Journal of Zhejiang University SCIENCE A
ISSN 1009-3095 (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Streaming and congestion control using scalable video coding based on H.264/AVC

NGUYEN Dieu Thanh†, OSTERMANN Joern
(*Institute of Information Technology, University of Hannover, Hannover 30167, Germany*)
†E-mail: nguyen@tnt.uni-hannover.de

Received Dec. 10, 2005; revision accepted Feb. 26, 2006

Abstract: This paper presents a streaming system using scalable video coding based on H.264/AVC. The system provides a congestion control algorithm supported by channel bandwidth estimation of the client. It uses retransmission only for packets of the base layer to disburden the congested network. The bandwidth estimation allows for adjusting the transmission rate quickly to the current available bandwidth of the network. Compared to binomial congestion control, the proposed system allows for shorter start-up times and data rate adaptation. The paper describes the components of this streaming system and the results of experiments showing that the proposed approach works effectively for streaming video.

Key words: Scalable video coding, Congestion control, Bandwidth estimation, Transport protocols, Retransmission
doi:10.1631/jzus.2006.A0749 **Document code:** A **CLC number:** TN919.8

INTRODUCTION

Streaming of multimedia data over the Internet has rapidly increased in recent years. All commercial applications and most research in video streaming use conventional hybrid video coding. To adapt the data transmission rate on the server to the varying bandwidth caused by congestion in the Internet or to different available bandwidths of different clients, the simulcast solution is widely applied (Balk *et al*., 2003; Feamster *et al*., 2001; Schierl and Wiegand, 2004). A large number of available bit streams or real-time trans-coding is required on the server side. This problem can be solved by using scalable video coding. Scalable video coding is not only a convenient solution to adapt the data rate to varying bandwidth in the Internet but also a most promising solution for multicast congestion control (Perkins, 2003).

Due to its advantages for transmission scalable video coding has attracted attention recently. In January 2005, the ISO/IEC Moving Pictures Experts Group (MPEG) and the Video Coding Experts Group (VCEG) of the ITU-T started jointly MPEG's Scalable Video Coding (SVC) project as an Amendment of H.264/AVC standard. The scalable extension of H.264/AVC was selected as the first Working Draft (Reichel *et al*., 2005a). Furthermore, the Audio/Video Transport (AVT) Working Group of the Internet Engineering Task Force (IETF) started in November 2005 to draft the RTP payload format for the scalable extension of H.264/AVC and the signaling for layered coding structures (Wenger and Wang, 2005).

In this paper, we present the first real-time streaming system using the scalable video coding based on JSVM-3 (Reichel *et al*., 2005b). Our work focuses on a congestion control algorithm that plays an important role in streaming applications over a best-effort packet-switched environment like the Internet. For streaming applications, UDP is used as transport protocol. First, the latency which can be introduced by retransmissions when using TCP is not suitable for video streaming. Second, video streams are loss tolerant to some extent. Since UDP does not provide congestion control, the application layer must provide this function. Feamster *et al*.(2001) & Schierl

and Wiegand (2004) used TCP-friendly binomial congestion control algorithms. This algorithm family is based on a congestion window w_t which is the amount of bytes or the number of packets sent at the time t with following adjustment policy (Bansal and Balakrishnan, 2001):

$$I: w_{t+R} = w_t + \alpha / w_t^k,\ \alpha > 0,$$
$$D: w_{t+\delta t} = w_t - \beta w_t^l,\ 0 < \beta < 1,$$

where α, β, k, l are constants, I and D stand for increase and decrease, respectively. The window w will be increased, if the acknowledgements of a window are received in a round-trip-time R. If the server detects packet loss at the time (t+δt), the window w will be decreased.

Binomial congestion control algorithms have two disadvantages. Firstly a binomial session begins with slow-start state (Yang and Lam, 2000). If the session begins with the base layer of the scalable bit stream and at the same time the available bandwidth is enough to transmit up to the highest enhancement layer, the session must switch through many layers and produces an unpleasant subjective effect on the client displays. Secondly a binomial congestion control session increases and decreases the congestion window with a fixed proportion of the last window size. It does not always match the available bandwidth in the network.

