Using Fuzzy Logic Controller to Implement Scalable Quality Adaptation for Stored Video in DiffServ Networks

Xiaoyan Wang, Dejian Ye, Qiufeng Wu
Department of Automation, Tsinghua University
Beijing 100084, China,
E-mail: wxyan00@mails.tsinghua.edu.cn, yedejian99@mails.tsinghua.edu.cn, wqf-dau@tsinghua.edu.cn

Abstract

In this paper a video quality adaptation system is proposed for video servers to provide VOD services in DiffServ networks. This system keeps the receiver buffer from overflow or underflow by using feedback control, which guarantees a stable and continuous playback quality despite the rapid fluctuations of the available bandwidth. According to the characteristics of the DS networks, this control target is divided into sub-targets for per-class. Unlike existing schemes, the system implements quality adaptation independently of the underlying congestion control mechanisms since it gets the network status through the information about the receiver buffer occupancy provided by an observer running at the server. This greatly simplifies the implementation and makes the system more popular in networks. The system uses fuzzy logic to handle the nonlinear characteristics, time-varying delay and uncertainty of the model.

Key words: quality adaptation, DiffServ (DS) network, and fuzzy logic control.

I. INTRODUCTION

Many multimedia applications on the Internet, such as real-time video and stored video on demand (VOD), are becoming more and more popular than traditional ones. But there is a paradox between the stringent network bandwidth and these increasing applications, especially for their rigid requirements on quality of service (QOS). How to make maximal use of the available bandwidth or, in other words, how to get optimal QOS with the bandwidth available, is an urgent issue for both the intermediate networks and end systems.

To address this issue, three techniques, namely, scalable video coding, network-aware adaptation of end systems, and adaptive QOS supports from networks, have been developed [1]. The focus of these three techniques is bandwidth fluctuation. The control target of end systems is to adjust the video quality according to the fluctuation of bandwidth while keeping it at a relatively stable level at the same time, which requires a scalable video encoding scheme. The control target of networks is to decrease the fluctuation of bandwidth and provide QOS guarantee to applications.

Following scalable video encoding schemes, a raw video sequence is compressed into several layers: one base layer, and several enhancement layers. The base layer, which can be independently decoded, provides basic video quality. The enhancement layers can only be decoded together with the base layer and they further refine the quality of the base layer. If the available bandwidth decreases, some layers are dropped (of course the base layer will not be actively dropped since the other layers can be decoded without it).

The adjustment of quality can only be performed by the end system with network-aware adaptation [2]. As the name implies, this end system consists of two elements: network awareness and adaptation. The former refers to having knowledge about the underlying networks status, especially the available bandwidth; the latter is to adapt the video streams to this status. Existing network-awareness or network-monitoring techniques can be classified by some criteria such as method of monitoring, monitoring frequency and replication of information [1]. How to implement adaptation and the result are decided by the methods of getting knowledge about current network status.

Transmission of the adapted video streams also requires the support of the networks. The networks should provide differentiated QOS to different applications, even to different parts of the same application, which are called adaptive services. Such services include the following functions: 1) reserving a minimum bandwidth to meet the demand of the base layer, and 2) adapting the enhancement layers to the available bandwidth. The DiffServ (DS) network proposed in 1998 by IETF is such one in which services are leveled to meet different QOS requirements [3]. As one model of the next generation Internet, the DS model will take the place of Best-Effort (BE) since the latter only provides the same service to all applications.

A whole quality adaptive framework is the combination of the three components: the sender, the networks and the receivers.

The topic of quality adaptation is proposed in the BE networks and becomes quite an active area. However, the DS networks are now becoming the new developing trend, in which services are provided on a per-class basis rather than per-flow. But different users that fall into the same level of service still interact as in the BE networks, which means the same problems about network resources still exist in the DS networks. So how to implement the quality adaptation in the DS networks is quite a new but important problem that few researchers have focused on. The key issue is how to guarantee the synchronization of the playback since the video stream is split into sub-streams, which are separately transmitted with different bandwidths.

In this paper we propose an end system running in a video server to implement video quality adaptation in the DS networks. In control theory, it is a system with feedback control. To achieve these targets shown above, the receiver buffer occupancy is used as feedback information, which can perfectly reflect the networks status and also represent the playback quality of video. Fuzzy logic controller is used in the closed-loop, which has excellent characters in such complex, uncertain system with time-varying delay. The system has four main advantages: 1) to be robust under the rapid fluctuation of networks bandwidth, 2) to successfully solve the problem of time-varying delay in performance, 3) to be popular in more networks because of its independency of the underlying congestion control mechanisms, and 4) can be implemented easily.

