Proxy Design for Improving the Efficiency of Stored MPEG-4 FGS Video Delivery over Wireless Networks

Feng-Jung Liu and Chu-Sing Yang

Abstract: **The widespread use of the Internet and the maturing of digital video technology have led to an increase in various streaming media application. However, new classes of hosts such as mobile devices are gaining popularity, while the transmission became more heterogeneous. Due to the characteristics of mobile networks such as low speed, high error bit rate, etc., the applications over the wireless channel have different needs and limitations from desktop computers. An intermediary between two communicating endpoints to hide the heterogeneous network links is thought as one of the best approaches. In this paper, we adopted the concept of inter-packet gap and the sequence number between continuously received packets as the error discriminator, and designed an adaptive packet sizing mechanism to improve the network efficiency under varying channel conditions. Based on the proposed mechanism, the packetization scheme with error protection is proposed to scalable encoded video delivery. Finally, simulation results reveal that our proposed mechanism can react to the varying BER conditions with better network efficiency and gain the obvious improvement to video quality for stored MPEG-4 FGS video delivery.**

Index Terms: **BER estimation, FGS, packet length adaptation, QoS.**

I. INTRODUCTION

Transmission of video over hybrid network will become increasingly important in future Internet communication. There is much current interest in the idea of placing an intermediary between two communicating endpoints, such as client and server, to reduce traffic over low bandwidth or heterogeneous network links. In addition to increase bandwidth utilization, the servers are also isolated from the high bit-error rate channel by the intermediary proxy. So, packet loss handling over the wireless network is the responsibility of the proxies.

However, the probability of a packet being in error is directly proportional to the packet size. Therefore, the choice of packet size has a great impact on the performance of changing networks. In this paper, we proposed an adaptive packet sizing mechanism at application-level or over transport layer in proxy server to improve the network efficiency under varying channel conditions.

The remainder of this manuscript is organized as follows. In Section II, some related technologies for video delivery and transmission error distinction are presented. Section III describes the design of our proposed adaptive packet sizing proxy and error packetization scheme. In Section IV, simulation results and discussions are presented. Finally, Section V summarizes our research contributions.

Manuscript received October 28, 2003; approved for publication by Kenichi Mase, Division III Editor, April 19, 2004.

F.-J. Liu is with the Department of Computer Science and Engineering from National Sun Yat-sen University in Kaohsiung and works with Department of Management Information System at Tajen Institute of Technology, Pingtung, R.O.C, email: liufj@cse.nsysu.edu.tw.

C.-S. Yang is with the Department of Computer Science and Engineering from National Sun Yat-sen University in Kaohsiung, email: csyang@cse.nsysu.edu.tw.

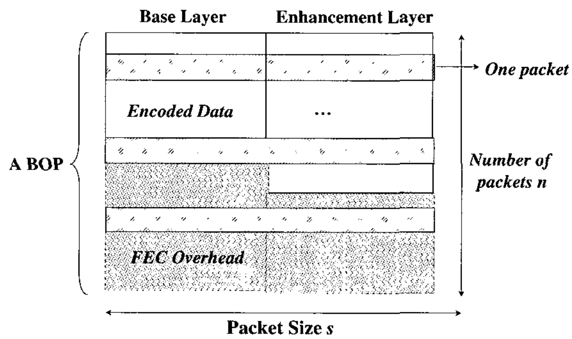


Fig. 1. The data structure of a BOP.

II. BACKGROUND AND RELATED WORKS

A. Scalable Video Coder

Many researches [1], [2] have proposed scalable video coders for Internet applications in the past. Among them, fine granular scalability (FGS) in MPEG-4 is one of the highly flexible coding techniques capable of delivering layered video data with precise rate control. The characteristics of FGS are considered as pluses especially for error-prone heterogeneous transmission environments, e.g., mobile video-on-demand systems. Several optimal FEC assignment schemes, which combined scalable coding with unequal error protection (UEP) and loss rate feedback, have already been proposed [3], [4]. Besides, the comparison of FEC encoding with UEP and equal error protection (EEP) also is discussed.

B. Packetization Scheme

Reed-Solomon (RS) codes are currently widespread for applications in digital communication and storage because they are convenient to analyze and are flexible. FEC codes are applied across packets, with unequal error protection to different layers according to their importance.

