

PRIORITY DROPPING IN NETWORK TRANSMISSION OF SCALABLE VIDEO

Tao Tian¹, Adam H. Li¹, Jiangtao Wen², and John D. Villasenor¹

¹ Electrical Engineering Department, University of California, Los Angeles, CA 90095
{ttian, adamli, villa} @icsl.ucla.edu

² PacketVideo Corp., San Diego, CA 92121
gwen@packetvideo.com

ABSTRACT

By constructing a model which takes into account the frame dependence of multimedia streams, we analyze the performance of different packet-dropping mechanisms and find that scalable video combined with the priority dropping mechanism can bring higher throughput, lower delay and lower delay jitter. We also find the optimum system parameters for multimedia transmission. This study is important for achieving better performance for video transmission over IP networks.

1. INTRODUCTION

With the growth of the Internet, packet-switched networks are becoming increasingly important in all aspects of communications. However, packet-switched networks, in particular, the widely used IP-based networks, are data oriented and have some shortcomings for real-time multimedia transmission. One of the problems is the dropping of packets when network congestion happens. For data service, the lost packets can be recovered by retransmission. But for a real-time multimedia stream, the retransmitted packets usually expire before received, and the retransmission itself aggravates the congestion.

Many solutions have been proposed to improve Quality of Service (QoS) in IP-based networks. One way is to include the priority-dropping mechanism in the network protocol. Multimedia frames are encapsulated in data packets which are assigned different priorities according to their importance in reproducing the original information. When congestion occurs, lower-priority packets are dropped first to ease the demand on network resources so those higher-priority packets may get through intact. There are many ways to assign priorities to different coding blocks. In video coding, protocols like MPEG-4 and ITU-T H.263+ have included scalable encoding for layered video [1]. In a recent study by Bajaj *et al.* [2], it is found that the performance benefit of priority dropping for layered video is lower than expected. A potential function is used in their model to evaluate the performance but it does not reflect the dependence between frames. A media

stream is a sequence of highly correlated frames, more than just the sum of bits. A frame loss can lead to severe damage to the quality of subsequent frames. We will show, with the frame-dependence taken into account, that scalable video combined with priority-dropping mechanism can bring higher throughput, lower delay and lower delay jitter in busy networks.

2. SIMULATION MODEL

We use the typical two-layer time-scalable frame structure shown in Fig. 1. The arrows indicate the dependence between frames. Each frame is transmitted as one packet.

N is the number of frames in a group beginning with an I frame. NoB is the number of B frames between two consecutive base layer frames.

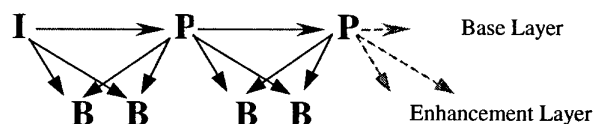


Fig. 1. Structure of the 2-layer model

We use a quasi-stationary Variable Bitrate (VBR) source with average frame sizes: S_I , S_P and S_B respectively. $N = N_I + N_P + N_B$ where N_I , N_P and N_B are the number of I, P and B frames in a group, respectively. For the 2-layer structure, $N_I = 1$ and $N_B / N_P = \text{NoB}$. The average frame size is $S_{\text{avg}} = (S_I + N_P S_P + N_B S_B) / N$.

To find the optimum parameters for scalable video, we evaluated the encoding schemes shown in Table 1. NoB = 0 represents the unlayered stream. Each frame is encoded to the same quality. I frames are assumed to have the same size 1. The relative sizes of P and B frames are derived from the results of the experimental study of Pegler *et al.* [3].

NoB	N_P	N_B	S_I	S_P	S_B	S_{avg}
0	24	0	1.00	0.41		0.43
1	12	12	1.00	0.60	0.20	0.42
2	8	16	1.00	0.65	0.29	0.43
3	6	18	1.00	0.70	0.34	0.45
5	4	20	1.00	0.80	0.41	0.50

Table 1. Five typical encoding schemes for $N = 25$

As Côté *et al.* pointed out [4], the case of NoB = 1 has the highest encoding efficiency for typical streams. However, we will find later in our model, that the cases of NoB = 2 or 3 have better overall performance.