The rest of this paper is organized as follows. In Section II we describe the conditions and problems in a supposed environment we should face, and our scheme is introduced accordingly in Section III. Then in Section IV we build a model of the quality adaptation system, and discuss the control algorithm by outlining the key components. Based on the model and algorithm we try to do some analysis in several aspects in Section Ⅴ. Results of experiments are presented in Section Ⅵ. The experiment results numerically show that our controller can ensure very smooth playback quality. Finally Section Ⅶ concludes this paper.

II. QUALITY ADAPTATION PROBLEM SETTING

A video server providing pre-encoded stored streams to different clients is a typical multimedia application, where the playback quality of video is a direct reflection of QOS provided by the intermediate networks. An urgent target for the server is to adjust the quality requirements of video according to the real-time network status since the resources are so stringent. Short-term improvement and long-term smoothing of quality are the two tasks of quality adaptation for end systems. They can be accomplished in two ways: congestion control and quality control. For video streams, congestion control takes the form of rate control, which adapts the transmitting rate to the available bandwidth in the networks, while quality control adapts the bit rate. A rate shaper is used to match the rate of a pre-compressed video bit-stream to the target rate constraint (See Fig.1) [4,5].

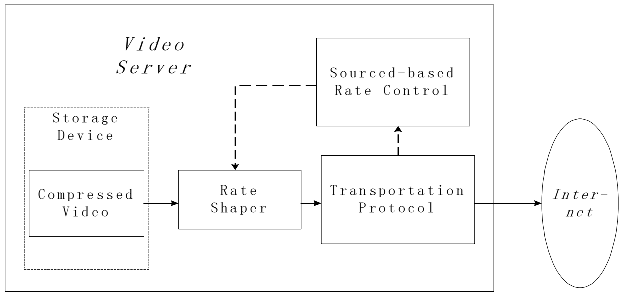

Fig.1. Conceptual model of video server

The key issue to short-term improvement is how to get the information about the networks status. In fact it is difficult to directly get the bandwidth because it changes rapidly and unpredictably. Some predictors and observers based on the models [6] or the probes [7] have been proposed, and many of such kinds of information about the sender buffer [8], the receiver buffer [9] and even the play rate [10] are used to suggest the networks status. These schemes are often dependent on the underlying congestion control mechanisms because they need special data provided by them. Whether this information about the networks status is precise will greatly affect the result of the adjustment.

Moreover, the perfect matching between the quality adjustment and changes of bandwidth will induce another issue: the video quality changes frequently and rapidly with the fluctuation of bandwidth, which will make the viewer feel uncomfortable. In fact, a video with medium but stable quality is more acceptable [11]. So, long-term quality smoothing is needed to eliminate the effects of bandwidth fluctuation. The receiver buffer is a better choice because it allows a “mismatch” between the rates of transmission and consumption. When there is sufficient bandwidth, more frames can be transmitted in advance and stored in the receiver buffer waiting for playback. If the bandwidth becomes stringent， the player can consume the frames stored in the receiver buffer at first instead of asking the server to decrease the quality of transmitting frame at once. Thus the quality of video can be maintained at a quite stable level.

These problems come into being in the BE networks. While in the DS networks different levels of services are provided with the different networks bandwidth. Assuming every packet has its priority index which is decided based on the content [12], a classifier is needed to map the packets into classes for different level of QOS based on the indexes. This function is helpful to implement service adaptation performed in the networks. Packets with higher priority will be mapped into the class provided with higher level of QOS. Of course packets belonging to base layer should be provided with the highest level of QOS, while the highest enhancement layer with lowest level. Thus quality adaptation should be separately provided for classes with different levels of QOS requirements. This is called quality adaptation for classes, which seems more complex than in the BE networks. How to credit the overall effect of quality adaptation, how to tackle the relations among adaptations for per class and how to make tradeoff between them still remain open.

III. Our Scheme About How To Implement Quality Adaptation

To address these issues shown above, we propose a new scheme to perform quality adaptation based on the information of the receiver buffer occupancy in the DS networks because the occupancy directly reflects the matching relation between the available bandwidth and the playback quality. When there is sufficient bandwidth, the system will increase the video bit rate for better playback quality; when the bandwidth is stringent, the system will reduce the bit rate to keep a basic playback quality without disruption. If the system can properly adjust the bit rate in time, the receiver buffer occupancy will be kept at a relatively stable level, which means the change of playback quality can perfectly match the fluctuation of the bandwidth without resource wasting. Since the receiver buffer occupancy is the control object, from the point of view on control theory the system has good performance only if the control object promptly approaches the ideal reference without going beyond the changing limitation. So we seperately set an upper and a lower threshold to limit the change of receiver buffer occupancy, indeed the control target of the system is not only to keep the receiver buffer change between the two thresholds but also to keep it approach the reference stablely. In actual applications the playback will come to a pause if the receiver buffer occupancy is below the lower threshold. And the buffer's capability is bounded in wireless equipments and set-top boxes, so we also set an upper threshold of the receiver buffer.