As an alternative, in this paper we propose a new congestion control method supported by a channel bandwidth estimation for scalable video coding. We apply the Receiver-Based Packet Pair (RBPP) method (Paxson, 1997) for bandwidth estimation but in a way that no extra probing packets during the streaming session are required, and thus the network is not loaded additionally. Moreover we provide a retransmission mechanism only for the base layer of scalable bit stream, and so to limit the load of just congested network which causes this retransmission.

The rest of the paper is organized as follows. In Section 2 the scalable video streaming system with congestion control and channel bandwidth estimation is presented. Section 3 provides experimental results of the developed system. Section 4 concludes the paper.

STREAMING SYSTEM USING SCALABILITY EXTENSION OF H.264/AVC

The overview of the streaming system using H.264/AVC scalable video coding is shown in Fig.1. On server and client side we use the RTP and RTCP protocols for transport of video data and feedback on transport quality (Schulzrinne *et al*., 2003) and the RTSP protocol (Schulzrinne *et al*., 1998) for establishment and control of media streams.

The client sends the acknowledgments for packets of the baser layer, its buffer state and the estimated bandwidth to the server. With this information the congestion control function will decide up to which layer of the scalable bit stream data will be sent to the client. To extract these layers from the bit stream a bit extractor is used.

Scalable video coding and bitstream extractor

The scalable video coder employs different techniques to enable spatial, temporal and quality scalability (Reichel *et al*., 2005b). Spatial scalability is achieved by using a down-sampling filter that generates the lower resolution signal for each spatial layer. Either motion compensated temporal filtering (MCTF) or hierarchical B pictures obtain temporal decomposition in each spatial layer that allows temporal scalability. Both methods process input pictures at the encoder and the bit stream at the decoder in group of pictures (GOP) mode. A GOP includes at least one key picture and all other pictures between this key picture and the previous key picture, whereas

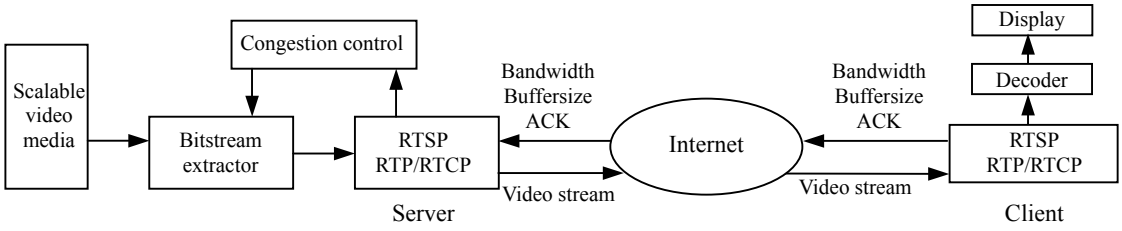

Fig.1 Overview of the streaming system using scalable video coding based on H.264/AVC

a key picture is intra-coded or inter-coded by using motion compensated prediction from previous key pictures. To remove redundancy within spatial layers, motion and texture information of the temporal level in the lower spatial layer are scaled and refined for prediction of motion and texture information in the current layer.

For each temporal level, the residual signal resulting from texture prediction is transformed. For quality scalability, the transform coefficients are coded by using a progressive refinement mode to create a quality base layer and several quality enhancement layers. This approach is called fine grain scalability (FGS). The advantage of this approach is that the data of a quality enhancement layer (FGS layer) can be truncated at any arbitrary point to limit data rate and quality without impact on the decoding process.