Typically, bit streams of all layers are interleaved into one block of packets (BOP) and the transmission packets are the rows of the BOP as Y. C. Su *et al.* described in [4]. The structure of a BOP is shown in the Fig. 1. In a generic FEC scheme, every

1229-2370/04/$10.00 © 2004 KICS

block of k data packets followed by $(n - k)$ redundancy packets forms a BOP with n packets. If at least k out of n packets were correctly received, the underlying source information can be correctly decoded. Otherwise, none of the lost packets could be recovered.

However, how many layers are transmitted and how video data and channel coding redundancy of each layer are distributed within the BOP needs to be transmitted as side information to the decoder. In our proposal, the proxy will take over the works with mobile clients. In another words, the proxy works without any modification to streaming server.

C. Adaptive Packet Length

With a constant bit error rate, the probability of a packet being in error is directly proportional to the packet size. Therefore, the choice of packet size has a great impact on the performance of changing networks [5]–[7]. The channel conditions over wired or fixed networks are not likely to change much over time. However, the conditions of wireless channels are not stable so it is not suitable to transmit fixed MTU over wireless links during the whole session. Generally, with high bit-error channel condition of wireless networks, the shorter packets are less likely to encounter packet error than long packets, but on the other hand, they are more burdened with header overhead. A tradeoff exists between the desire to decrease header overhead by increasing the size of packet and the need to reduce packet error by decreasing the packet size. Some papers [5], [6] have proposed schemes to do dynamic packet sizing at the link layer to reduce the packet loss rate. However, the adaptation at link layer is not fit well for the requirements of video delivery. After all, without FEC coding, the sender cannot guarantee that these segmented packets in data link layer will be received successfully with peer and uncompleted data is useless to the upper layer or application layer. And, even, with FEC coding, it is hard to differentiate between data packets for unequal error correction. Besides, automatic repeat request (ARQ) scheme may be used in high-speed LANs where round trip latencies are small [8].

D. The Error Discriminator

There are two main reasons for packet loss: Due to buffer overflow or due to transmission bit errors. Many papers discussed these problems utilizing the packet jitter or packet inter-arrival gap [9]–[11]. In a paper [11], it introduced the concept of inter-arrival packet gap (IPG) as the error discriminator to distinguish between them. Based on its description, the following heuristics are developed:

1. Let T_{min} denote the minimum inter-arrival time observed so far by the receiver during the connection.
2. Let P_o denote an out-of-order packet received by the receiver. Let P_i denote the last in-sequence packet received before P_o. Let P_g denote the time between arrivals of packets P_o and P_i. Finally, let the number of packets missing between P_i and P_o be n (assuming that all packets are of the same size).
3. If $(n+1) \times T_{min} \leq T_g < (n+2) \times T_{min}$, then the n missing packets are classified into wireless transmission errors. Otherwise, the n missing packets are assumed to be lost due to congestion.

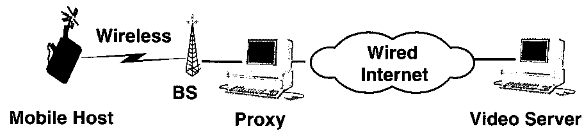


Fig. 2. System model.

In [11], it notes that the condition for identifying a packet loss as a wireless loss is quite restrictive. In addition, for the sake of the network protocols, it is preferable to mistake a wireless loss for a congestion loss, rather than the opposite. We utilized the use of sequence number of packets and IPG to improve the efficiency of error identification. The detail will be presented in next section.

III. ADAPTIVE CONTROL MECHANISM

In order to improve the quality of video delivery and reduce the packet loss rate with better good-put, we proposed a scheme of adaptive packet sizes to reduce the probability of packet being in error with the lossy environments [12]. Our scheme performed adaptation at application level or over the transport layer. The system model and the architecture are shown in Figs. 2 and 3, respectively.

For the sake of simplicity, we assume that the processing time at the base station and at the receiver is negligible. We also assume that:

1. Only the last link on the path is wireless.
2. The proxy is near to the base station. With such assumed restriction, only one route path exists between proxy and mobile host. No out-of-order packet is received. And, it's good for the accuracy of our proposed mechanism for packet error classification.

In the architecture of proxy as shown in Fig. 3, the proxy server is primarily used to mediate a proper network resource between the wired and wireless segments. Besides, the session manager is intended to maintain the consistency session state between streaming server to proxy and proxy to mobile client. It mainly consists of 4 components, including data buffering, rate control, the BER estimator, and data packetizer to support video delivery over heterogeneous network environments. Hence, our proposed proxy not only works as agent to forward requests and queue video data, but also plays a role in flow mediator and adaptive packet sizing actuator to mobile host. The executions of proxy are shown in Fig. 4 and illustrated clearly later.