In our design, we use the receiver buffer occupancy, instead of the receiver buffer rate, which are used in some previous works, as the control object. Here is our consideration. Such a system is a typical uncertain one with large delay. The introduction of differential control component in a traditional PID controller will introduce impetuous oscillations, which are not acceptable. In our design, the receiver buffer occupancy is used as the P component. So, the receiver buffer rate, which is the differential of the occupancy, plays the rule of D component. It is not accepted because of the abandonment of the differential control component [17]. In fact, fuzzy controller is a switched PID controller.

Summarily the receiver buffer occupancy is chosen for three reasons. Firstly, it can perfectly reflect the actual playback quality at the client: if it doesn't overflow or underflow, a continous and stable playback is achieved with no data lost or shortage. Secondly, it can also be used to smooth the playback quality. Thirdly, it can be precisely calculated at the server to reflect the bandwidth status. So the system is built in form of closed-loop in control theory, and the receiver buffer occupancy is set as the feedback. Thus we can achieve quality adaptation by controlling the receiver buffer occupancy. With the control function the receiver buffer will approach the reference set in advance despite of other disturbances.

To reflect the bandwidth provided for different classes, packets belonging to the same class can share a division in sender/receiver buffers called as buffer share of the class. If each of the receiver buffer share of class doesn't underflow or overflow, the total receiver buffer doesn't either. So, we can divide the control target into sub-targets which are separately accomplished in the closed-loops for per class. In our design, quality adaptation is achieved on a per-class basis, and the overall effect is an accumulation of each subsystem under some conditions. The number of subsystems is K if there are K levels of service provided in the DS networks and packets are mapped into K classes. The physical model of our design is shown in Fig.2.

IV. Model of This System and Control Algorithm

Firstly we give an abstract model of this quality adaptation system in the form of block diagram(See Fig.3). The receiver buffer observer estimates the receiver buffer occupancy at the server. This information is set as feedback of the closed-loop in control theory. All variables in Fig.3 will be explained later. Here is how the system works.

The trigger for the system to work is the threshold γ of the sender buffer. When the occupancy of the sender buffer falls below this threshold, the quality controller begins to work.

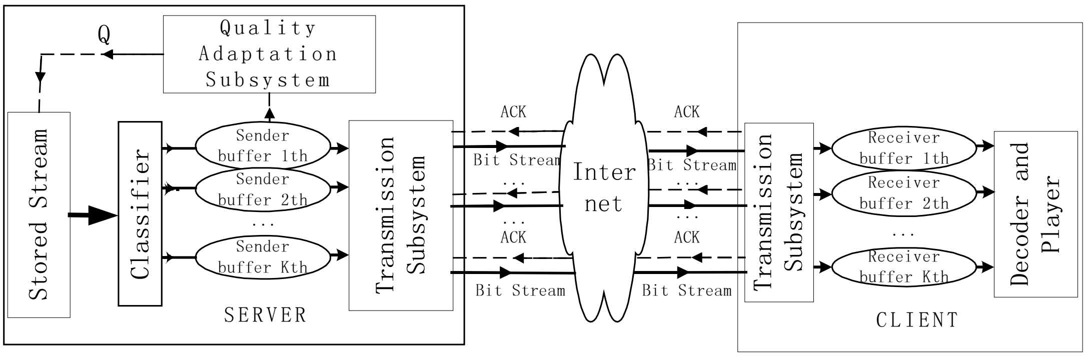

Fig.2. Physical model of quality adaptation system

The video quality index Q is the output of controller, which is defined as the reduction of the encoded rate from the ideal rate [13]. When Q is bigger, more layers of the video stream will be transmitted, and a better quality of video can be provided. That is Q is proportional to the encoded bit rate $R(t)$. Since in our design we assume 30 frames are played per second, which means the encoded frame size is proportional to encoded bit rate, Q is proportional to encoded frame size. So the system changes the size of frame i based on Q_i to adjust the video quality. We simpifly the curve of $Q-f$ as a linear function, then get

$$f_i = f_i^0 \cdot Q_i = f_i^0 \cdot \sum_{k=1}^{K} Q_{ik} \tag{1}$$

where f_i^0 is the ideal size of frame i ,and f_i is the modulated size. The group of index Q_{ik} are reference quality indexes for parts of frame i belonging to different classes.

The modulated frame i is then sent into the classifier, in which it will be split into several packets with some priority. These packets are mapped into different classes according to the ratio r_{ik} and their priorities. The mapping ratio r_{ik} is decided as $r_{ik} = Q_{ik} (k = 1,...,K)$. Thus in class k, the total size of packets in frame i is $f_{ik} = f_i^0 \cdot r_{ik}$.

