実時間動画像マルチキャストのための
フィルタリング手法の実装と評価

アカミネ エクトル　中田 和久　若宮 直紀　村田 正幸　宮原 秀夫

大阪大学大学院基礎工学研究科
〒 560–8531 大阪府豊中市待兼山町 1–3
Tel: 06–6850–6588　Fax: 06–6850–6589
{akamine,k-nakada,wakamiya,murata,miyahara}@ics.es.osaka-u.ac.jp

あらまし

サーバから複数のユーザに対して同時にマルチメディアデータを配送するマルチキャスト通信においては，ユーザごとに要求する通信品質が異なるという問題がある．高機能で柔軟性の高いネットワークの構築を可能にするアクティブネットワーク技術は，ユーザからの様々な要求を効率的に満たすことのできるマルチメディア通信を実現するものとして注目されている．アクティブネットワークにおける動画像マルチキャストでは，ネットワーク上の適切な位置に配置されたアクティブノードにおいて，マルチメディアデータの品質調整を実時間で行うことにより，下流のユーザの要求品質を満足することができる．本報告では MPEG-2 アルゴリズムを用いて符号化された動画像のための品質調整手法であるフレーム棄却フィルタ，ローパスフィルタ，再量子化フィルタを対象に，アクティブネットワークにおける実時間動画像マルチキャストへの適用を検討する．それぞれのフィルタリング手法について，ユーザやネットワークの指定した目標レートに応じて適切に動画像品質を調整するアルゴリズムを提案し，実証実験によりいずれの手法においても所望のレートを達成できるが，より高い動画像品質を得るにはローパスフィルタを用いることが有効であることを示した．

キーワード　動画像マルチキャスト，ビデオフィルタリング，アクティブネットワーク

Implementation and Evaluation of Video Filtering Mechanisms
for Real-Time Multicast

**Héctor Akamine　Kazuhisa Nakada　Naoki Wakamiya　Masayuki Murata
Hideo Miyahara**

Graduate School of Engineering Science
Osaka University
Toyonaka, Osaka 560–8531, Japan
{akamine,k-nakada,wakamiya,murata,miyahara}@ics.es.osaka-u.ac.jp

Abstract

QoS requirements in video multicast services are diverse due to the heterogeneity of the network and clients, and the difference in users' preferences on the requested video quality. In previous work, we have proposed a multicast tree construction algorithm, considering an active network in which active nodes can filter or transcode the video stream, which assists the server to provide various qualities of a video stream to satisfy different quality requirements. In this paper, we implemented and evaluated three filter mechanisms, i.e., frame dropping, low-pass and requantization filters for MPEG-2 video streams, intended to realize heterogeneous video multicasting in active networks. We propose heuristic algorithms for those filters to appropriately adjust an incoming video stream to the specified output rate. Through experimentation with MPEG-2 video streams, in which we observe the rate and quality variation of filtered streams, we show that the proposed algorithms are practical and effective in regulating video rate without degrading in excess the perceived video quality.

Keywords　heterogeneous video multicast, video filtering, active network

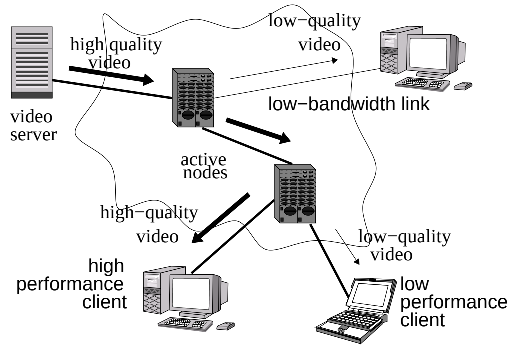

Figure 1: active nodes and heterogeneous multicast

1 Introduction

The use of video transmission over the Internet becomes ubiquitous as the popularity of multimedia communications increases, fueled by the improvements in computer hardware and software, and the increase in network bandwidth. Real-time video applications are demanding applications for the network, because they usually need strict conditions of bandwidth, delay and jitter.

