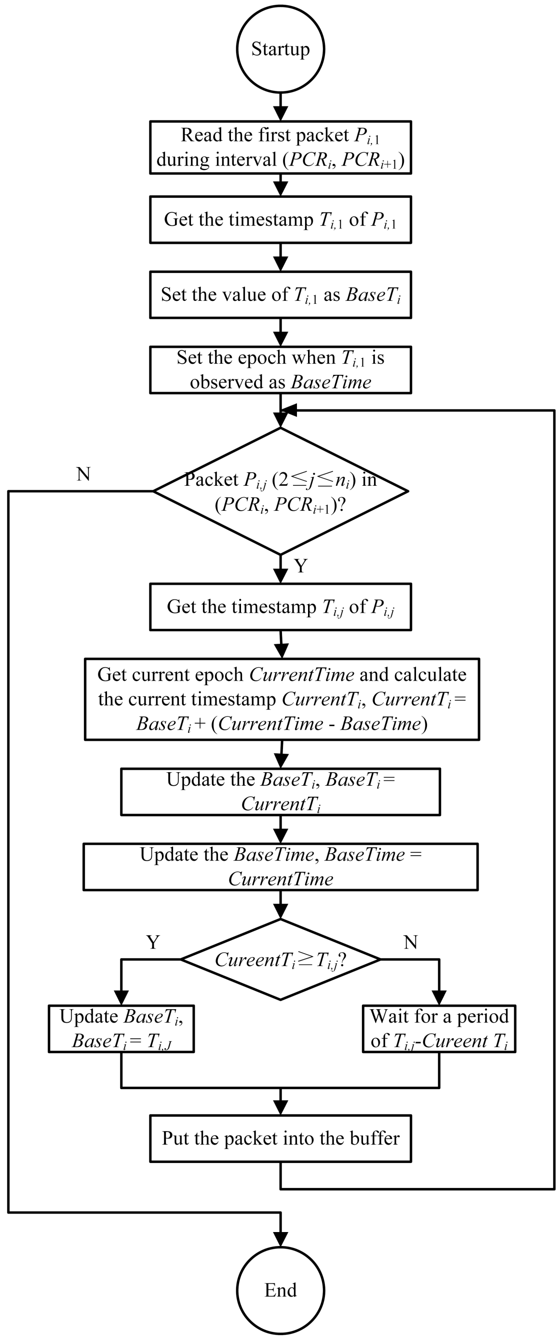


Fig. 2 The control diagram of PCR level rate allocation

The operation of the PCR level rate allocation is shown in Fig. 2 (see front page), where $BaseT_i$ denotes the reference timestamp used to calculate the current timestamp $CurrentT_i$, and $BaseTime$ and $CurrentTime$ are the timing information obtained from the operating system, respectively. To reduce the computation error, $BaseT_i$ and $BaseTime$ are updated during each cycle of packet transmission. Also the protocol of network time protocol (NTP) should be used to regulate the system time accuracy periodically. Fig. 3 depicts the transmission of packet $P_{i,j}$, where solid line represents the case of ahead schedule while dashed line represents the case of behind schedule. Different cases will yield the regulation of transmission rate accordingly.

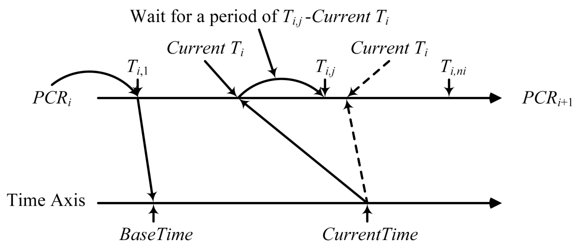


Fig. 3 The transmission of packet $P_{i,j}$

The refined mechanism of PLRC makes the decided rate reflect the variation of video streams more detailedly and reduces the difference between the decided rate and the real bit rate. This is more important for the transmission of those video streams which contain lots of high-motion frames and are encoded at a much higher bit rate with larger variation range.