To model the network, we assume a single server, single flow system. There is one bottleneck with size b , Poisson arrival (intensity λ frames / sec) and exponentially distributed processing time ($\frac{1}{\mu}$ sec / unit size). Note we

assume the processing time of a frame is proportional to its size, hence, it takes one second to process one I frame and less to process one P frame or one B frame because of their smaller sizes. Actually, μ is the bandwidth metric. The service policy can be uniform or non-preemption priority dropping. Our model is bursty to some extent for both arrival and service, which is the nature of the IP-based network. $(\lambda / \mu)_0$ is the balance point where the average arrival rate and processing rate match.

In this environment, frame loss occurs only when the buffer overflows and certain frames are dropped. The data corruption during transmission is negligible and re-transmissions are not engaged.

3. SIMULATION RESULTS

We compare four cases:

[U]: Unlayered stream with uniform dropping. NoB = 0 in this case.

[L_i] (i = NoB = 1, 2, 3, and 5): Layered stream with uniform dropping. We use this case to evaluate the performance of layered video transmitted in a network without priority dropping support.

[H_i] (i = NoB = 1, 2, 3 and 5): Layered stream with head dropping (a priority-dropping scheme where frames are dropped at the head of the buffer).

[T_i] (i = NoB = 1, 2, 3 and 5): Layered stream with tail dropping (a priority-dropping scheme where frames are dropped at the tail of the buffer).

R_{eff} (Effective Frame Rate, frame / sec) is defined as the rate of frames that are successively received and correctly decoded (all of the frames that the current decoded frame depends on for decoding are received). Fig. 2 and 3 show the relationship between R_{eff} and load for the four cases defined above.

We see from Fig. 2 and 3 that there exists an *optimum working point* where R_{eff} is highest. For [L_i] and [U], this point is below $(\lambda / \mu)_0$; for [H_i] and [T_i], it is above $(\lambda / \mu)_0$. R_{eff} of [L_i] (i ≤ 3) is higher than or at least close to that of [U], which means the performance does not deteriorate if we try to receive the layered video through a uniform dropping network. R_{eff} of [H_i] and [T_i] is much higher than that of [U]. The width of the curve increases with NoB; hence, higher throughput is achieved in very busy networks. The reason is that when higher NoB value is

adopted, a greater proportion of the frames can be dropped without severely affecting the decoding process.

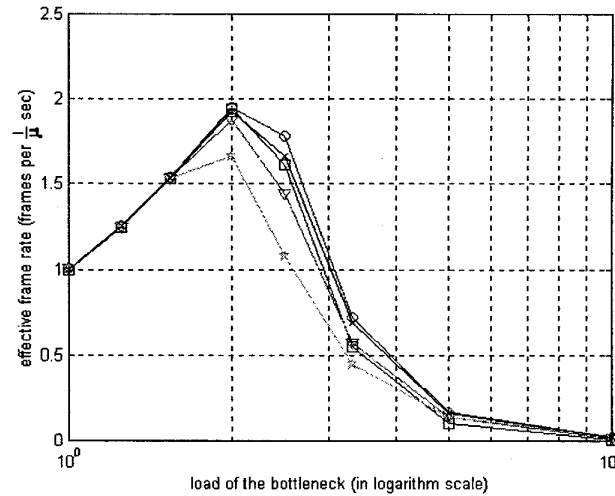


Fig. 2. $R_{eff} \sim$ Load ($N=25$, $b = 10$) for [U] and [L_i]
 ■ for [U]; ○ for [L₁]; × for [L₂]; ▽ for [L₃]; ☆ for [L₅]

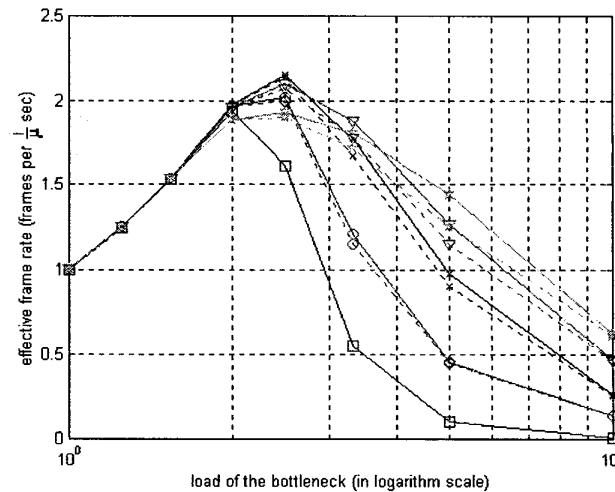


Fig. 3. $R_{eff} \sim$ Load ($N=25$, $b = 10$) for [U], [H_i] and [T_i]. ■ for [U];
 ○ for [H₁]/[T₁]; × for [H₂]/[T₂]; ▽ for [H₃]/[T₃]; ☆ for [H₅]/[T₅]
 Solid lines for [H_i]; dashed lines for [T_i]