Multicast is intended to reduce the load of the network in one-to-many sessions. With multicast, we avoid the need for the server to send separate data streams to each requesting client. The server sends a single stream to a specified multicast address, and the network efficiently distributes it to all the clients. Nevertheless, it is likely that all members of the same session can not receive the same volume of data. In a video session, clients behind a slow connection will not be able to get a stream with the same quality as one that has access to a faster link. In addition, it is possible that the client can not play out high quality video due to limited CPU capacity, even if the network can afford to transfer the stream. Multicast alone is not able to deal with this heterogeneity, and some other mechanisms are required to provide "individualized" streams without recurring to the inefficient unicast.

Recently, the research community is investigating a new paradigm, called *active networking* [1], which introduces programmability into the network and offers new services and flexibility that conventional networks can not.

In [2] we proposed a framework for heterogeneous multicast using active networking, in which some properly located active nodes filter the video stream coming from the server or other active nodes to satisfy different quality requests, as depicted in Figure 1. Through simulation experiments, we showed that the proposed framework and tree construction algorithm achieves efficient use of the network bandwidth. We assumed that active nodes can appropriately adjust the rate of an incoming video stream on demand, but the implementation of filtering mechanisms was not covered.

In this paper, we discuss some video filtering mechanisms that can be applied on MPEG-2 [3] video streams to realize the active heterogeneous multicast. We implemented and evaluated three types of filters, and propose simple rate control algorithms. Previous work in filtering [4, 5, 6] does not make a proper comparison between different filter types, and does not cover the relationship between the perceived quality and rate reduction. We study these issues here, and also discuss briefly the possibility of being implemented in an active node.

The rest of this paper is divided as follows: in Section 2 we explain the mechanism of the implemented filters. In Section 3 we propose rate control algorithms to achieve a desired target rate. In Section 4 we evaluate and compare the filters. We draw our conclusions and point to further research topics in Section 5.

2 Filtering Mechanisms for MPEG Video Streams

The filter operates within the network or at the network edge to adjust the characteristics of a continuous media stream to the requirements of distinct receivers by intentionally degrading the quality of the stream and reducing its rate. To achieve compression, the filters partially decode, alter and re-encode the video data in real-time.

We consider MPEG-2 elementary video streams in this paper. MPEG encoding/decoding consists of a series of steps, as shown in Figure 2. The three types of filters we implemented and evaluated, frame discarding, low-pass and requantization filters [4], are also shown.

Frame discarding filter. An MPEG video stream is composed of several *Group of Pictures* (GoPs), which consist of a sequence of three types of pictures: I, P and B. I pictures can be independently decoded, while P pictures depend on the preceding I or P picture in the sequence, and B pictures depend on both preceding and following I or P pictures. This filter discards entire pictures to achieve rate reduction, avoiding the need for decoding and re-encoding. Pictures must be discarded taking into consideration the picture dependencies.

Low-pass filter. Like a television signal, each picture in MPEG consists of one luminance and two chrominance components. Each of the components is divided into *blocks* of 8x8 pixels. Four luminance blocks and additional chrominance blocks (their number depending on the chrominance format, typically two) are grouped in a *macroblock*. Blocks in I pictures and some blocks in P and B pictures are independently coded and called *intra blocks*. The others are coded using motion compensation techniques. An intra block is coded using information only from itself by converting pixel data to the frequency domain using DCT (discrete cosine