1.2 Source buffer controller

Some packets must be selectively discarded and not transmitted when the network bandwidth can not accommodate the transmission of all packets. For this purpose, we propose an SBC module implemented at the application layer of the sender which dynamically and intelligently discards packets from its headend and this scheme is depicted in Fig. 4.

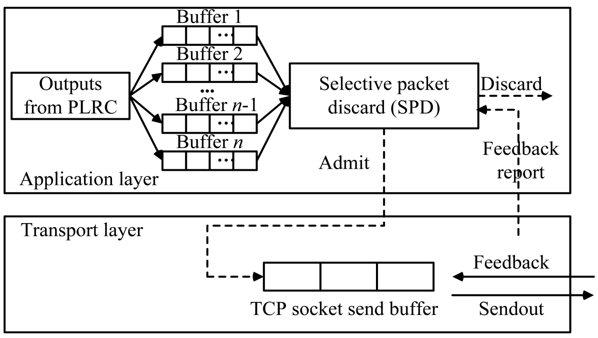


Fig. 4 Source buffer controller

The SBC actually consists of multiple buffers and each of such buffers works like a FIFO queue with a different importance level or priority. The packets coming from PLRC are placed into different buffers according to the priority mark included in the packet header. The definition of the priority here can be very generic. For example, we can have 3 queues for I, P, and B frames separately, and the queue for I frames has the highest priority. It can also work in a more refined way. For a specific GOP encoded as IBBPBBPBBPBBPBBP, we have $1 + 2 \times 5 = 11$ levels. For layered video, different layers can directly be used as importance levels. The packets with different importance levels wait in the corresponding queues until they reach the headend of the buffer and a decision is then made by the selective packet discard (SPD) module whether the corresponding packet should be passed towards the transport layer or is simply discarded.