In order to evaluate the overall performance, a performance function P is used. It can be assumed that P is affected mainly by two factors: the transmission throughput and the encoding efficiency. Hence, $P = (R_{eff}, S_{avg})$, where R_{eff} represents the transmission throughput and S_{avg} for the encoding efficiency. As an example, we assume $P = R_{eff} / S_{avg}$ and get the performance curve shown in Fig. 4. Obviously H₂ and H₃ have the best performance since the best P they can achieve is higher,

and the range in which they can provide satisfactory performance is wider compared to other cases.

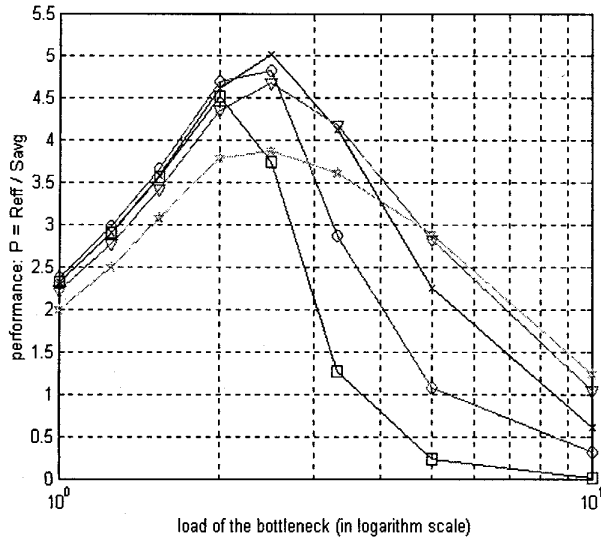


Fig. 4. Performance ~ Load ($N=25, b=10$) for [U] and $[H_i]$
 \square for [U]; \circ for $[H_1]$; \times for $[H_2]$; ∇ for $[H_3]$; \star for $[H_5]$

Fig. 4 and 5 show the relationship between delay of effective frames and load. $[L_i]$ generally have slightly higher delay than [U]. The priority dropping schemes, especially $[H_i]$, have lower delay than [U]. The reason is that in the priority dropping schemes, the frames dropped usually have higher delay than the frames received. $[H_i]$ are still better than $[T_i]$ because they make smarter decisions about which frames to drop (shown later).

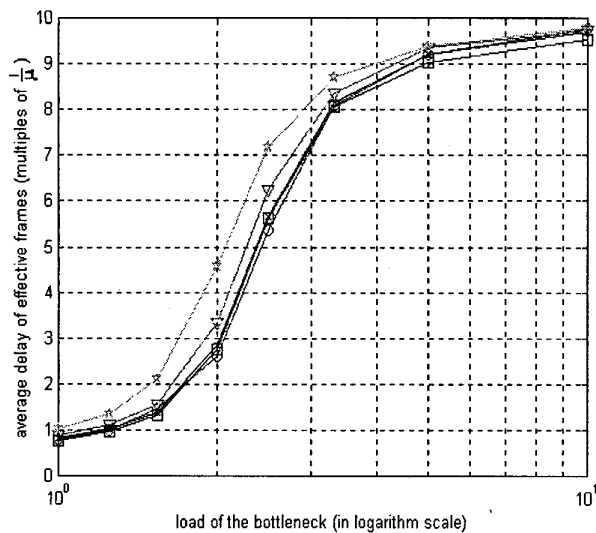


Fig. 5. Delay ~ Load ($N=25, b=10$) for [U] and $[L_i]$
 \square for [U]; \circ for $[L_1]$; \times for $[L_2]$; ∇ for $[L_3]$; \star for $[L_5]$