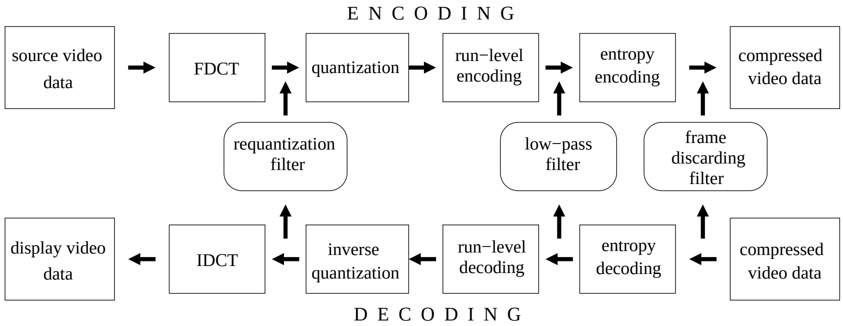

Figure 2: MPEG encoding/decoding steps and filter types

transform). The resulting DCT coefficients are quantized by dividing them by a value called *quantizer scale*, and run-level encoding is further applied to the quantized coefficients. The low-pass filter achieves rate reduction eliminating some of the quantized DCT coefficients starting from the high frequency ones, because they are less influential on the picture quality.

Requantization filter. The requantization filter applies larger quantization scale to reduce the video rate. Both low-pass and requantization filters decreases the picture quality, but the latter requires more processing capability that the former because it involves more re-coding steps.

3 Rate Control Algorithm

In this section, we propose rate control algorithms for the filtering mechanisms described previously. Rate control is performed GoP by GoP to adjust the *average* rate to the specified target rate. Thus, we should note here that our algorithms are not intended to convert VBR streams to CBR.

3.1 Prediction of GoP and Picture Sizes

The rate control algorithm requires to know the size of the GoP and pictures to modify the parameters appropriately. Filtering must be done in "one pass" to ensure real-time operation. Due to this condition, it is not possible to know the size values in advance, and we use predictors instead.

For the sake of simplicity in the implementation, we use exponential moving average to predict the size of the i-th GoP, G_i as:

$$G_i = \alpha G_{i-1} + \beta g_{i-1} \quad i \geq 2 \tag{1}$$

where g_{i-1} (bits) is the measured size of the $(i-1)$-th GoP and G_{i-1} (bits) is the predicted value for the $(i-1)$-th GoP. $G_1 = g_0$. We examined two predictors: predictor 1, with $\alpha = 7/8$ and $\beta = 1/8$, and predictor 2, with $\alpha = 1/8$ and $\beta = 7/8$. Predictor 1 puts more importance on the previously predicted value, and the other on the size of the previous GoP.

In the case of the frame discarding filter, we also need to maintain predictors for the picture sizes:

$$I_i = \alpha I_{i-1} + \beta i_{i-1} \ , \ i \geq 2 \tag{2}$$

$$P_i = \alpha P_{i-1} + \beta p_{i-1} \ , \ i \geq 2 \tag{3}$$

$$B_i = \alpha B_{i-1} + \beta b_{i-1} \ , \ i \geq 2 \tag{4}$$

where I_i, P_i, and B_i (bits) are respectively the predicted sizes for the i-th I, P and B pictures, i_{i-1}, p_{i-1}, and b_{i-1} are the measured values for the $(i-1)$ pictures (bits). Analogously to the GoP size predictor $I_1 = i_0$, $P_1 = p_0$, and $B_1 = b_0$. We also have predictor 1, with $\alpha = 7/8$ and $\beta = 1/8$, and predictor 2, with $\alpha = 1/8$ and $\beta = 7/8$.

Since the header accounts for an important fraction of the video stream, we also consider a predictor for the total bits used by header data in the i-th GoP, H_i as:

$$H_i = h_{i-1} \ , \ i \geq 1 \tag{5}$$

where h_{i-1} is the measured header size of the $(i-1)$-th GoP.

3.2 Rate Adjustment in the Frame Discarding Filter

First, given a specified target rate R (bits), we calculate the number of bits T_i allowed to the current GoP. We introduce an adjustment value a_i derived from the filtering of previous GoPs:

$$T_i = \frac{R \times N}{F} - a_i - H_i \tag{6}$$

where N stands for the number of pictures in a GoP and F for the frame rate (fps).