Our proposed protocol is based on RTP/RTCP protocol [13]. The RTP is a transport protocol for real-time data. The data part of RTP is a thin protocol providing support for applications with real-time properties such as continuous media (e.g., audio and video), including timing reconstruction, loss detection, security, and content identification. RTCP is used for the quality of service and to convey information such as packets sent, packets lost, the jitter, and the round trip time in an ongoing media session.

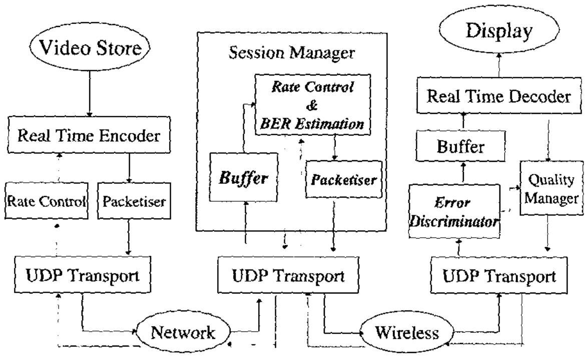


Fig. 3. The system architecture of proxy.

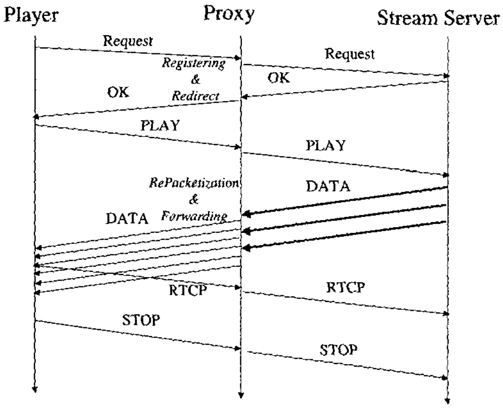


Fig. 4. The execution of proxy.

As shown in Fig. 4, the player firstly issues a request to the proxy, and the proxy registers and re-directs it to streaming server. Upon receiving "OK" message from the proxy, the player sends "PLAY" message to the proxy, and the proxy redirects it to stream server. The stream server starts to deliver the streaming data to the proxy. And the proxy will change data packet size dynamically according to the state of wireless channel and forward these streaming data to the player. During the execution, the player reports periodically its received status to the proxy by RTCP protocol. The proxy could learn the state of wireless channel from the receiver report. At last, the player sends "STOP" message to the proxy and finishes the session. At the same time, proxy forwards this message to the stream server.

A. Network State Estimation

There are two main reasons for packet loss: Packets get lost due to buffer overflow or due to bit errors. In the proposed mechanism, it is assumed that all packet losses in wired network be classified as congestion error and the available bandwidth resource of the wired network is much greater than wireless network. Therefore, we adopted the use of jitter as congestion indicator. Fig. 5 illustrated the concept of congestion indicator. As shown in Fig. 5, there exist two errors, one for congestion error and another for transmission error. The proxy received each packet from server with different intervals (T_1, T_2, and T_3)

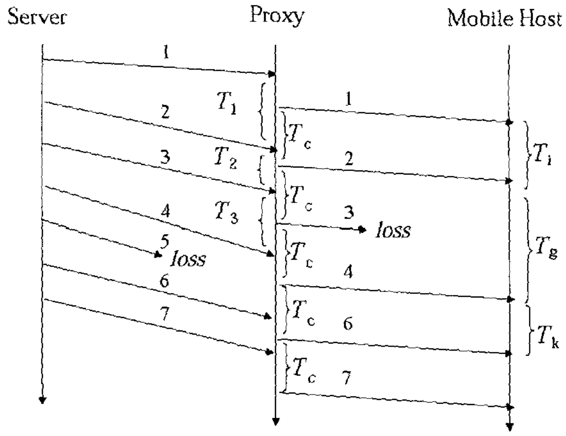


Fig. 5. Transmission error and congestion loss with inter-arrival gap.

because of transmission latency and queue delay. And, the proposed proxy acts as flow mediator to forward these packets in same time interval, T_c, to mobile host. In general, the T_i is nearly equal to T_k and T_g is nearly equal to $2 \times T_i$. By monitoring the sequence number of packets and measuring the jitter of packets, the error discriminator distinguishes transmission error from congestion loss.