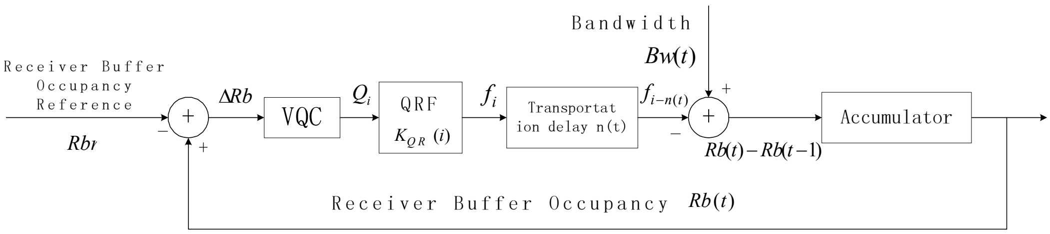

Fig.3. Block diagram of quality adaptations ystem

After this process, these packets begin to queue in the corresponding sender buffer shares. Each new group of incoming packets increases $Sbcounter$, the sender buffer counter, by one, since the counter records the number of the frames sent into the sender buffer. The underlying congestion control system gets packets from the sender buffer, then transmits them through the networks. Different bandwidths are provided for packets belonging to different classes. The sender buffer is thus used to match the bit rate and the transmission rate.

At last, these packets reach their corresponding receiver buffer shares at the client, then the occupancies of the receiver buffer shares become $Rb_k(t)(k = 1,...,K)$. After waiting in the receiver buffer for some time, these video packets are disposed by the decoder and consumed for playback.

This will increase $Rbcounter$, the receiver buffer counter, by one, since this counter records the number of frames being consumed for playback. The delay time comes into being due to the effect of buffers and transmission in the networks. But the transmission delay is omitted since it is too small compared with the buffer effect.

The variable t is the sample time point. The control algorithm is implemented discretely. So $Bw(t)$ is an accumulation of the network bandwidth from $(t-1)$ to t, and $Bw_k(t)$ is such an accumulation for class k. So does $Rb_k(t)/Sb_k(t)$, the receiver/sender buffer share occupancy.

The blocks in Fig.3 will be outlined as follows.

A. Receiver Buffer Observer

The receiver buffer is the key to achieve quality adaptation, so the occupancy of the receiver buffer $Rb(t)$ must be obtained in real-time as precisely as possible. But it is very difficult for the client to directly report the receiver buffer occupancy to the server on time. However, the video server has the knowledge about how many bytes of data are sent and how many are consumed for playback, thus we can precisely calculate $Rb(t)$ in real-time based on the dynamic characters of the receiver and sender buffers. Following this idea, we build observer to output occupancy of these receiver buffer shares for class: $Rb_k(t)$. At first, we give a mathematical description about the occupancy of sender buffer share for class k:

$$Sb_k(t)-Sb_k(t-1)=\begin{cases} f_{ik}-Bw_k(t), & \text{If frame } i \text{ is put into sender buffer share } k\text{th at } t. \\ -Bw_k(t), & \text{else.} \end{cases} \quad (2)$$

And the mathematical description about the occupancy of the receiver buffer share for class k is

$$Rb_k(t)-Rb_k(t-1)=\begin{cases} Bw_k(t)-f_{j_k}, & \text{If frame } j \text{ is removed from sender buffe share } k\text{th at } t. \\ Bw_k(t), & \text{else.} \end{cases} \quad (3)$$

We add Eq.(2) and Eq.(3) together to get the receiver buffer share occupancy:

$$Rb_k(t)=Rb_k(t-\tau)+Sb_k(t-\tau)-Sb_k(t)+f_{ik}-f_{jk}\,. \quad (4)$$

If there is no frames put into the sender buffer, set $f_{ik}=0$; and if no frames consumed in the receiver buffers, $f_{jk}=0$. Since we can get $Sb_k(t)$ easily at the server, $Rb_k(t)$ can be precisely calculated by Eq.(4) in real-time. The bandwidth of networks $Bw_k(t)$ is eliminated in Eq.(4), which means the current bandwidth of the networks is not directly needed. Thus we can build the observers to output the occupancy of these receiver buffer shares for classes.

In our design, the impact of packet loss on the receiver buffer observer can be ignored if the packet loss ratio in networks is below some threshold [17].

B. Video Quality Controller (VQC)

It is the VQC that performs the control function of the system. So the VQC is the key component of the system.

Simply saying, when the receiver buffer is far below the reference, the controller will output a smaller quality index to correct the difference. On the contrary, the controller will output a bigger quality index when the difference between the receiver buffer and the reference is big.