Then we calculate the GoP compression ratio r_{Gi}:

$$r_{Gi} = \frac{T_i}{G_i - H_i} \tag{7}$$

We calculate the number of B pictures to discard to achieve the desired rate reduction:

$$Bdrop_i = \min(N - \frac{N}{M}, \lceil \frac{(G_i - H_i) - T_i}{B_i} \rceil) \tag{8}$$

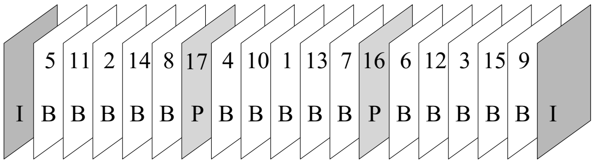

Figure 3: frame discarding order example

If we still have a higher than desired rate after discarding all the B pictures, we discard P pictures:

$$Pdrop_i = min(\frac{N}{M} - 1, \lceil \frac{(G_i - B_i \times Bdrop_i - H_i) - T_i}{P_i} \rceil) \quad (9)$$

We don't discard I pictures to assure smooth reproduction of the stream.

Once we know the number of B (and optionally P) pictures to discard, we use an algorithm to decide which pictures to discard. We do not explain it here for limitations in the space (refer to [7] for more details). The algorithm discards pictures considering symmetry, to assure a smooth reproduction. An example is shown in Figure 3.

Because of the errors introduced with the prediction of GoP and picture sizes, a filtered GoP sometimes becomes smaller or larger than the target size. To make the average rate of the filtered stream coincide with the desired target rate, we introduce an adjustment value a_i in equation 6, that is calculated as:

$$a_i = \sum_{k=\max(0,i-5)}^{i-1} \frac{T_i - f_i}{5} \quad (10)$$

where f_i is the size of the filtered i-th GoP.

3.3 Rate Adjustment in the Low-Pass Filter

We define a value called *low-pass parameter* for each picture type (I, P and B), which stands for the number of DCT coefficients to be left in a block.

We first use equations (6) and (7) to find r_{Gi} from the desired target rate. We set the compression ratio for all the picture types to this value ($r_I = r_P = r_B = r_{Gi}$), and then use the following equations to set the initial value of the low-pass parameter (for the first intra macroblock in the GoP):

$$l_I(r_I) = \lfloor -6.17329 + 59.7498 r_I - 112.427 r_I^2 + 111.905 r_I^3 \rfloor \quad (11)$$
$$l_P(r_P) = \lfloor -11.8626 + 85.5488 r_P - 159.667 r_P^2 + 139.499 r_P^3 \rfloor \quad (12)$$
$$l_B(r_B) = \lfloor -71.9536 + 360.75 r_B - 590.353 r_B^2 + 353.265 r_B^3 \rfloor \quad (13)$$

where l_I, l_p, l_b and r_I, r_p, r_b represent the low-pass parameters and the desired compression ratios for I, P and B pictures respectively. These equations were obtained investigating the relationship among the low-pass parameter and resultant picture size, and give an approximation of the low-pass parameter that can produce the desired compression. However, they are not accurate enough and are only used to derive initial values. For example, for $r_{Gi} = 0.5$, we get $l_I = 9$, $l_P = 8$ and $l_B = 4$.

For each of the following intra macroblocks, the low-pass parameters are changed dynamically using the following formula to manage prediction errors:

$$l_j = \begin{cases} l_{j-1} + 1 & , \quad t_{MBj-1} - f_{MBj-1} > 0 \\ l_{j-1} - 1 & , \quad t_{MBj-1} - f_{MBj-1} < 0 \\ l_{j-1} & , \quad t_{MBj-1} - f_{MBj-1} = 0 \end{cases} \quad (14)$$

where l_j is the low-pass parameter (for I, P or B pictures) to apply to the j-th macroblock, t_{MBj} is the target macroblock size calculated as the product of the macroblock size before filtering o_{MBj} and the picture compression ratio r_{Gi}, and f_{MBj} is the size of the filtered macroblock.

When filter processing for all the pictures of a GoP is finished, we calculate the adjustment value a_i for the next GoP, in the same way as the case of the frame discarding filter.