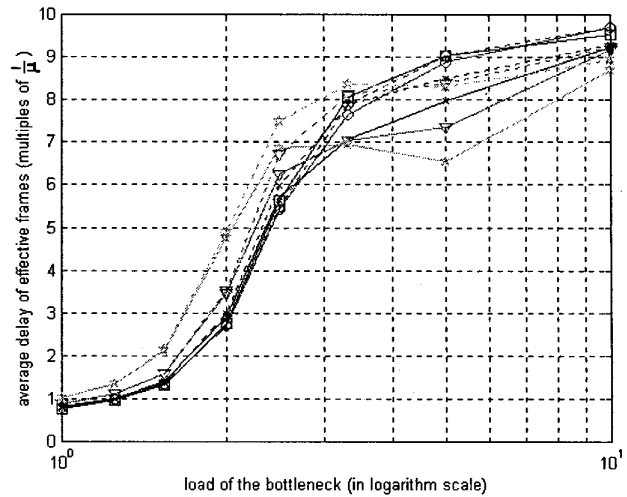


Fig. 6. Delay ~ Load ($N=25, b=10$) for [U], $[H_i]$ and $[T_i]$. \square for [U]; \circ for $[H_1]$ / $[T_1]$; \times for $[H_2]$ / $[T_2]$; ∇ for $[H_3]$ / $[T_3]$; \star for $[H_5]$ / $[T_5]$
 Solid lines for $[H_i]$; dashed lines for $[T_i]$

The delay and delay jitter for received frames can be shown with the delay distribution curve which is a slightly modified version of the probability density function. The only difference is the placement of an impulse at ∞ to represent the lost frames. See Fig. 6 and 7.

We choose the case for $\lambda / \mu = 2.5$ which is close to the balance point $(\lambda / \mu)_0$. Since the network is not very busy with this load, we see from Fig. 7 and 8 that there is no great difference between the position of the center of the curves. But there does exist difference between the width of the curves which represents the delay jitter. $[H_i]$ has a narrower distribution about a higher peak than [U]. The delay jitter of $[T_i]$ is between that of [U] and $[H_i]$.

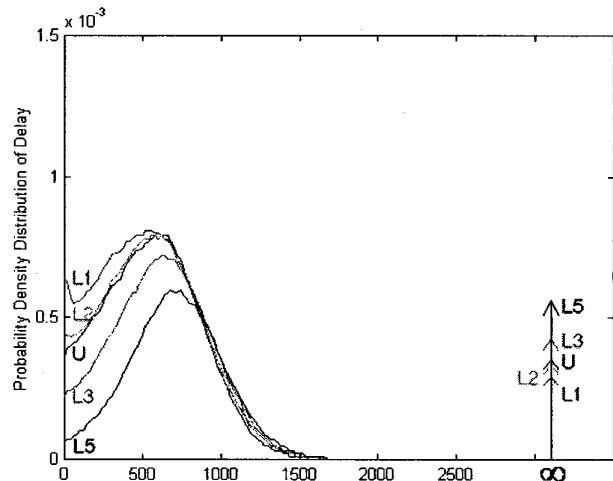


Fig. 6. Delay distribution curve ($N=25, b=10, \lambda/\mu=2.5$) for [U] and $[L_i]$

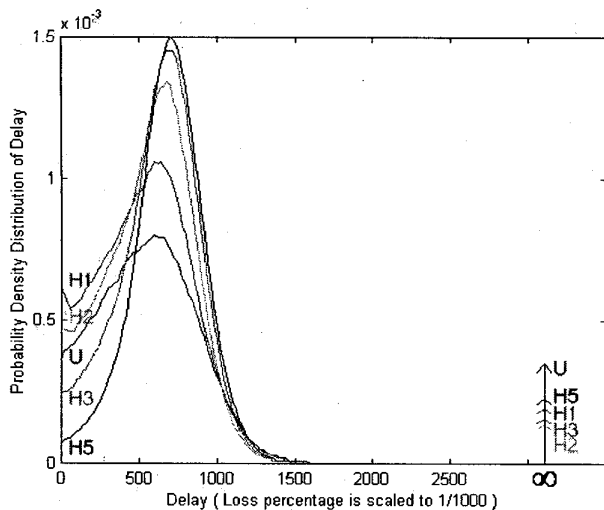


Fig. 7. Delay distribution curve ($N=25$, $b = 10$, $\lambda/\mu = 2.5$) for [U] and $[H_i]$

The reason that priority dropping achieves much higher throughput, lower delay and delay jitter is that it always tries to keep the most important information. The percentage of base layer frames in the received stream is much higher than that of enhancement layer frames when the load goes beyond the bandwidth limit.