Fig.2 shows the data rate allocation for each spatial-temporal resolution with two additional FGS layers of a typical H.264/AVC scalable video bit stream. The lowest spatial layer (layer 0) has QCIF resolution and four temporal levels at 1.875, 3.75, 7.5 and 15 Hz, respectively. The higher spatial layer (layer 1) has CIF resolution and five temporal levels that give the additional maximal frame rate of 30 Hz. The bar for data rate of each spatial-temporal resolution is divided into three blocks, where the bottom presents the data rate of the quality base layer for this spatial-temporal resolution, the middle the data rate of the first FGS layer and the top the data rate of the second FGS layer. Note that the data rate of the quality base layer of the spatial-temporal resolution (S_n,T_m) includes the quality base layer data rate of all spatial layers from 0th layer (S_0) to nth layer (S_n) for the mth temporal level T_m. Fig.2 shows that an arbitrary data rate between 70 and 950 kbps can be adapted by using this bit stream.

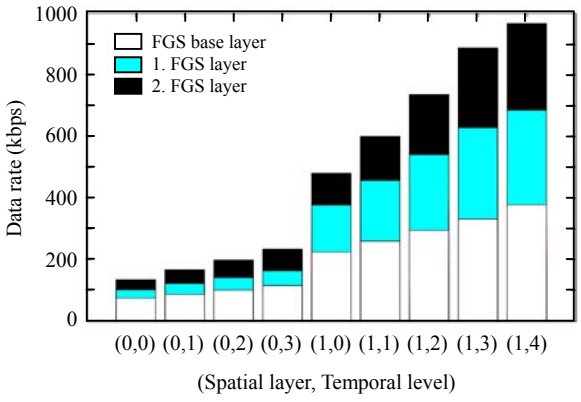

Fig.2 Data rate allocation for spatial-temporal resolution (S_n,T_m) with two additional FGS layers of a typical H.264/AVC scalable video bit stream, where S_n is nth spatial layer and T_m is mth temporal level

The Network Abstraction Layer (NAL) arranges the data from the afore described coding layer in NAL units. For each frame of a given spatial-temporal resolution its data is divided into one NAL unit for the quality base layer and one NAL unit for each quality enhancement layer. For a given data rate the bit stream extractor must only discard NAL units of high spatial-temporal resolution and truncate NAL units of quality enhancement layers.

We use the reference software JSVM-3, in which only the first picture is an IDR picture and other key pictures are intra- or inter-coded, so called I- or P-pictures. An IDR picture is intra-coded and all of its previous pictures cannot be used as reference frame for prediction of its following pictures. So the spatial layer switching is only possible when an IDR picture occurs. For this purpose, we extend the JSVM-3 software by coding key pictures as IDR pictures.

Channel bandwidth estimation

For channel bandwidth estimation we use the method called Receiver-Based Packet Pair (RBPP) (Paxson, 1997). This method is based on the principle that if two or more packets leave the sender with spacing smaller than the transmission delay of the packets over the bottleneck link, this spacing will be expanded by the bottleneck. The receiver can use the expanded time spacing to estimate the speed of the bottleneck link which corresponds to the available bandwidth between the sender and the receiver. The available bandwidth B_{est} is computed as follows:

$$B_{\text{est}} = \sum_{i=1}^{N} d_i \Big/ \sum_{i=0}^{N-1} \tau_{i,i+1},$$

where N is the number of packets sent back-to-back, d_i is the length of the ith packet and $\tau_{i,j+1}$ the inter-arrival time between the ith packet and the (i+1)th packet.

For our system we use probing packets sent from server to client only at the beginning of the session to estimate the available channel bandwidth. During the session we utilize RTP packets for this

purpose. Since every GOP includes at least one key picture that in general is divided into more than one RTP packet, the RTP packets of the key picture can be sent back-to-back. The first RTP packet of the next GOP is sent after the time needed to display the current GOP. Thus the server does not cause buffer overflow on the client. The last packet of a GOP is annotated by setting the marker bit in RTP header, so the receiver can notice the end of back-to-back packets. These packets are used to estimate the available bandwidth according to the RBPP method.

We implemented the optional RTSP method called SET_PARAMETER to transmit the estimated bandwidth from client to server.