3.4 Rate Adjustment in Requantization Filter

The process is analogous to the low-pass filter. We define a *requantization parameter*, the value to be added to the quantization scale to achieve compression, which is updated for each macroblock. As with the low-pass filter, we derived some equations that define the relationship between the compression achieved and the requantization parameter to set the initial value of the requantization parameter.

$$q_I(r_I) = \lfloor -2.48863 + \frac{3.16293}{r_I - 0.1} + \frac{0.0125172}{(r_I - 0.1)^2} \rfloor \quad (15)$$
$$q_P(r_P) = \lfloor -8.04467 + \frac{3.00808}{r_P - 0.13} + \frac{0.0413831}{(r_P - 0.13)^2} \rfloor \quad (16)$$
$$q_B(r_B) = \lfloor -2.08088 + \frac{1.16466}{r_B - 0.35} \rfloor \quad (17)$$

where q_I, q_P, q_B and r_I, r_P, r_B are the requantization parameters and the compression ratios for I, P and B pictures, respectively. Using the equations (15) through (17) we find the initial values for the quantization parameters given a compression ratio, and then modify the values for each macroblock.

In the same way as the low-pass filter, we calculate r_{Gi} for the i-th GoP and then we set $r_I = r_P = r_B = r_{Gi}$. For example, for $r_{Gi} = 0.5$, we get 5, 0 and 5 for I, P and B pictures respectively.

The requantization parameter is varied for each of the following macroblocks using the following equation:

$$q_j = \begin{cases} q_{j-1} - 1 & , \quad t_{MBj-1} - f_{MBj-1} > 0 \\ q_{j-1} + 1 & , \quad t_{MBj-1} - f_{MBj-1} < 0 \\ q_{j-1} & , \quad t_{MBj-1} - f_{MBj-1} = 0 \end{cases} \quad (18)$$

where q_j is the requantization parameter to apply to the j-th macroblock.

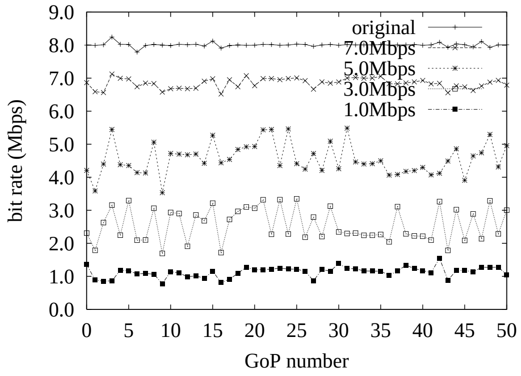

Figure 4: bit rate for frame discarding filter

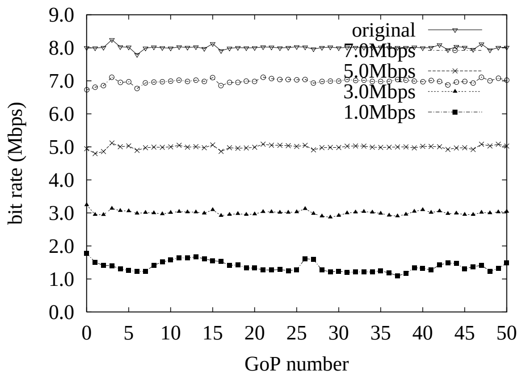

Figure 5: bit rate for low-pass filter

Finally, when processing for all the pictures of a GoP is finished, we calculate the adjustment value a_i for the next GoP.

4 Evaluation of Filtering Mechanisms

We implemented and evaluated the proposed mechanisms with several different CBR and VBR video streams. All the results are not included on account of limited space. The graphs show the results for an 8 Mbps CBR stream, a karate scene captured from the motion picture "The Matrix". Refer to [7] for the complete results for other streams. Comparisons are made in terms of the resultant rate variation, the subjective and objective video quality, and the required processing time.

4.1 Bit Rate Variation

Figures 4, 5 and 6 show the bit rate variation for the three filter types, using predictor 1. Bit rate is averaged over each GoP. All the three filters have a lower limit in the achievable rate. This is due to the nature of the mechanism used. For example, the frame discarding filter does not discard I pictures to assure a smooth reproduction, and the low-pass and requantization filters can be applied only to intra macroblocks.