The control targets are achieved separately: keeping the receiver buffer share of classes from underflow or overflow. Since every share does not overflow or underflow, the total receiver buffer must have a reasonable occupancy.

Removing the impaction of the bandwidth fluctuation on the video quality demands the control function be performed in a long-term observation of the buffer occupancy changes. So the accumulation of the difference ($\sum_0^t \Delta Rb_k$) is specified as the other input to the VQC as the supplement of ΔRb_k.

Fuzzy logic controllers are used in this system because they have been found to excel than the classical controller (such as the proportional-integral-differential (PID) controller) in the systems, which are very complex, highly nonlinear and with parameter uncertainty that just accord with the characters of our design. We may view a fuzzy logic controller as a real-time expert system that employs fuzzy logic to analyze input to output performance.

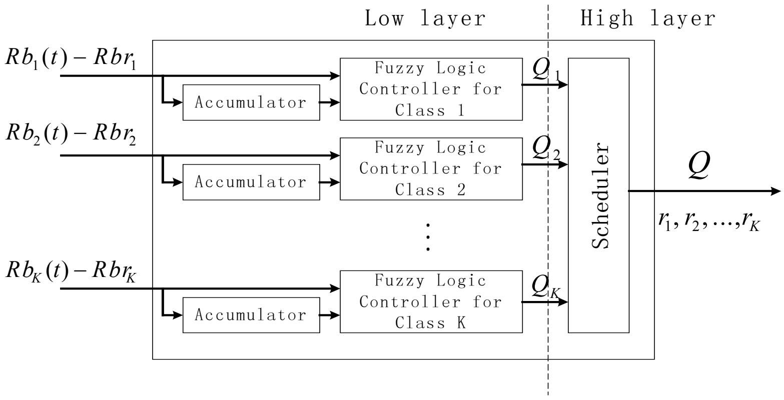

Fig.4. Architecture of VQC

As shown in Fig.4, the VQC is divided into two layers according to its logic function. Controllers for per class construct the low layer. They outputs the index $Q_k (k = 1,2,..., K)$ based on the receiver buffer shares $Rb_k(t)$. The scheduler in high layer will do an accumulation of the group of Q_k. Then the scheduler outputs the sum and a group of ratio r_k. In summary, the control algorithm is shown in Fig.5.

Control Algorithm

for $k = 1: K$;

$Q_k = g(\Delta Rb_k)$;

if $(\sum_{k=1}^{K} Q_k \le 1)$ { $Q = \sum_{k=1}^{K} Q_k$; $r_k = Q_k$ (for $k = 1: K$); }

else { $Q = 1$; $r_k = \dfrac{Q_k}{\sum_{l=1}^{K} Q_l}$ (for $k = 1: K$); }

end if

Fig.5. Control algorithm

In Fig.5, the function $g(x)$ refers to the class controllers' operation, which is designed with expert experiences. We adjust the criteria of fuzzy logic controller to get ideal performance in advance. Then the criteria are summarized as the experiences based on serials of experiments. For simplicity we assume every class controller uses the same $g(x)$.

C. Quality-Rate Function(QRF)

In Fig.4, the QRF block models Rate Adjustment and depicts the relationship between the video quality and the bit rate. For simplicity, this research uses the ratio Q between modulated and original rate to represent video quality. Thus, for fine-grain Rate Adjustment technique, the QRF becomes a time-varying ratio K_{QR}. Frame size f is the accumulation of bit rate, and can be calculated by $f = K_{QR} \cdot Q$ with 0 and 1 as the saturation points of Q. To keep these subsystems independent, we simplify the curve of $Q - f$ as a linear function. This means K^i_{QR} is a constant for frame i, but K^i_{QR} and K^j_{QR} may be different if $i \neq j$. The nonlinear effects can be removed by fuzzy controllers in the VQC.

D. Transportation Delay

Modulation is implemented on frame i. While at that time the frame being consumed at the client is j, not i, the difference between i and j is the number of delayed frames: $n(t)$, which is time-varying, dependent on the dynamic performance of the sender and receiver buffers. We can calculate $n(t)$ with the method shown in Fig.6. So, it is $f_j = f_{i-n(t)}$ that makes the occupancy of the receiver buffer change with the disturbance of the network bandwidth $Bw(t)$.

```
Do while Rb_k(t) ≠ 0
    if Sb_k(t) < γ
        Dispose the next frame and put it into sender buffers;
        Sbcounter++;
    End if
    Sbcounter--;
    Rbcounter++;
    n(t)=(sbcounter-rbcounter)/30;   /*1/30 second is the period of a frame lasting while playing*/
    t = t + 1;
next
```

Fig.6. Calculation process for numbers of delay frame $n(t)$

D. Special Situation For $K = 1$

$K = 1$ means the quality adaptation system works in the BE networks since there is only one level of QOS provided. That is the simplest case of the control algorithm.