For the sake of the network protocols, it is preferable to mistake a wireless loss for a congestion loss. Therefore, if packet loss happens, i.e., the difference of sequence number between continuous packets is greater than 1, the conditions for wireless loss are are first verified. These lost packets excluding wireless loss packets could be classified as congestion error. The following illustrates the rules for wireless loss. If a variable, no_wireless_loss, in (1) is greater than 1, it shows that wireless transmission error exists in these packet loss. Additionally, there exists some unknown lost packets if excluding these classified wireless loss packets, these packets are considered as congestion error. As to the parameter, α, in (1), it is used to cover the reduction of packet processing and queuing delay for packet dropping. However, the measurement of packet jitter between packets is one of essential factors for the error discriminator. It is measured as receiving packets. (2) for jitter measurement is used to reflect the changing traffic environment. The parameter, γ, is a proportion value to avoid the estimated jitter fluctuating frequently in a short period of time. In another words, if the variable, γ, gets closer to 1, the estimated jitter will be much sensitive to the transient state of network and be changed substantially. The processing of error discrimination is described clearly in Algorithm 3.1.

$$\text{no\_wireless\_loss} = \frac{J_{real}}{J_{current} \times (1-\alpha)} - 1, \tag{1}$$

where

J_{real}: The current IPG.

$J_{current}$: The currently estimated jitter.

α: Weighted value.

$$Jitter_{i+1} = Jitter_i \times (1-\gamma) + Jitter_{current} \times \gamma, \tag{2}$$

where

$Jitter_i$: The estimated jitter at time i.

$Jitter_{current}$: The currently estimated jitter.

γ: Weighted value.

```
int Wireless_loss=0;
// Count the number of the wireless loss packets.
int Congestion_loss=0;
// To count the number of the congestion loss packets.
int Received_pkts=0;
// To count the total received packets.
float α=0.1,γ =0.7;
// Weighted values for (1) and (2).
float Jitter, Jitter_real;
// To represent the estimated jitter and the currently measured jitter in (2).
void account_recv_pkt(packet P)
{
int Diff, wloss;
static int Last_seq;
// The last received packet's sequence number.
Jitter_real=local_time-last_time;
// The current time - the time of last received packet.
Received_pkts++;
// Increase the number of the received packets.
If((Diff= P.seq- Last_seq-1)>0) {
// Packet losses happen, if Diff>0.
wloss=Jitter_real/(Jitter*(1-α))-1;
// Calculate the number of wireless packets loss by (1).
If(wloss<=Diff) {
Wireless_loss+=wloss;
// Increase the number of congestion loss packets, excluding the wireless loss packets.
Congestion_loss+=(Diff-wloss);
}
else Wireless_loss+=wloss;
}
else Jitter= Jitter*(1-γ)+ Jitter_real*γ;
// Adjust the estimated Jitter by (2).
Last_seq=P.seq;
}
```

Algorithm 3.1: Packet error discriminator.

B. Adjustment Decision and Packetization

To improve the network efficiency of wireless transmission, we proposed the packet length adaptive scheme with varying channel conditions. The application good-put is defined as a function of user data size with respect to varying bit-error rate (BER) of the wireless networks. The application good-put refers the bandwidth of networks that have actually received after all the protocol header, including transport layer and network layer protocol, and the PHY and MAC overheads are removed. And, the normalized good-put, G_N, represents the efficiency of bandwidth to application. In order to analyze the behavior of our system model, let us assume these parameters as follows:

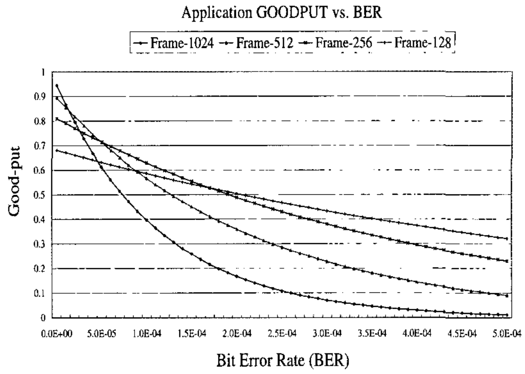


Fig. 6. The application good-put for different packet sizes with respect to varying BER condition.

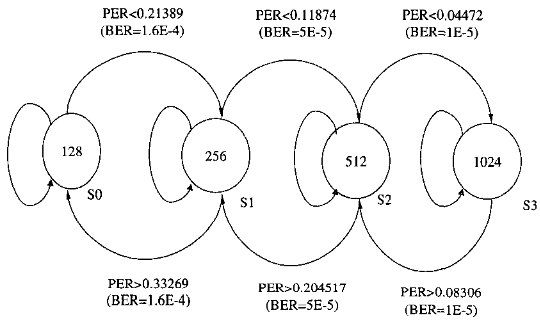


Fig. 7. The adaptive packet length model (initial state is $S2$).

