DESIGN AND IMPLEMENTATION OF AN INTERACTIVE TRUE VOD SYSTEM ON ADSL WITH SCARCE RESOURCES AT THE SET-TOP BOX

Chia-Hui Wang[#], Hung-Yu Kao[&], Shin-Hung Chang[&], Ray-I Chang[&], Shun-Chin Hsu[#], Tei-Wei Kuo[#], Jan-Ming Ho[&]

[&]Institute of Information Science,
Academia Sinica, Nankang, Taipei, Taiwan, R.O.C.
{bobby, viola, william, hoho}@iis.sinica.edu.tw

[#]Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan, R.O.C.
{d5526006, schsu, ktw}@csie.ntu.edu.tw

ABSTRACT

The prevalent ADSL service facilitates simultaneous use of telephone and high-speed data communication services. It is one of the best choices in current network technology for local loops. In this paper, we present solutions to design an interactive true video-on-demand (ITVOD) system for ADSL services with scarce resources at the set-top box. In the proposed system, we not only provide interactive functions (including fast-forward and fast-backward schemes for seeking video), but also furnish an innovative playing mechanism of the video-based program menu pages. The proposed mechanisms can save a large amount of system resources required beyond the capability of the original OSD (on-screen-display) device at the set-top box. To achieve a reliable service with scarce resources at the set-top box and variant qualities on ADSL links, we also implement an effective and efficient error control mechanism to tightly couple the packet retransmission and rate control mechanisms. The proposed mechanisms can also be deployed on hand-held devices (such as cellular phone and PDA) to support the multimedia networking services with scarce system resources.

Keywords: VOD, ITVOD, OSD, ADSL, MPEG, Error Control, Rate Control.

1. INTRODUCTION

The prevalent ADSL[1] service, which facilitates the simultaneous use of normal telephone and high-speed data communication services, is one of the best choices in current network technology for local loops. In these years, a free ADSL-based VOD (video-on-demand) trial has been deployed to 400 users in the city of Taipei by Chunghwa Telecom Co. to provide multimedia services for home users. The basic architecture of ADSL service is shown in Figure 1. At each household, a set-top box and an ATU-R (ADSL Terminal Unit-Remote) are installed. An ATU-R is a connector from POTS (Plain Old Telephone Service) to LAN devices such as set-top box and personal computer.

[1] ADSL provides downstream speeds of 1.5 to 8Mbps and upstream speeds of 16 to 640 kbps, depending upon line length and conditions.

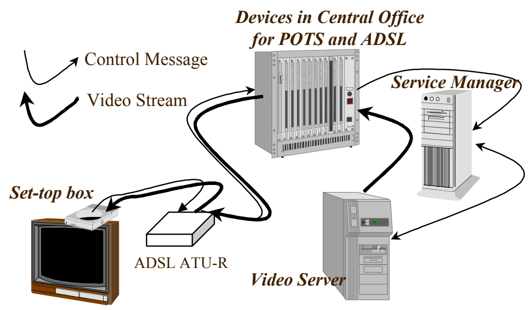

Figure 1. The transmission model of True VOD system

VOD is known as one of the most important applications for providing informative and entertaining data streams to remote users. In this ADSL trial, an interactive true VOD (ITVOD) system is deployed for the user-friendly control of real-time video services. Different from the batch service model provided by a near VOD (NVOD) system [1] [9], in an ITVOD system, the server will immediately deliver the video stream after receiving a user request. If the same content is requested by another user, a second copy of the video stream starts to deliver with no wait time. To completely provide the ITVOD service, interactive functions such as volume control (includes mute), fast-forward/fast-backward and program menu pages are important for system implementation. The volume control function has been accomplished by the software interface of the decoder provided at the set-top box. However, to implement VCR-like fast-forward and fast-backward operations, the video server must be involved. Although they have been studied in literatures for many years [3], the conventional approaches usually cost extra bandwidth and extra decoding resources. They are not feasible for the ADSL VOD system, in which the network bandwidth is limited and the set-top box has low speed CPU. In this paper, we focus on designing economical and effective schemes for fast-forward and fast-backward subject to network bandwidth and CPU power constraints.