The frame discarding filter achieves a "coarse" rate adjustment, because discarding is done in picture units. Therefore, we experience great variation in the bit rate.

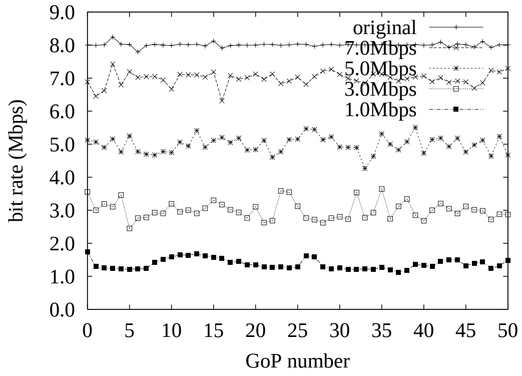

Figure 6: bit rate for requantization filter

Since the low-pass filter dynamically regulates the low-pass parameter in macroblock units instead of pictures, we have less variation compared to the frame discarding filter. The requantization filter also operates at the macroblock level. The output rate is smoother than the frame discarding filter but fluctuates more than the low-pass filter because a small variation in the requantization parameter can produce great difference in the compression ratio.

Although not shown in the figures, we compared predictors and found that predictor 1 needs more time to reach the desired rate when prediction fails but achieves more stable rate variation than predictor 2. Despite the difference, averaged rate over the entire video stream is almost the same.

From the above, we can conclude that the low-pass filter is more effective to achieve the desired bit rate, followed by the requantization filter. Between the two predictors, it would be better to use predictor 1 to produce a smoother rate.

4.2 Video Quality

We use two criteria to compare the quality of the filtered video stream: one based in objective measure using the PSNR (Peak Signal-to-Noise Ratio), and subjective measure using MOS (Mean Opinion Score).

4.2.1 PSNR Evaluation

PSNR is a per-picture, per-pixel measure of the fidelity of the displayed video against the original. PSNR evaluation cannot be applied to the frame discarding filter which only discards or holds the entire picture. A value of about 40 dB means that there is no noticeable difference between the original and the measured picture, whereas with 30 dB degradation is noticeable, and the degradation is very high and unacceptable with 20 dB.

Figure 7 summarizes the results of the measured average PSNR for low-pass and requantization filters at different target rates. For comparison purposes. we also show results of a TM5 encoder [8] which produces CBR streams. As shown in the figure, the low-pass

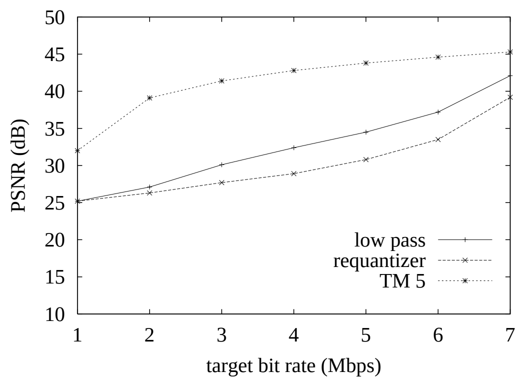

Figure 7: PSNR comparison

Table 1: MOS evaluation

matrix(CBR,8Mb/s)	target rate			
	7Mb/s	5Mb/s	3Mb/s	1Mb/s
frame discard	5.000	3.250	1.750	1.000
low-pass	4.083	3.000	2.333	1.000
requantization	4.083	2.750	1.583	1.000
TM5	4.250	4.188	3.500	1.438

filter outperforms the requantization filter. TM5 always obtains the highest quality, but it requires a full decoding/encoding of the video stream and cannot be employed as a real-time filtering mechanism.

4.2.2 MOS evaluation

MOS is a subjective measure of the perceived video quality. MOS values are obtained by applying weighted average to scores that testees give to the perceived video playout by comparing it with the original. The scores ranges from 1 (poor, distorsion noticeable) to 5 (excellent, no distorsion noticeable). In contrast to PSNR, MOS can measure the degradation in the temporal resolution caused by the frame discarding filter.