G_N: Normalized good-put.

BER: Bit error rate.

PER: Packet error rate.

D: The length of user data.

H: The length of all protocol headers.

B: The total used bandwidth.

$$G_N = Normalize(\frac{\text{Good-put}}{B}) = \frac{(1-\text{BER})^{(D+H)}}{1+H/D}. \quad (3)$$

According to (4), the relationship between packet error rate (PER) and its related bit-error rate (BER) is obvious. In this situation, we assume that BER is constant over time, so the distance between two consecutive bit errors is exponentially distributed over the bit stream.

$$\text{PER} = 1 - (1-\text{BER})^{(\text{length of packet in bits})}. \quad (4)$$

For the simplicity and performance of implementation, there exist 4 states in this model. Within this model, these states are labeled with $S0$, $S1$, $S2$, and $S3$, and represented with different data sizes such as 128, 256, 512, and 1024 bytes. In Fig. 6, these

Table 1. Good-put for different packet sizes over varying BER environment.

	Packet length = 1024		Packet length = 512		Packet length = 256		Packet length = 128	
BER	PER	Good-put	PER	Good-put	PER	Good-put	PER	Good-put
0.00000	0.00000	0.94465	0.00000	0.89510	0.00000	0.81013	0.00000	0.68085
0.00001	0.08307	0.86618	0.04473	0.85507	0.02496	0.78990	0.01493	0.67069
0.00002	0.15923	0.79423	0.08746	0.81682	0.04930	0.77018	0.02963	0.66068
0.00005	0.35184	0.61229	0.20452	0.71204	0.11874	0.71393	0.07244	0.63153
0.00006	0.40568	0.56143	0.24010	0.68019	0.14074	0.69611	0.08629	0.62210
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0.00015	0.72771	0.25722	0.49664	0.45056	0.31561	0.55444	0.20198	0.54334
0.00016	0.75033	0.23585	0.51916	0.43040	0.33270	0.54060	0.21389	0.53522
0.00017	0.77108	0.21625	0.54067	0.41115	0.34936	0.52710	0.22563	0.52723
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0.00049	0.98574	0.01347	0.89384	0.09503	0.71033	0.23467	0.52152	0.32577
0.00050	0.98693	0.01235	0.89859	0.09077	0.71757	0.22881	0.52867	0.32091
0.00051	0.98801	0.01132	0.90313	0.08671	0.72462	0.22309	0.53571	0.31612

curves are derived from Table 1, which illustrates the application good-put for different packet sizes with respect to varying BER conditions. However, we tried to gain the higher efficiency of video delivery, so the ideal curve is along with on the top of all curves. As shown in Fig. 7, it depicted the model of channel states, along with different packet sizes, with different BER thresholds. The thresholds could be determined with these intersection points with maximum good-put among these lines over varying BER as shown in Fig. 6.

In the proposed packetization scheme, it is also determined that the protection ratio of video data packets is estimated in each state. The protection ratio of data packets represents the video data could be retrieved correctly if the packet error rate is not greater than the protection ratio. For example, if the current network is estimated to be in state $S2$, it implies that packet length is set to 512 bytes and its possible PER exists between 0.04472 and 0.204527. If it gets worse or better, the adaptive packet size will be performed with our proposed model as shown in Fig. 7. However, in order to gain more stable video quality, the proxy has to perform different error protection ratio for video delivery to react to changing BER in its own covered range. The value of the protection ratio is set to maximum tolerable PER of its currently estimated state.

$$\mathrm{PER}_{i+1} = \mathrm{PER}_i \times (1-\beta) + (L/T) \times \beta, \quad (5)$$

where

PER: The estimated PER of wireless link.

L: The number of loss packets reported by receiver.

T: Total packet sent from sender to receiver.

β: Weighted value.

Mobile clients used Algorithm 3.1 to identify different errors. According to the mobile client's feedbacks, the proxy could estimate the PER by (5). Thereafter, the proxy is able to process the packet sizing and determines the protection ratio of video delivery. However, the period time of receiver's report and the proportion variable, β, in (5) have a great effect on the estimation of PER. If the period of receiver's report becomes shorter and the variable, β, gets closer to 1, the estimation of PER could reflect the channel condition sooner. Comparatively, the proxy will change the packet sizing more frequently to react to the change of network state.

IV. SIMULATION RESULTS AND DISCUSSION

We used NS-2 to simulate our proposed scheme with different channel conditions. The simulation results are divided into 2 parts and shown below.