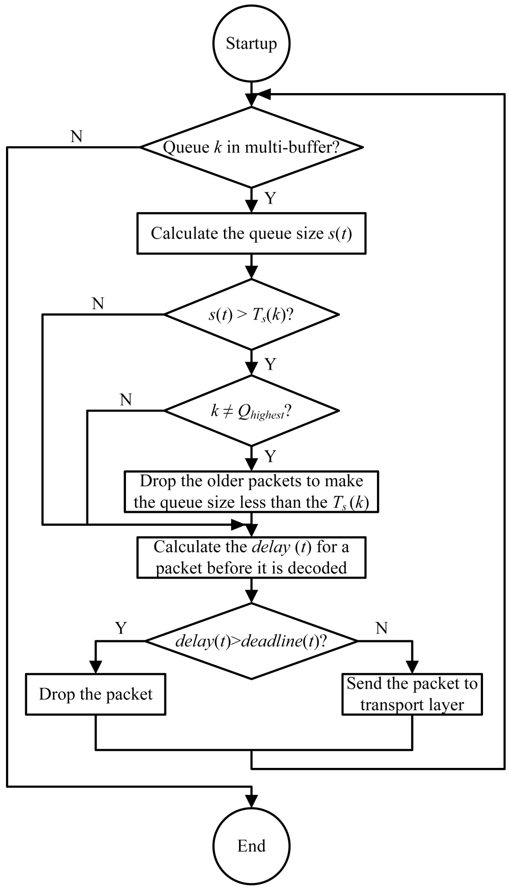


Fig. 5 The control diagram of selective packet discard

The operation of the SPD module is regulated by the flow shown in Fig. 5, where $s(t)$ is the instantaneous source buffer size, $T_s(k)$ is the source buffer threshold indicating the buffer size at which the drop policy starts being enforced, $Q_{highest}$ is the mark of the queue with the highest priority, $delay(t)$ is the delay for a packet before it is decoded and played at the client, and $deadline(t)$ is the playback deadline for this packet. Here, the PLRC module can be viewed as "producer", whereas the SPD module can be viewed as "consumer". In the classical "producer-consumer" problem, the "producer" has to wait when there is no empty place in the buffer to guarantee the transmission in right order, but ignores the real-time characteristics of the video transmission. In Fig. 5, the SPD module first checks the queue size before it deals with the packet at headend of the buffer. If the queue size exceeds the threshold, the older packets will be discarded to reduce the queue occupancy until it is less than a threshold, which enables the PLRC module to put the newer packets into the queue without delay and minimize the reconstruction distortion of the presentation at the clients. In this way, the pushing rate of the queue is driven by the "producer" and thus can reduce the coupling relationship between "producer" and "consumer". But for the queue with the highest priority, this kind of discard should not be executed because little loss of key-frame packets will lead to a severe degradation of the presentation quality at the client side. In addition, the choice of the source buffer threshold $T_s(k)$ is important for the overall system performance. Having a small threshold will lead to unnecessary packet drops at the source buffer, while having a large threshold will increase the overall delay and eventually cause the receiver buffer underflow. It should be better to set a different $T_s(k)$ for a different level buffer, with a lower level buffer having a smaller threshold while a higher level buffer having a larger threshold. For a given level buffer, $T_s(k)$ should also vary dynamically with the network status. If there is congestion, $T_s(k)$ is decreased, otherwise, $T_s(k)$ is increased. In our framework, the network congestion degree is measured through rate of change of receiver buffer size included in the feedback report, which will be discussed in the next section.

Another kind of discard happens when the packet at the headend of the queue is found to be unable to meet the playback deadline. For a packet, $delay(t)$ can be calculated as follows:

$$delay(t) = D_{i,s} + D_{i,tcp} + D_{i,n} + k(t) \times F(t) \tag{3}$$

where $D_{i,tcp}$ and $D_{i,n}$ are the delays in the TCP buffer and in the network for packet i, respectively, $D_{i,s}$ is the waiting time or the shaping delay in the source buffer for packet i, $k(t)$ is the current number of GOP in the client buffer that waits to be decoded, and $F(t)$ is the current playback duration for one GOP.

$D_{i,s}$ is the difference between the injection epoch for the i-th packet to the source buffer and the discard decision epoch for the ith packet, which can be obtained through PLRC module and SPD module separately. The number of GOPs in the client buffer and playback duration are available from the feedback. $D_{i,tcp}$ can be ignored.

We need to estimate the current single trip time from the server to the clients as follows:

$$D_{i,n} = \delta \times D_{i,n} + (1-\delta) D_{i-j,n} \tag{4}$$

where $0 < \delta < 1$ is the regulatory factor and $D_{i-j,n}$ is the most recently obtained network delay from the feedback report. In our experiments, we have observed that $\delta = 0.8$ is the most suitable value. According to the playback schedule, it is easy to calculate whether the packet can meet the deadline or not. If not, this packet will be discarded to decrease the requirement of the bandwidth. This scheduling works from the lowest level buffer to the highest level buffer to discard the least important data first.

1.3 Playback buffer controller

The estimation of current single trip time from the server to the clients in the above subsection may be not right, which makes few undiscarded packets arrive at the clients after their deadline. The playback buffer is used to minimize the effects because of the estimation error. Once a packet is found to miss the playback deadline, it will be discarded. Also, the rate of change of playback buffer size and the instantaneous playback rate are fed back to the server periodically. For the server side, since the feedback information of instantaneous playback buffer size provides some knowledge of congestion status about the recent past due to the network delay, it is rate of change of playback buffer size that gives a better understanding of the present. We use the following equation to calculate the rate of change of playback buffer size.

$$R_p = \frac{q_t - q_{t'}}{t - t'} \tag{5}$$

where q_t and $q_{t'}$ are instantaneous buffer sizes at t and t'. Since we allow reasonably long intervals for measuring t and t', the denominator in (5) is never close to zero and R_p measurement is reasonably freed from measurement noise.

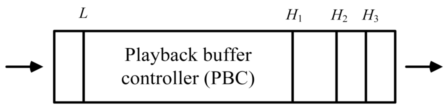


Fig. 6 Playback buffer controller

At the same time, we can achieve adaptive playback rate through the observation of playback buffer occupancy. Compared with the AMP[3], our adjustment algorithm introduces control theory and over-boundary strategy. It uses four threshold values as shown in Fig. 6, which divide the playback buffer into five levels. The playback rate adjustment depends on which level the current buffer size belongs to. To smooth the jitter of received data rate, we use the following recursion function:

$$\overline{q_t} = \gamma \overline{q_{t-1}} + (1-\gamma) q_t \tag{6}$$

where $0 < \gamma < 1$ is the smoothing factor that determines how sensitive our algorithm should be to instantaneous buffer jitter. In our experiments, we have observed that $\gamma = 0.6$ is the most suitable value. After the process, we get the filtered buffer size denoted as $\overline{q_t}$, which is used to make the playback adjustment as below.