Feedbacks and packet retransmission

The channel bandwidth estimation using the RBPP method has certain limitations and can fail, if there is an high number of outliers in the measurement of the inter-arrival time. Furthermore, the server learns the estimated bandwidth during a round-trip time later than it actually happens. To avoid this problem we provide the extended acknowledgement of the buffer size and the arrival of important packets in addition to the RTCP information (Schulzrinne *et al*., 2003).

As a result of the scalable video coding structure a GOP cannot be reconstructed, if packets of the quality base layer of the lowest temporal level and of all sending spatial layers, so-called non-discardable packets, are not received by the client. On the other hand, if quality enhancement layers, higher temporal layers or higher spatial layers are lost, the scalable video decoder can still reconstruct a GOP with lower spatial-temporal resolution and/or lower quality. In this respect, a scalable bit stream provides error resilience. Hence, the feedbacks have to be sent only for non-discardable packets. The server will resend a non-discardable packet if its feedback is not received after a time-out.

For the investigations we modified the RTCP transport layer feedback message in the extended RTP profile (Ott *et al*., 2004) to generate feedbacks for non-discardable packets. This message includes a number of 32-bit information fields, each of which consists of a 16-bit sequence number of the first non-discardable packet for a spatial layer and a 16-bit bit mask of the following non-discardable packets. That means if a following non-discardable packet is lost, its bit is set to zero. This message is sent after receiving all RTP packets of a key picture.

Furthermore the client informs the server about its buffer size at the beginning of a session or if the size has changed by sending an RTSP command. Within the session the information about the client buffer usage is sent together with the estimated bandwidth value via RTSP, but only if a buffer overflow or underflow is imminent.

Congestion control

To adapt bandwidth variation to resist packet loss and to alleviate the problem of a client buffer usage, a congestion control is required on the server. We present in this paper a congestion control algorithm which is mainly based on the estimated channel bandwidth and takes the packet loss and client buffer usage into consideration additionally.

Given that B_{est} is the newly estimated bandwidth, $R_{i,j}$ the data rate of spatial-temporal resolution (S_i,T_j) and spatial-temporal resolution (S_n,T_m) with data rate $R_{n,m}$ is currently used, the spatialtemporal resolution (S',T') with data rate R' will be chosen according to the following criteria:

If $B_{\text{est}}>R_{n,m}$:

$$(S',T')=\begin{cases}(S_n,T_j), & \text{if } B_{\text{est}}<R_{n+1,m-2},\\ (S_{n+1},T_j), & \text{otherwise,}\end{cases} \tag{1}$$

with $m-2\le j\le m+2$ and $R'<B_{\text{est}}$.

If $B_{\text{est}}\le R_{n,m}$:

$$(S',T')=\begin{cases}(S_n,T_j), & \text{if } B_{\text{est}}>R_{n-1,0},\\ (S_{n-i},T_j), & \text{otherwise,}\end{cases} \tag{2}$$

with $0\le j\le m+2$, $\min\limits_{0\le i\le n}(B_{\text{est}}-R')$.

The quality enhancement layers will be truncated to fit the rest of the available bandwidth $\Delta R=B_{\text{est}}-R'$.

These criteria adapt the sending data rate to the estimated bandwidth also considering the visual quality at the client. Especially when the spatial layer can be increased as a result of a higher estimated bandwidth, the new temporal level should not be much higher or lower than the last temporal level. If the estimated bandwidth decreases, the spatial-

temporal resolution should be adapted strictly to avoid packet loss.

If one or more non-discardable packets are lost because of bandwidth estimation fault or fast bandwidth variation, then the following rules are employed:

(1) If $B_{est}>R_{n,m}$: $(S',T')=(S_n,T_m)$ and $\Delta R=0$;

(2) If $B_{est}\leq R_{n,m}$: Eq.(2) is applied and $\Delta R=0$.

We define p as the client buffer usage with $0\leq p\leq 1$. Then in the case of p being sent together with B_{est}, the server adapts its sending behaviour as follows:

(1) If client buffer is at risk of overflow ($1-p<<p$), then $(S',T')=(S_{n-1},T_m)$ and $\Delta R=0$.