A. Simulation of Network Efficiency

Firstly, the adaptive scheme's initial state is set to $S2$, i.e., packet size is set to 512 bytes, The time duration of simulation is 90 seconds and receiver reports by 5 seconds periodically.

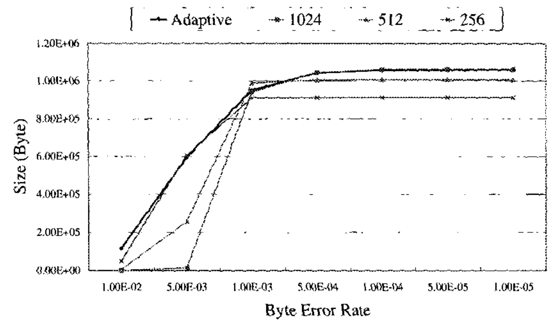
The Figs. 8 and 9 showed the total received data size and the network efficiency with different packet sizes with respect to varying uniform BER conditional networks. Obviously, if the channel condition is better, the application with larger packet length get more user data. On the opposite, if the channel condition got worse, the smaller packet got the better utilization.

As shown in Fig. 10, the simulation results proved the proposed mechanism has gained the obvious piece-wise improvement over the varying channel conditional environment. In Fig. 10, it shows the dynamic behavior of the proposed mechanism for the time-varying BER environment. It illustrated the proposed adaptive packet sizing mechanism could react to the time-varying BER wireless environment.

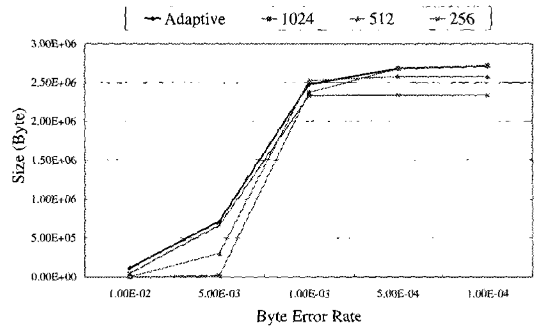
B. Video Quality Analysis

In this section, we applied the adaptive packet sizing mechanism to the BOP packetization scheme to the proxy design. The protection ratio of BOP is determined with the estimated channel state. The packetization scheme could be adapted to different error protection schemes. The unequal error protection (UEP) and the equal error protection (EEP) schemes were discussed in [12]. For the sake of simplicity, the EEP scheme is selected for analysis and discussed the efficiency of our proposed scheme. The test conditions for video stream are shown below:

1. Frame resolution = CIF format (352 × 288).
2. Constant stream rate = 256 kbps.

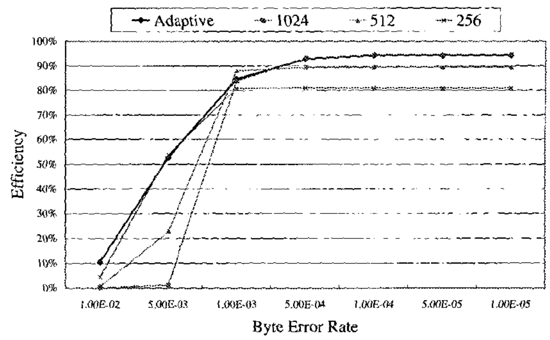


(a)

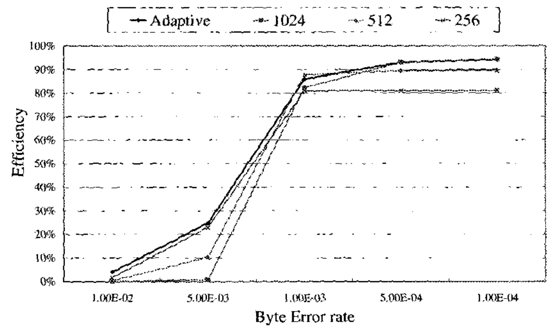


(b)

Fig. 8. Total received data with respect to varying channel conditions with receiver's report every 5 seconds: (a) CBR = 0.1 Mbps, duration = 90 secs, (b) CBR = 256 kbps, duration = 90 secs.



(a)



(b)

Fig. 9. Total network efficiency with respect to varying channel conditions with receiver's report every 5 seconds: (a) CBR = 0.1 Mbps, duration = 90 secs, (b) CBR = 256 kbps, duration = 90 secs.