1) $L \leq \overline{q_t} \leq H_1$: No adjustment should be made in this case. The selection of L and H_1 may not be too near to the center of receiver buffer to avoid frequent adjustment of playback rate, which is not very welcome due to visible change in display quality. But extreme end values of L and H_1 are also not expected, which may cause underflow and overflow during heavy jitter before making adjustment.

2) $\overline{q_t} < L$: The playback rate should be reduced in this case due to the possibility of underflow. Assume that the normal playback rate is V_d and the adjusted rate is $V_d' = V_d \times \alpha$, where α is the reducing factor. The fixed value of α is obviously not suitable to the variable buffer status. A simple selection is the proportional controller as follows:

$$\alpha = \frac{\overline{q_t}}{L}, \quad 0 < \overline{q_t} < L \tag{7}$$

To make a better adjustment, we choose the proportional-integral (PI) controller[13], which can ensure the steady state at the critical point and avoid frequent change of playback rate. Considering the speciality of video transmission, we propose an over-boundary adjustment mechanism based on PI controller. With the start of playback rate reduction, $\overline{q_t}$ is increased to the level $[L, H_1]$ gradually. Contrary to adjusting the playback rate immediately, we continue the rate reduction until $\overline{q_t} \geq (H_1 + L)/2$, which can prevent the playback rate from switching frequently at L further. The mechanism can be formulized as follows:

$$\Delta\overline{q_t} = \overline{q_t} - \overline{q_{t-1}} \tag{8}$$

$$e_t = \overline{q_t} - L \tag{9}$$

$$\alpha = \min(\frac{K_i\Delta\overline{q_t} + K_p e_t}{L}, \alpha_{\max}), \quad 0 < \overline{q_t} < \frac{H_1 + L}{2} \tag{10}$$

where $\Delta\overline{q_t}$ and e_t reflect the buffer variability and buffer level, respectively, K_i and K_p are the integral and the proportional gain of the PI rate controller, and $\alpha_{\max} = 0.98$ is proved to be the most suitable value in our experiments.

3) $H_1 < \overline{q_t} < H_2$: The playback rate should be increased in this case due to the possibility of overflow. The adjusted rate is denoted as $V_d' = V_d \times \beta$, where β is the increasing factor and similar to 2), the over-boundary adjustment mechanism based on PI controller is also used. By way of contrast, the playback rate acceleration will aggravate the system burden, which may limit the value of β. Assuming that the maximum decoding capability is θ, we have

$$\Delta\overline{q_t}' = \overline{q_t} - \overline{q_{t-1}} \tag{11}$$

$$e_t' = \overline{q_t} - H_1 \tag{12}$$

$$\beta = \max(\min(\frac{K_i'\Delta\overline{q_t'} + K_p' e_t'}{H_1}, \theta), \beta_{\min}), \quad \frac{H_1 + L}{2} < \overline{q_t} < H_2 \tag{13}$$

where $\Delta\overline{q_t}'$ and e_t' reflect the buffer variability and buffer level, respectively, K_i' and K_p' are the integral and the proportional gain of the PI rate controller, and $\beta_{\min} = 1.02$ is proved to be the most suitable value in our experiments.

4) $H_2 \leq \overline{q_t} \leq H_3$: The B frames should be discarded in this case because it is unable to avoid buffer overflow only through the playback rate acceleration discussed above. For a typical GOP shown in Section 1.2, the active B-frame discarding will make the playback rate triple without mosaic appearing.

5) $\overline{q_t} > H_3$: Only I frames should be decoded in this case. For a typical GOP shown in Section 1.2, the selective I-frame decoding will accelerate the playback rate fifteen times without mosaic. To prevent physical overflow, another conservative approach is to halt streaming for a short period in this case.