(2) If client buffer is at risk of underflow ($p<<1-p$), then the packets are sent successively until the client buffer usage is not close to underflow anymore.

EXPERIMENTAL RESULTS

To verify our streaming system we use the simulation scenario shown in Fig.3. Our streaming server and client run on two hosts A and C which are connected to each other via host B. On host B we install the software package ns-2 (Network Simulator ns-2) for network emulation. The packets sent from server and client are captured into our emulation network with two routers $R1$ and $R2$. The link between these routers has a capacity of 1500 kbps representing the bottleneck link in our network. To emulate the bandwidth fluctuation and packet loss we attach a UDP application and a telnet application over the bottleneck link as competitors of our streaming session.

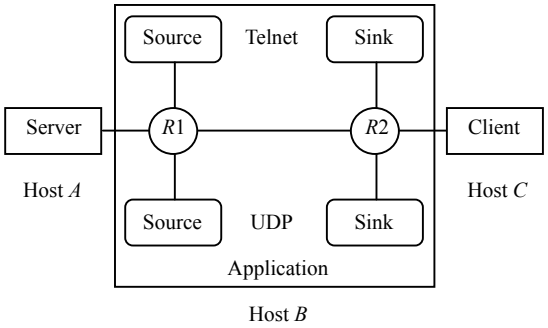

Fig.3 Network emulation scenario with ns-2

At beginning of the session we have the bottleneck link for ourselves. After 2 s the UDP application is started at a constant bit rate of 1400 kbps for 20 s. The telnet application is started from 21 s for 80 s and sends its packets at a data rate of 1000 kbps. That means the total network congestion occurs on the bottleneck link from 2 s for 20 s. The two routers $R1$ and $R2$ have typical drop-tail queues that will reject the incoming packets in case of overflow leading to packet loss. For our streaming session we use the scalable bit-stream at data rate distribution depicted in Fig.2. This bit-stream includes 1200 frames from sequences Mobile & Calendar, Foreman, Flower, Stefan and Bus with GOP size of 16.

Fig.4 shows the sending data rate in solid line and the estimated bandwidth in dashed line over more than 100 s. Note that we adapt the sending bandwidth for each GOP as result of the scalable video codec mode. The sending data rate on the server is well adapted to the bandwidth variation. Especially despite the bandwidth estimation error of 2 s for 20 s the sending rate is adapted correctly because the packet loss is detected. The number of lost packets in this time frame is 60 packets. These packets are retransmitted if they are not acknowledged after a round trip time of about 1 s in this congestion time frame. That means the client needs a buffer size of at least two GOPs. That is a typical size due to the GOP based mode of scalable codec. Therefore the retransmitted packets can still be received in time.

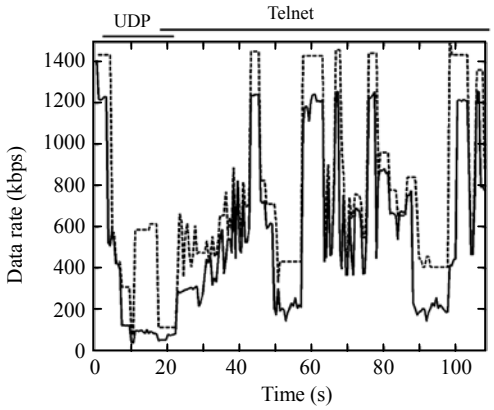

Fig.4 Sending data rate (solid line) vs estimated bandwidth (dashed line)

To evaluate the adaptation behavior of the server after 22 s we must take the visual quality on the client into account. Fig.5 shows the spatial, temporal and FGS layer combination received by the client for each

corresponding GOP. If the combination is between two spatial-temporal resolutions, the FGS enhancement layers are included. The sending data rate and the spatial, temporal and FGS layer combination increases after 22 s. The sending data rate is well adapted to the varying estimated bandwidth and the spatial-temporal resolution remains at the highest layer and level.