While a user starts to use the VOD service, his first view is the program menu pages. In this paper, we try to play the program menu pages with a friendly interface for providing the indices of contents. In the conventional system, all font images (over 700KB), background images (200KB per-image) and menu text strings of the program menu pages are downloaded from the program server after the start-up of set-top box. Via the function calls provided by the OSD (on-screen-display) chip, the set-top box needs to construct images of the menu strings and insert these images into the background image to display the menu page. Notably, as there are so scarce resources and so many program menu pages needed to be presented (for the diversified menu strings and background images), it is not possible to preserve enough resources for pre-downloading and making up these menu pages at the set-top box. In this paper, we design a new playing mechanism of program menu to provide users not only a high-quality image/video-based menu page but also a low-cost requirement of the set-top box. Our mechanism constructs the menu pages at the server site. Then, we make use of the original functions of video playing and pausing in the decoder to furnish the interactive program menu playing. It costs no additional computation and storage resources at the set-top box. Although the video-based menu pages may consume more network bandwidth, it is acceptable for the ADSL services that provide a dedicated channel to each user.

In the Section 2, an overview of the ITVOD system is introduced. The interactive functions for fast-forward and fast-backward control and the playing mechanism for program menu are presented in Section 3. Under bounded buffer constraints, an error control mechanism furnished by packet retransmission and rate control to provide reliable QoS is presented in Section 4. The performance analysis and experimental results are presented in Section 5. Section 6 shows the conclusion remarks and future works.

2. AN OVERVIEW OF THE ITVOD SYSTEM

Our ITVOD system can be simply divided into two subsystems: the set-top box subsystem and the VOD sever subsystem. The set-top box is a supplemental piece of television equipment. It can either compensate for tuning deficiency or add capability like interactive device to a television receiver or a VCR recorder. In the set-top box subsystem, these basic function modules are resident. In the VOD server subsystem, besides streaming requested video data, the service requests and responses from remote users via ADSL link are served. More detailed specifications and functions of these two subsystems are described as follows.

2.1 Set-top box subsystem

The set-top box utilized in this ADSL trial contains three hardware components: CPU board (Intel 486 DX4-100), MPEG-1 [10] video decoder and 10Mbps Ethernet adapter. When a user issues an interactive control message from his remote controller in hand, the Ethernet adapter on the set-top box subsystem will send out this message via the ADSL ATU-R to the server (as shown in Figure 1). The interactive control messages include initial service request message, fast-forward and fast-backward operations, and the interactive operations for program menus. After receiving the messages from users, the servers will feedback suitable control signals and video streams to users. On the CPU board at the set-top box, software modules are executed to receive video streams and the control signals on the packet headers. Once a video stream is received, the software module will initiate the decoding command that invokes the MPEG-1 video decoder to decode the video stream. In the set-top box subsystem, the MPEG-1 decoder is equipped with preliminary application-level software interfaces. Any applications can invoke the provided function calls to feed the media streams directly to the decoder and to display the decoded images. Note that the later incoming media streams fed into the decoder should be started at the PACK header boundaries (as shown in Figure 2) while the interactive operation is issued for new video stream. Otherwise, the media streams starting with improper header boundaries may cause decoder failure. We need to utilize the provided preliminary software interfaces to implement sophisticated functions for furnishing the interactive operations in the application.