3. 1 GOP = 1 Intra-frame accompanied with 14 inter-frames and frame rate = 15 fps.
4. Sequence length = 9 GOPs.
5. MPEG-4 FGS method has been adopted to generate the scalable video streams. Each MPEG-4 FGS video stream is composed of one base layer and one enhancement layer.
6. We used equal protection to different layers for the sake of simplicity.
7. Assumed that the max tolerable BER of channel condition is set to 2×10^{-4}.

The analytical PSNR of stored MPEG-4 FGS video with different protection ratio over varying BER environments is shown in Fig. 11. The avg. PSNR is calculated with the mean PSNR of these 9 BOPs with corresponding error protection ratio under varying BER environment. The analysis result shows that our proposed mechanism could gain a more stable video quality for stored MPEG-4 FGS video with BOP packetization scheme.

V. CONCLUSION

In this paper, we proposed an intermediate proxy deployment to alleviate the error packetization load of video server and accommodate the wired network to varying BER wireless environment. In addition, the proxy acts as the flow mediator to forward data packet at constant bit rate and to reduce inter-arrival packet gap (IPG) variation to mobile hosts. On network state estimation, we adopted the IPG and the sequence number between continuously received packets as the error discriminator. Based on the result of network estimation, an adaptive packet sizing mechanism is proposed to improve the network efficiency. According to the mobile client's report feedback, the proxy server estimates the status of wireless network and chooses a proper packet length for improving the network efficiency. With our proposed scheme, the smaller IPG variation will be better to the veracity of proposed error discrimination. This proposal is simple but rather effective since the possibility of packet corruption depends much on the size of packet. Simulation results reveal that our proposed mechanism not only reacts to the varying channel condition with better network efficiency, but also gain the more stable video quality for stored MPEG-4 FGS video with BOP packetization scheme.

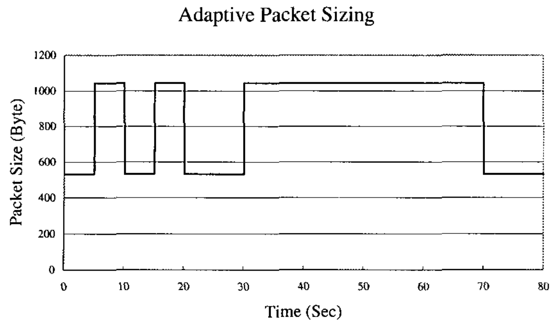
ACKNOWLEDGMENTS

This work is supported in part by the National Science Council with grants NSC 92-2213-E-110-020.

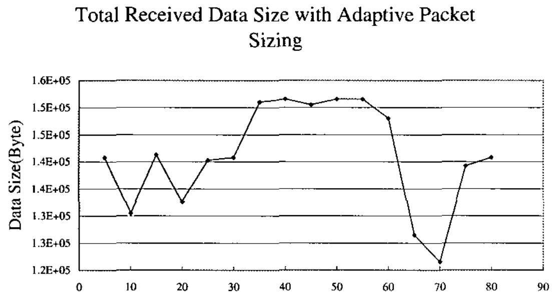
REFERENCES

[1] W. Li, "Overview of fine granularity scalability in MPEG-4 video standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 3, pp. 301–317, Mar. 2001.

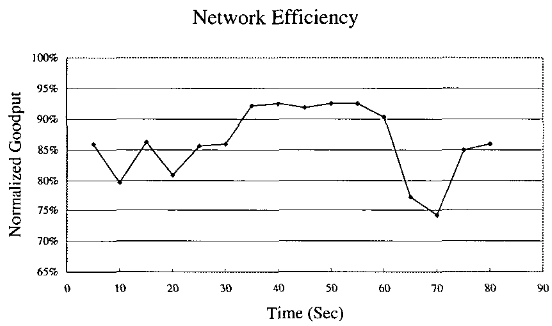
[2] F. Wu, S. Li, and Y. Chang, "A framework for efficient progressive fine



(a)



(b)



(c)

Fig. 10. A case of simulation, which at the beginning of a simulation run, the BER is set to 0.001, the BER is changed into 0.0001 at 30 seconds, and then the BER is changed to 0.001 at 60 seconds (CBR = 256 kbps, α = 0.1, γ = 0.7, β = 0.7, receiver's report every 5 seconds): (a) Adaptive packet sizing with the time-varying BER environment, (b) the total received data size over the time-varying BER environment, (c) the averaged network efficiency over the time-varying BER environment.