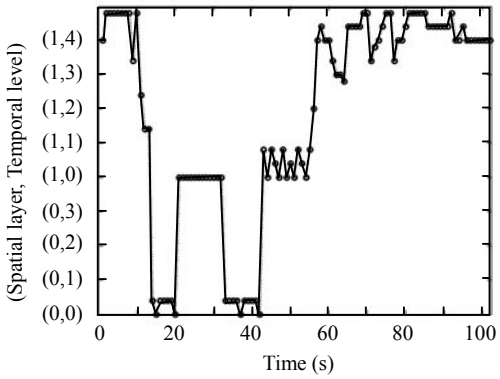

Fig.5 Spatial-temporal resolution received on the client

CONCLUSION

In this paper a real-time streaming system based on H.264/AVC scalable video coding is presented. The client transmits packet acknowledgements and bandwidth estimates to the server. The task of the congestion control on the server is not only to adapt the sending data rate to bandwidth variations but also to optimize the visual quality on the client side by avoiding frequent spatial and temporal resolution changes. Therefore, the congestion control prefers to adapt by using fine grain scalability instead of changing the spatial-temporal resolution, as this adaptation is more comfortable for the viewer. In contrast to the binominal congestion control, our scheme enables a faster start-up time without congesting the network. The experimental result shows that our system adapts the sending data rate rapidly to the available bandwidth. Furthermore, network overload is avoided in case of bandwidth estimation errors. The streaming system works well also in extreme network congestion situations when the competitors are TCP or UDP applications.

References

Balk, A., Maggiorini, D., Gerla, M., Sanadidi, M.Y., 2003. Adaptive MPEG-4 Video Streaming with Bandwidth Estimation. Proceedings of the Second International Workshop on Quality of Service in Multiservice IP Networks, **2601**:525-538.

Bansal, D., Balakrishnan, H., 2001. Binomial Congestion Control. IEEE INFOCOM.

Feamster, N., Bansal, D., Balakrishnan, H., 2001. On the Interactions Between Layered Quality Adaptation and Congestion Control for Video Streaming. 11th International Packet Video Workshop. Kyongju, Korea.

Ott, J., Wenger, S., Sato, N., Burmeister, C., Rey, J., 2004. Extended RTP Profile for RTCP-based Feedback (RTP/AVPF). Internet Engineering Task Force, Internet Draft, draft-ietf-avt-rtcp-feedback-11.txt.

Paxson, V., 1997. Measurements and Analysis of End-to-End Internet Dynamics. Ph.D Dissertation, Computer Science Department, University of California at Berkeley.

Perkins, C., 2003. RTP Audio and Video for the Internet. Addison-Wesley.

Reichel, J., Schwarz, H., Wien, M., 2005a. Scalable Video Coding—Working Draft I. Joint Video Team of ITU-T VCEG and ISO/IEC MPEG, Doc. JVT-N020.

Reichel, J., Schwarz, H., Wien, M., 2005b. Joint Scalable Video Model JSVM-3. Joint Video Team of ITU-T VCEG and ISO/IEC MPEG, Doc. JVT-P202.

Schierl, T., Wiegand, T., 2004. H.264/AVC Rate Adaption forInternet Streaming. 14th International Packet Video Workshop. Irvine.

Schulzrinne, H., Rao, A., Lanphierand, R., Jacobson, V., 1998. Real Time Streaming Protocol (RTSP). Internet Engineering Task Force, RFC 2326.

Schulzrinne, H., Casner, S., Fredrick, R., Jacobson, V., 2003. RTP: A Transport Protocol for Real-Time Applications. Internet Engineering Task Force, RFC 3550.

Wenger, S., Wang, Y.K., 2005. RTP Payload Format for SVC Video. Internet Engineering Task Force, Internet Draft, draft-wenger-avt-rtp-svc-00.txt.

Yang, R.Y., Lam, S.S., 2000. Analysis of Binomial Congestion Control. Technical Report TR-00-14, Department of Computer Sciences, The University of Texas at Austin.