As analyzed in this section, the full utilization of the playback buffer can minimize the playback degradations or the changes perceived by the users that are caused by changes of the network.

2 Simulation results

Our simulations of video transmission were carried out using the ns-2 simulator. The simulation topology is depicted in Fig. 7, where there are three links (R1-R2, R2-R3, R3-R4). Each link has a variable capacity which depends on the simulation scenario. To simulate the real network environment, various cross-traffic (FTP flows and HTTP flows) combinations have been used to produce the competing flows.

We used the standard test video sequences, stefan, encoded by MPEG-2 in the simulation. These video sequences were in the YUV 4:2:0 format with 352×288 pixels per frame and 30 frames per second. The GOP consisted of 16 frames in the order of IBBPBBPBBPBBPBBP. A total of 256 frames were encoded into 16 GOPs, and the average bit rate was 1.36 Mbps. The 16 GOPs were repeatedly sent from the server. 6 GOPs were prefetched before the playback began, and so the initial playback delay was approximately 3.2 seconds. The feedback interval was 16/30 seconds in our simulation. At the end of the playback of a GOP, the feedback information was sent back to the source.

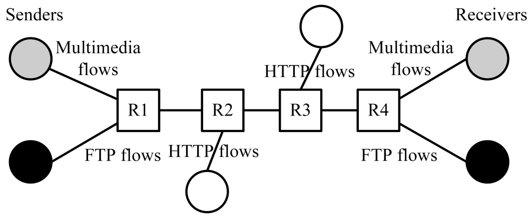


Fig. 7 Simulation topology

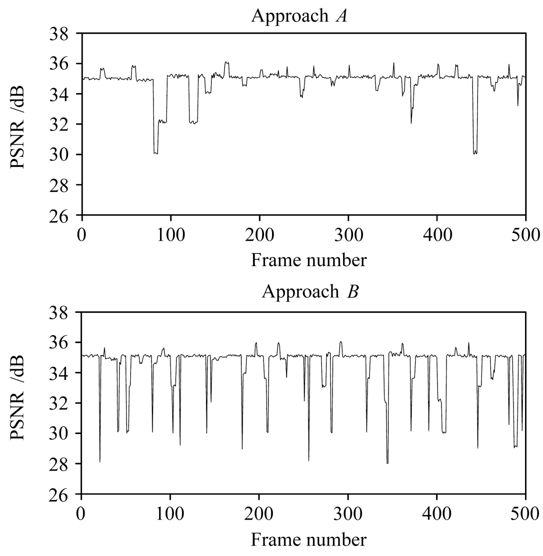


Fig. 8 PSNR comparison of video sequences under the link bandwidth of 5 Mbps

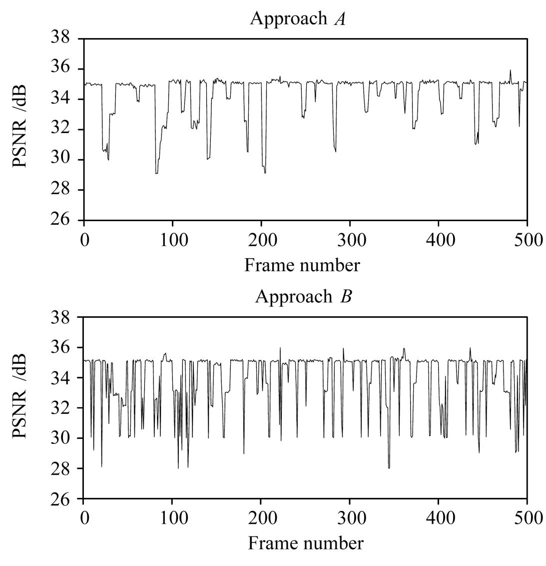


Fig. 9 PSNR comparison of video sequences under the link bandwidth of 4 Mbps

To illustrate the advantages of our approach, we compared our proposed framework (denoted as approach A) with the efficient bandwidth utilization approach proposed in [14] (denoted as approach B) in terms of PSNR under the same network environment.