We note that among the two priority dropping schemes $[H_i]$ and $[T_i]$, the former is a better choice since it has better throughput and delay performance as we showed above. The reason can be explained with the next example.

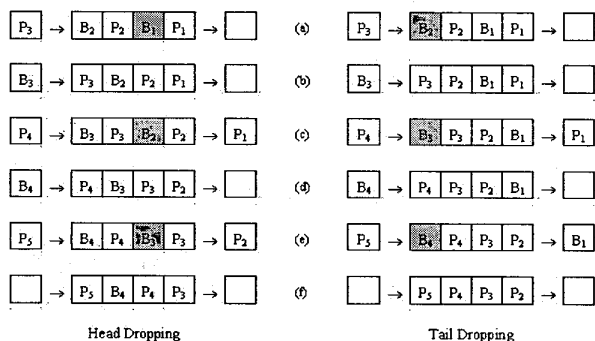


Fig. 8. An example to show the difference between $[H_i]$ and $[T_i]$

In this example, we have a busy network with $NoB = 1$, $b = 4$ and equal frame sizes. The arrival rate is 1 frame per second. The service rate is 0.5 frames per second. Both schemes start at the same state with P_1 , B_1 , P_2 and B_2 occupying all the vacancies of the buffer. At time (a), P_3 arrives. It kicks off B_1 in $[H_i]$ and B_2 in $[T_i]$. At time (b), B_3 comes and fortunately P_1 moves out at the right time to make a vacancy for it. Then at time (c), P_4 arrives and finds no vacancy. One B frame has to be removed. B_2 and B_3 are removed in $[H_i]$ and $[T_i]$ respectively. At time (d), we can see different distributions in two buffers: In the

$[H_i]$ buffer, more P packets are concentrated at the head of the buffer, while in the $[T_i]$ buffer, P packet are concentrated at the tail. So at time (e), $[H_i]$ moves out P_2 while $[T_i]$ moves out B_1 , which is less important for the decoding process. At time (f), the $[T_i]$ buffer are congested by P packets while the $[H_i]$ buffer still has one B packets to be kicked off if another P frame arrives. So we say that head dropping scheme works better than tail dropping scheme because the former sends out the most important information as soon as possible which guarantees a better throughput and delay performance.

We also find that delay increases almost linearly with buffer size b . A buffer size of around $10S_i$ can provide the best performance. Larger buffers bring little R_{eff} improvement but cause longer delay.

4. CONCLUSION

We studied the throughput and delay performance of different dropping mechanisms for layered and unlayered video streams. We found that by using priority dropping, especially head dropping, we can improve overall performance (throughput, delay and delay jitter).

5. REFERENCES

- [1] ITU-T, Recommendation H.263 Version 2, "Video coding for low bitrate communication", Jan. 1998.
- [2] S. Bajaj, L. Breslau, and S. Shenker, "Uniform versus Priority Dropping for Layered Video," in *Computer Communication Review*, vol. 28, pp. 131 - 143, 1998.
- [3] D. Pegler, N. Yeadon, D. Hutchison and D. Shepherd, "Incorporating Scalability into Networked Multimedia Storage Systems," Department of Computing, Lancaster University, Lancaster LA 4YR, UK.
- [4] G. Côté, B. Erol, M. Gallant, and F. Kossentini, "H.263+: Video Coding at Low Bit Rates," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 849 - 866, Nov. 1998.
- [5] S. Bajaj, L. Breslau and S. Shenker, "Is Service Priority Useful in Networks?" in *Performance Evaluation Review*, vol. 26, pp. 66 - 77, April 1998.
- [6] W. Tan and A. Zakhor, "Real-time Internet Video Using Error Resilient Scalable Compression and TCP-Friendly Transport Protocol," *IEEE Trans. Multimedia*, vol. 1, no. 2 pp. 172 - 186, June 1999.
- [7] A. Adas, "Supporting Real Time VBR Using Dynamic Reservation Based on Linear Prediction," in *INFOCOM'96*, San Francisco, CA, 1996.