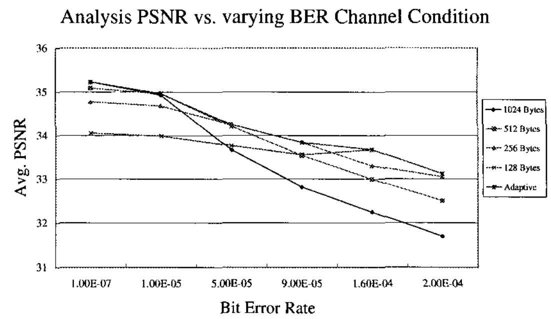


Fig. 11. The analytical PSNR with different packet size and protection ratio over varying BER environment.

granularity scalable video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 3, pp. 332–344, Mar. 2001.

[3] G. Wang *et al.*, "Channel-adaptive unequal error protection for scalable video transmission over wireless channel," in *Proc. SPIE VCIP 2001*, vol. 4310, San Jose, CA, Jan. 2001, pp. 648–655.

[4] Y. C. Su, C. S. Yang, and C. W. Lee, "Optimal FEC assignment for scalable video transmission over burst error channel with loss rate uplink," in *Proc. Packet Video*, Nantes, France, 2003.

[5] W.-T. Chen and J.-S. Lee, "Some mechanisms to improve TCP/IP performance over wireless and mobile computing environment," *IEEE ICPADS 2000*, Iwate, Japan, July 2000, pp. 437–444.

[6] E. Modiano, "An adaptive algorithm for optimizing the packet size used in wireless ARQ protocols," *Wireless Networks*, vol. 5, no. 4, pp. 279–286, 1999.

[7] T. Vu, D. Reschke, and W. Horn, "Dynamic packet size mechanism (DPSM) for multimedia in wireless networks," in *Proc. 8th Workshop on "Multimediale Informations- und Kommunikationssysteme"* im Rahmen der Net.ObjectDays 2002, Erfurt, Tagungsband S., pp. 667–674.

[8] B. Vandalore *et al.*, "A survey of application layer techniques for adaptive streaming of multimedia," *J. Real Time Syst., Special Issue on Adaptive Multimedia*, Apr. 1999.

[9] R. Rejaie, M. Handley, and D. Estrin, "RAP: An end-to-end rated-based congestion control mechanism for realtime streams in the internet," in *Proc. INFOCOM'98*, vol. 3, 1998, pp. 1337–1345.

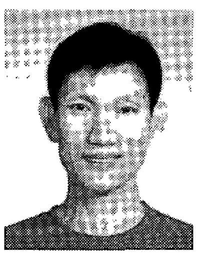
[10] S. Cen, P. C. Cosman, and G. M. Voelker, "End-to-end differentiation of congestion and wireless losses", in *Proc. Multimedia Computing and Networking(MMCN)*, Jan. 2002, pp. 1–15.

[11] S. Biaz and N. H. Vaidya, "Discriminating congestion losses from wireless losses using inter-arrival times at the receiver," in *Proc. IEEE Symposium ASSET'99*, Richardson, TX, USA, Mar. 1999, pp. 10–17.

[12] F. J. Liu, C. W. Tseng, and C. S. Yang, "A proxy design for improving the efficiency of video delivery over wireless networks," in *Proc. 7th International Symposium On DSP and Communication Systems and 2nd Workshop on The Internet and Telecommunications and Signal Processing*, Dec. 2003.

[13] RTP Resource, *http://www.cs.columbia.edu/ hgs/rtp/*.

Feng-Jung Liu received M.S. degrees in Computer Science and Engineering from National Sun Yat-sen University in 1994. Currently, he is a Ph.D. student in Dept. of Computer Science and Engineering at National Sun Yat-sen University in Kaohsiung and works with Dept. of Management Information System at Tajen Institute of Technology, Pingtung, ROC. He affiliated with Internet Technology Lab. His research interests include distributed system and multimedia communication.



Chu-Sing Yang received the B.S. degree in Engineering Science, and the M.S. and Ph.D. degrees in Electrical Engineering from the National Cheng Kung University, Taiwan, in 1976, 1984, and 1987, respectively. He joined the faculty of the Department of Electrical Engineering, National Sun Yat-Sen University, Taiwan, as an associate Professor in 1988. Since 1993, he has been a professor of the Department of Computer Science and Engineering, National Sun Yat-Sen University. He was a Chair of Department of Computer Science and Engineering, National Sun Yat-Sen University from August 1995 to July 1999. He was a director of Computer center, National Sun Yat-Sen University from August 1998 to October 2002. He was a Program Chair of ICS-96 and Program Co-Chair of ICPP-2003. His research interests include multimedia communication, parallel and distributed systems, scalable web servers, and wireless communication.