Ⅴ. Analysis of The Control Algorithm

In this section we will analysis the relations between these closed-loops and that between our quality adaptation control and the underlying congestion control. At last we explain the expert experiences used to design the fuzzy logic rules.

A. Relation between these closed-loops for control

Following this algorithm, the overall control target is divided into a group of sub-targets. These sub-targets can be separately achieved in closed-loops for per class, which requires no relations among these loops.

f_{ik} is calculated based on two formulas: $Q_{ik} = g(\Delta Rb_k)$ and $f_{ik} = f_i^0 \cdot Q_{ik}$. We can only treat the calculation as a black box where the nonlinear elements are beyond our control, thus f_{ik} is decided only by ΔRb_k. Since there are no variables belonging to other loops affecting f_{ik}, every loop for class is independent to others. So a closed-loop of control for class has no impacts on other loops in our design.

Of course we can get this independence only if some assumptions are made to simplify this problem. At first we decompose the curve of $Q-f$ into two parts: linear and nonlinear parts though it is a typical nonlinear function in fact. The block of the QRF only refers to the linear part while the nonlinear part is seen as disturbance to the system which can be removed by the VQC.

In addition, the linear part used in the QRF is not overall linear because of the physical meaning of $Q_k(0 \le Q_k \le 1)$. 0 and 1 are both saturation points of $Q-f$. So we have to do some supplimentation to the control algorithm when $\sum_{k=1}^{K} Q_k > 1$ (shown in Fig.5). Unfortinately when the saturation point is reached, the ratio r_k can not be determined by the variables only belonging to the loop for class k. That means some relations exist among these closed-loops. But this situation ($\sum_{k=1}^{K} Q_k > 1$) will happen only when there is enough networks bandwidth provided for this frame's transmission, and in fact it does not happen so often because of the stringent networks bandwidth. So the work points of these closed-loops are in the area of nonsaturation, which means we can consider the curve of $Q-f$ as a linear function and no relations exist among these closed-loops under usual conditions.

B. Relation between quality adaptation and underlying congestion control mechanisms

Some existing quality adaptation mechanisms have close relations with underlying congestion control because they get the network status from the information provided by these control mechanisms. Since there are so many different underlying congestion control mechanisms exiting in the networks and new ones are just coming into being continuously, the dependence of a special mechanism will make the quality adaptation unpopular in the networks which use other mechanisms. So the quality adaptation nearly independent on the underlying congestion controls should be preferred to.

In our quality adaptation system, the networks status is obtained through the information of the receiver buffer. With the group of ΔRb_k as the parameters, the bit rate and the classification of the video stream are determined. Then the modulated stream is sent into the sender buffer in the form of packets. It is the responsibility of transportation protocols to perform the underlying congestion control. The two mechanisms do not change any other information. The relations between them are just that one sending data into the buffer and the other getting it out there for transportation. That means our quality adaptation mechanism is independent of the underlying congestion control, which makes our design more popular with no limitation of matching with some transportation protocols.

C. The Expierence of Fuzzy Control Rules Designing

Fuzzy rules are the soul of the controller, which are designed traditionally following the expert experiences. Here our experiences used to design the rules for the class VQC is based on analysis of the physical situations.

Since the number of consumed video frames is a constant during a fixed period (1 frame for 1/30 sec.) with no relation with its bit rate, the big bit rate will make the receiver buffer occupancy decrease more rapidly, while the occupancy will increase if the consumed frame rate is smaller than that of just arriving ones. In order to achieve the control target, which is keeping the receiver buffer occupancy approach the reference, our experiences can be described as: output a bigger quality index Q which can be used to modulate a bigger frame rate when the actual occupancy is bigger than the reference; on the contrary situation output a smaller Q. When the difference ΔRb is zero, Q will be kept unchanged. If the control rules make Q sensitive to the changes of ΔRb, the effect of short-term improvement will be good; on the contrary a Q insensitive to ΔRb will achieve long-term smoothing effect. So the criteria of designing fuzzy logic rules are to choose an appropriate degree of sensitivity.

Ⅵ. EXPERIMENT AND SIMULATION RESULTS

We implement the quality adaptation system with Visual C++, and choose 300 Hz as the sample frequency. The network environment is a DS network and four differentiated QOS levels are provided ($K = 4$). Fig. 7 shows the assumed available network bandwidth traces of different services, which lasts 1 minute, for transmission of enhancement layers. In order to test the performance of our design when it works under tough networks conditions, we set the bandwidth very stringent and exaggerate the fluctuations intentionally. We set the bandwidth provided to class 4 larger than those of other classes to give the comparison. In our experiments, the available bandwidths are viewed as the server's sending rate.