Simulations were run under different link bandwidth to test the performance of our framework under different network congestion scenarios and packet loss ratios. The comparison of PSNR is displayed in Fig. 8 (see front page) with the link bandwidth of 5 Mbps. It contains 500 frames, which lasts about 16 seconds. It can be seen from the figure that under the same network conditions, our approach can get better PSNR values than those of approach B. Another scenario with a larger packet loss ratio is compared in Fig. 9 with the bandwidth of 4 Mbps.

Table 1 gives the average PSNR of the video frames under different scenarios. From this table, it can be easily seen that our proposed scheme outperforms Approach B under all bandwidth limitations. Moreover, we have a larger PSNR gain when the network is more congested. In addition, our framework can achieve smoother variation of the displayed frames. Figs. 8 and 9 show that the PSNRs of the neighboring frames change more smoothly and less frequently in Approach A, which gives a better presentation effect to the users.

Table 1 Comparison of PSNR in different scenarios

Link Bandwidth	PSNR (dB) Approach A	PSNR (dB) Approach B
6 Mbps	35.18	34.82
5 Mbps	34.81	34.23
4 Mbps	33.96	32.92

To compare the computational complexity of our approach with that of Approach B, we ran both of them under the same hardware configurations to observe the changes of CPU occupancies at the sender and receiver. The CPU frequencies of the sender and receiver were set as 800 MHz and 500 MHz, respectively, and the duration was 500 s. As shown in Fig. 10, Approach A outperforms Approach B at both sides with lower CPU occupancies and smoother variations of CPU occupancies. This demonstrates that our approach has lower computational complexity than the previous related work.

3 Conclusions

In this paper, we have presented an integrated video communication framework for VBR video delivery in a congested network. This framework regulates the transmission rate through a refined rate allocation algorithm based on PCR value embedded in the video streams. Multiple buffers for different importance levels, along with an intelligent selective packet discard algorithm, are applied at the source, via which the scheduling scheme differentiates packets with different priorities. At the receiver side, an over-boundary playback rate adjustment mechanism based on PI controller is incorporated to maximize the displayed video quality in response to congestion. In the simulations, we compared it with another approach and the simulation results have shown that the proposed integrated framework can achieve larger PSNR improvement when the network is more congested, and the quality of the decoded frames is smoother, which is more favorable to the users. In addition, we also demonstrate that our approach can get lower computational complexity through the observation of CPU occupancies at both sender and receiver sides.

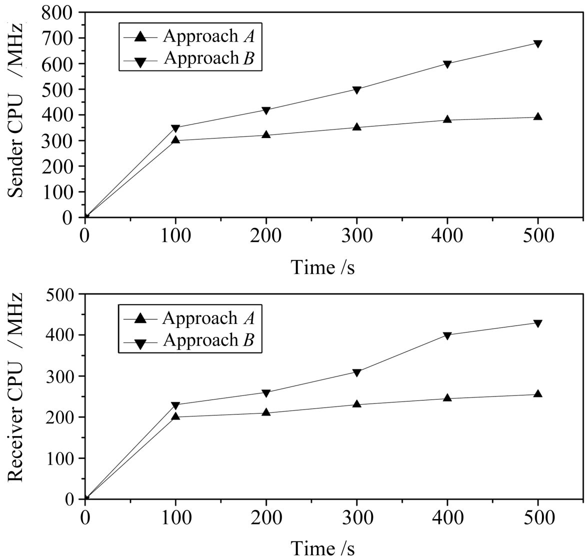


Fig. 10 The CPU occupancies of both sender and receiver

References

1 Wu D P, Hou Y T, Zhang Y Q. Transporting real-time video over the internet: challenges and approaches. *Proceedings of the IEEE*, 2000, **88**(12): 1855−1877

2 Chou P A, Miao Z R. Rate-distortion optimized streaming of packetized media. *IEEE Transactions on Multimedia*, 2006, **8**(2): 390−404