Table 1 summarizes the results for one of the evaluated streams. We also conducted several experiments on other video sources. The low-pass filtered streams achieves good scores in most of the cases. For the requantization filter, it scores low on VBR scenes with high bit rate variation and on complex scenes. In the case of the frame discarding filter, it obtains high scores in fast scenes, but low on smooth, slow-moving scenes. This is due to the fact that in slow-moving scenes it is easier to detect the absence of frames, thus user satisfaction is reduced. Between the low-pass and requantization filters, the former obtains higher scores. We think this is due to the reason that the low-pass filter degrades pictures gradually by blurring the edges of the objects in the picture, whereas general degradation in spatial quality is perceived with the requantization filter.

4.3 Processing Requirements

We measured the processing time required by the filters, using gprof [9] to investigate their practicality in real-time video filtering. The filtering programs were written in C and ran on a 1 GHz Pentium III PC. We tested video sequences of 1, 2, 4, 6 and 8 Mb/s and 50 seconds long against different target rates.

The frame discarding filter is very fast compared with the other two, taking less than 0.5 seconds to filter any of the 50-second sequences. The low-pass filters could filter the input stream between 10 and 53 seconds, and the requantization filter took between 10 and 100 seconds, depending on the input and target rate.

The above values can only be considered as a preliminary insight about their applicability in active routers. The programs were not totally optimized for speed. We must also consider that the router processor's performance and instruction set can be completely different from those of a PC. Even if they are used on a PC-based router, we must take into consideration that the processing occurs per packet unit, introducing processing overhead.

5 Conclusions

We implemented and evaluated three different filtering mechanisms for MPEG video streams, and proposed simple rate controlling algorithms. From the evaluation results, the frame discarding filter is the simplest, fastest, but it is not able to offer smooth rate control, and the artifacts it produces are very noticeable in slow moving scenes. The low-pass filter has higher requirements in processing capabilities, but can offer smooth rate adjustment. Compared with the low-pass filter, the requantization filter seems ineffective, since for the same input and target rate, it requires more processing and usually produce an inferior quality output.

As future work, we will further explore the applicability of the filters in active routers, in particular to network-processor-based systems.

References

[1] J. M.Smith, K. L.Calvert, S. L.Murphy, H. K.Orman, and L. L.Peterson, "Activating networks: A progress report," *IEEE Computer*, vol. 12, pp. 32–41, Apr. 1999.

[2] H. Akamine, N. Wakamiya, M. Murata, and H. Miyahara, "An approach for heterogeneous video multicast using active networking," in *Proceedings of Second International Working Conference on Active Networks (IWAN2000)*, pp. 157–170, Oct. 2000.

[3] ISO/IEC DIS 13818-2, "MPEG-2 Video." ISO Standard, 1996.

[4] N. Yeadon, F. García, D. Hutchinson, and D. Shepherd, "Filters: QoS support mechanisms for multipeer communications," *IEEE Journal on Selected Areas in Communications*, vol. 14, pp. 1245–1262, Sept. 1996.

[5] S. Gopalakrishnan, D. Reininger, and M. Ott, "Realtime MPEG system stream transcoder for heterogeneous networks," in *Proceedings of Packet Video 99*, Apr. 1999.

[6] M. Hemy, U. Hengartner, P. Steenkiste, and T. Gross, "MPEG system streams in best-effort networks," in *Proceedings of Packet Video 99*, Apr. 1999.

[7] K. Nakada, "Implementation and evaluation of filtering mechanisms for real-time heterogeneous video multicast." Bachelor Thesis (in Japanese), Osaka University, February 2001.

[8] ISO/IEC/JTC1/SC29/WG11, "Test Model 5," Jan. 1994.

[9] J. Fenlason and R. Stallman, "GNU gprof - the GNU profiler." available at `http://www.gnu.org`.