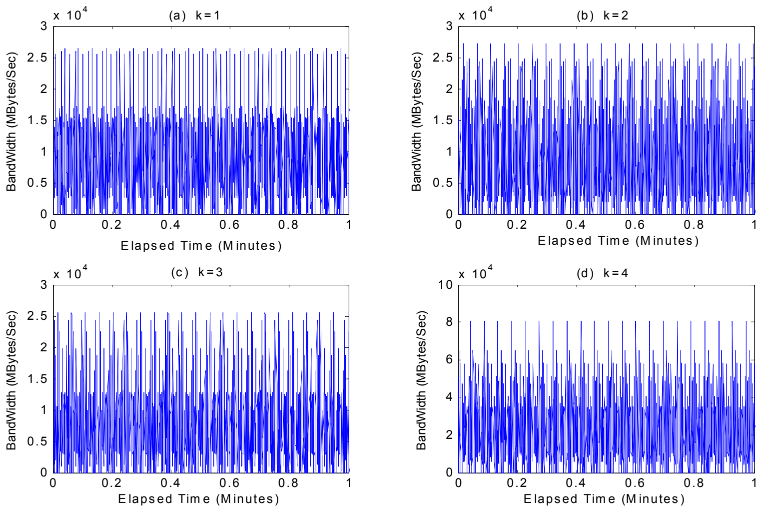

Fig.7. The available bandwidth for differentiated services

In the experiments video foreman_cif is used as test clip. The bandwidths shown in Fig.7 are repeated, lasting about ten minutes. Some variables are initialized as: the receiver buffer reference share for per class $Rbr_k = 0.5M$ Bytes $(k = 1,2,3,4)$, preloaded video streams' size before transmission is $\sum_{k=1}^{4} Rb_k(0) = 2M$ Bytes, and the threshold of the sender buffer occupancy is $\gamma = 0.5M$ Bytes. In the fuzzy logic controller for per class, five fuzzy sets are separately chosen for inputs and output, and triangular memberships are set. The combination of 5 fuzzy sets of ΔRb_k and 5 of $\sum_0^t \Delta Rb_k$ yields 25 rules given based on experience of experts.

The result is given in Fig.8. At the beginning underflow happens for three receiver buffer shares due to the low bandwidths, but about five minutes later they were adjusted to increase gradually. Though they can't approach the reference occupancy, they did not underflow any more provided with the same stringent bandwidths as those at the beginning. While the share for class 4 approached $0.5M$, the reference occupancy, because packets belonging to class 4 are provided with more available bandwidth (see Fig.8 (b)-(e)).

Compared with the fluctuation of bandwidth (see Fig.7), a much more smooth video quality (see Fig.8 (a)) is achieved five minutes later under the quality adaptation control algorithm. The degree of fluctuation of video quality is dependent on the granularity of fuzzy sets classification. If more fuzzy sets are chosen for inputs and output, the fluctuation of video will become smoother.

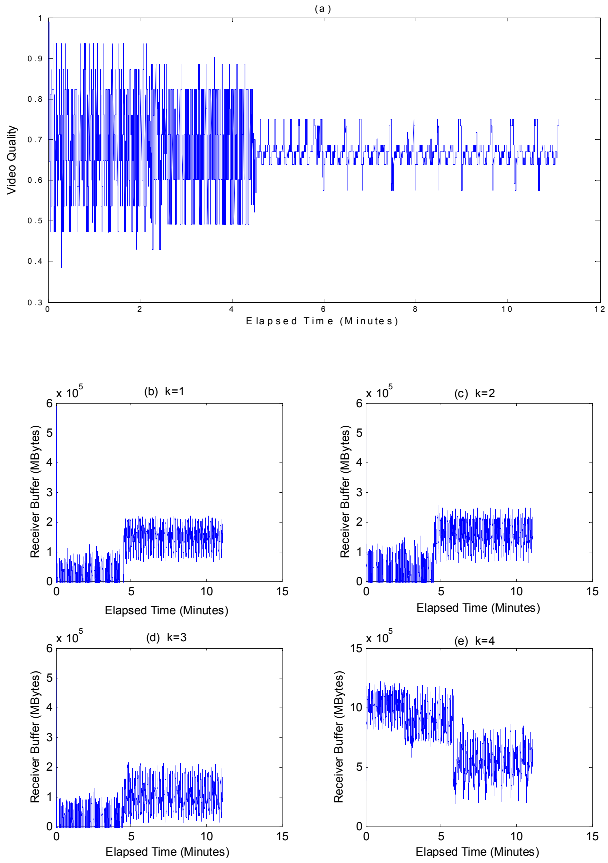

Fig.8.(a) Video quality, (b) Receiver buffer share for class 1st, (c) Receiver buffer share for class 2nd, (d) Receiver buffer share for class 3rd, (e) Receiver buffer share for class 4th.