3 Kalman M, Steinbach E, Girod B. Adaptive media playout for low-delay video streaming over error-prone channels. *IEEE Transactions on Circuits and Systems for Video Technology*, 2004, **14**(6): 841−851

4 Luo H L, Shyu M L, Chen S C. A multi-buffer scheduling scheme for video streaming. In: Proceedings of IEEE International Conference on Multimedia and Expo. Amsterdam, The Netherlands: IEEE, 2005. 1218−1221

5 Wong C W, Au O C, Lam H K. PID-based real-time rate control. In: Proceedings of IEEE International Conference on Multimedia and Expo. Taiwan, China: IEEE, 2004. 221−224

6 Wagner J P, Frossard P. Playback delay optimization in scalable video streaming. In: Proceedings of IEEE International Conference on Multimedia and Expo. Amsterdam, The Netherlands: IEEE, 2005. 860−863

7 Kang S H, Zakhor A. Effective bandwidth based scheduling for streaming multimedia. In: Proceedings of IEEE International Conference on Image Processing. Barcelona, Spain: IEEE, 2003. 633−636

8 Tao D Y, Cai J F. A novel reverse frame selection scheme for video streaming over VBR channels. In: Proceedings of IEEE International Conference on Multimedia and Expo. Toronoto, Canada: IEEE, 2006. 1845−1848

9 Krastic C, Walpole J, Feng W C. Quality-adaptive media streaming by priority drop. In: Proceedings of the 13th International Workshop on Network and Operating System Support for Digital Audio and Video. Monterey, USA: ACM, 2003. 112−121

10 Kamimura K, Hasegawa T, Hoshino H, Ano S. A practical multicast transmission control method for multichannel HDTV IP broadcasting system. In: Proceedings of Pacific Rim Conference on Multimedia. Jeju Island, Korea: Springer-Verlag, 2005. 429−440

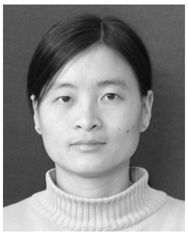
11 Du D H C, Hsieh J, Lee H J, Chang T S. PCR-assist CBR for delivering pre-recorded MPEG-2 transport streams. In: Proceedings of IEEE International Conference on Multimedia Computing and Systems. Ottawa, Canada: IEEE, 1997. 646−647

12 Zhang Y H, Huangfu W, Xu J, Li K H, Xu C Q. Integrate rate control and buffer management for scalable video streaming. In: Proceedings of IEEE International Conference on Multimedia and Expo. Beijing, China: IEEE, 2007. 248−251

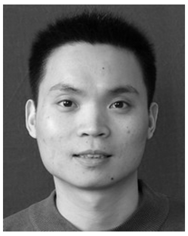
13 Hollot C V, Misra V, Towsley D, Gong W B. On designing improved controllers for AQM routers supporting TCP flows. In: Proceedings of IEEE International Conference on Computer Communications. Anchorage, Alaska, USA: IEEE, 2001. 1726−1734

14 Luo H L, Shyu M L, Chen S C. An end-to-end video transmission framework with effcient bandwidth utilization. In: Proceedings of IEEE International Conference on Multimedia and Expo. Taiwan, China: IEEE, 2004. 623−626

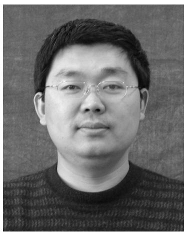
ZHANG Yuan-Hai Ph. D. candidate at Institute of Software, Chinese Academy of Sciences. His research interest covers video transmission, video coding, and streaming technology. Corresponding author of this paper.
E-mail: yuanhai02@ios.cn



LI Kai-Hui Ph. D. candidate at Institute of Software, Chinese Academy of Sciences. Her research interest covers video transmission and multimedia communication. E-mail: kaihui01@ios.cn



XU Chang-Qiao Ph. D. candidate at Institute of Software, Chinese Academy of Sciences. His research interest covers video transmission and P2P multimedia streaming.
E-mail: changqiao@ios.cn



SUN Li-Min Professor at Institute of Software, Chinese Academy of Sciences. His research interest covers wireless sensor networks and mobile IP networks.
E-mail: limin@ios.cn