Ⅶ. CONCLUSION

In this paper we propose a system to implement quality adaptation in the DS networks. The control theory is the guidance of the designing process, so we build a closed-loop model of the system, but it is divided into several loops for different classes aiming at the characteristics of the DS networks. The receiver buffer occupancy is observed to reflect the networks status in real-time. This information is inputted into the VQC as the feedback. VQC, the key block of the loop, is built by using fuzzy logic control technology, and it can perfectly solve the nonlinear and uncertainty problems of the system. It can be proved efficient in the DS, even BE networks.

As an addition, the system can easily be used in multicast applications due to its simplicity. It can be implemented as an agent in the multicast networks to implement quality adaptation.

Acknowledgements

Thanks for the helpful suggestions from the reviewers. Thanks for the edit by Chuoxi Wang and Huafeng Lu.

Reference

[1] D. Wu, Y. T. Hou, and Y.-Q. Zhang, “Scalable video coding and transport over broad-band wireless networks,” *Proc. IEEE,* VOL.89, NO.1, Jan.2001, pp.6-20.

[2] N. Miller and P. Steenkiste, “Collecting network status information for network-aware applications,”

presented at the *IEEE INFOCOM'2000*, Tel Aviv, Israel, Mar. 2000.
[3] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," *RFC2475, IETF,* Dec.1998.
[4] D. Wu, Y.T. Hou, W. Zhu, Y.-Q. Zhang, and J. Peha, "Streaming Video over the Internet: Approaches and Directions," *IEEE Trans. Circuits Sys. Video Technol.,* VOL. 11, NO. 3, Mar. 2001, pp.282-300.
[5] D. Wu, Y. T. Hou, and Y.-Q. Zhang, "Transporting real-time video over the Internet: Challenges and approaches," *Proc. IEEE,* VOL.88, Dec.2000, pp.1855-1875.
[6] S. Floyd and K. Fall, " Promoting the use of end-to-end congestion contrl in the Internet," IEEE Trans. Networking, VOL.7, Aug.1999, pp.458-472.
[7] D. Wu, Y. T. Hou, W. Zhu, H.-J. Lee, T. Chiang, Y.-Q. Zhang, and H. J. Chao, "On end-to-end architecture for transporting MPEG-4 video over the Internet," *IEEE Trans. Circuits Syst. Video Technol.,* Sept.2000, pp.923-941.
[8] J. Walpole, R. Koster, S. Cen, C. Cowan, D. Maier, D. McNamee, C. Pu, D. Steere, and L. Yu, "A player for adaptive MPEG video streaming over the Internet," *in Proceedings of the 26th Applied Imagery Pattern Recognition Workshop AIPR-97, SPIE*, (Washington DC), Oct. 1997.
[9] K. Lee, R. Puri, and T. Kit, "An integrated source coding and congestion control framework for video streaming in the Internet", *IEEE INFOCOM 2000*, Israel, March 26 - 30, 2000. pp.747-756.
[10] R. Rejaie, M. Handley, and D. Estrin, "Layered quality adaptation for Internet video streaming," *IEEE Journal on Selected Areas in Communications,* VOL.18, NO.12, Dec.2000, pp.2530-2543.
[11] B. Girod, "Psychovisual aspects of image communications," *Signal Proc.,* VOL.281, 1992, pp.239-25.
[12] J. Shin, J-G. Kim, J. W. Kim, D. C. Lee and C.-C. J. Kuo, "Aggragated QoS mapping framework for relative service differentiation-aware video streaming," *Packet Video Workshop*, 30 April - 1 May 2001, Kyongju, Korea,
[13] N.G.Duffield, K.K.Ramakrishnan and Amy R.Reibman, "SAVE: An algorithm for smoothed adaptive video over explicit rate networks", networking, IEEE/ACM Transaction on, Vol 6, Issue 6, Dec.1998, pp.717-728
[14] S. Jacobs and A. Eleftheriadis, "Real-time dynamic rate shaping and control for Internet video applications," *Workshop on Multimedia Signal Processing*, June 1997, pp. 23-25.
[15] W. Li, "Fine granularity scalability in MPEG-4 for streaming video," *ISCAS 2000-IEEE international Symposium on Circuits and Systems*, May 28-31, 2000, Geneva, Switzerland.
[16] N. Feamster, D. Bansal, and H. Balakrishnan, "On the interactions between layered quality adaptation and congestion control for streaming video," *http://nms.lcs.mit.edu/projects/videocom/*.
[17] D. Ye, X. Wang and Q.Wu, "A novel buffer-driven adaptive streaming approach for stored video", in draft.