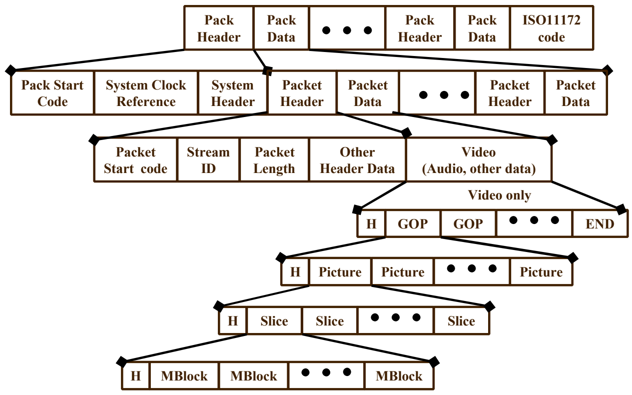

Figure 2. MPEG-1 System Stream Hierarchy for video only

2.2 VOD server subsystem

The VOD server subsystem running on Windows NT server 4.0 deliver the video streams when users request for the video service. In the server, the MPEG-1 files are stored on large volume and high performance hard disks. SCSI hard disks are adopted for their steady I/O throughputs. Totally, K+1 Ethernet cards are installed on each one of our video servers. The first K network adapters carry the video streams toward the remote set-top box. Each of them is hard-wired to a broadband communication device connecting 4 ADSL ATU-Cs in central office. Therefore, one video server may simultaneously support K*4 users. The remaining one of the K+1 network cards is dedicated as the communication path connecting the service manager for handling the interactive control messages from the set-top box (as shown in Figure 1). In the VOD server subsystem, the service manager plays an important role for system scalability. For example, while the (K*4+1)-th user is coming, the service manager can easily forward the request control to another video server upon its administrative records for the previous K*4 users on-line. In each video server, the threading mechanism is applied to improve its computation performance. Every incoming request will be served efficiently by a spawned thread to deliver the video content. The threading mechanism will save a lot of the overhead of context switch when CPU switches to another task.

While a user requests a video service, firstly the set-top box subsystem sends its request with its own identity number to the service manger. Then, the service manger will initiate the respective data channel referred to the identity number and forward the control message to one of the video servers in the system. The assigned video server will check the predefined address table to find out the corresponding multicast group address referred to the identity number of the set-top box. Finally, it will start the multicast transmission with the obtained multicast IP address as the destination to deliver the MPEG-1 stream. Note that, every set-top box has its own identical multicast IP to join and then to receive the stream requested. The network adapter will replace the low-order 23 bits of the reserved MAC address with the low-order 23 bits of the multicast address while the server sends out the requested stream in multicast transmission. However, the number of multicast groups adopted in the system is less than the number of 23 bits may represent. In the central office, the real destination address for the requested stream to the corresponding ADSL link can be efficiently resolved in the data-link layer by the modified MAC address and not have to send to higher layers to perform filtering.

3. INTERACTIVE FUNCTIONS

The interactive functions of the fast operations and program menu playing functions needs co-operation with VOD server and requires more sophisticated implementation to accomplish because of the limited bandwidth on ADSL and the scarce system resources on the set-top box. These two complicated interactive functions are respectively presented in the following subsection 3.1 and 3.2.

3.1 Economical Fast-forward and Fast-backward Functions

To support VCR-like interactive controls including fast-forward and fast-backward is one of the most difficult part in implementing an ITVOD system. They have been studied in literature for many years [3]. In the method proposed by DSM-CC [4], bandwidth and decoding speed of the decoder will be doubled (3Mbps for a 1.5Mbps MPEG-1 stream) to implement the fast-forward and fast-backward operations. It is not applicable for the ADSL trial that provides only a dedicated link with a less 3Mbps bandwidth. Other research results [3] propose methods like extra bit sequence method, skipping method and header control method to deploy playing fast-forward and fast backward. Most of them have deficiency in wasting bandwidth (e.g. CPU and network), worse playing quality, additional large amount of storage and big requantizing load at server. In [3], they propose easier control in fast operations by inserting index data in MPEG stream with addition in very small fraction to the bit-rate (i.e. 0.0064%). Though the modified MPEG stream shows compatibility when playing normally, it may give much more burden for set-top box subsystem to parse the incoming MPEG bit streams in advance before the MPEG hardware decoder starts to decode while playing fast-forward or fast-backward.

For the scarce resources of the set-top box and the limited bandwidth of the dedicated ADSL links, we implement the economical fast operations. User can skip playing pictures and jump forward to another scene or backward to review the video stream. Our proposed method for fast operations is quite similar with skipping method [3] in decoding data from the first picture of next GOP for fast-forward or previous GOP for fast-backward in the stored MPEG-1 files. While ITVOD server receives request of fast operations from users, it will stop the job of reading and sending the MPEG-1 file immediately, and move the current file index to position of the PACK header with next or previous GOP (as shown in Figure 2) and continue to read and deliver the file. For the indices of PACK headers with GOP were already built for each stored MPEG-1 file by the application tool before ITVOD system starts to run, the atomic seek-and-read file access for fast operations can be initiated without delay in the ITVOD server. Then set-top box will decode the data from the first picture of the next or previous GOP in the MPEG-1 stream and continue to play. Our implementation of the fast operations neither costs additional large resources on set-top box nor consume additional bandwidth on ADSL. Thus our economical fast-forward and fast-backward mechanism can provide users the fast operations in low latency and lower cost of system resources (comparison table is shown on Table 1).

Methods / Resources	in DSM-CC	in [3]	in ASIS ITVOD
Memory (*m*)	<=3*m*	*m*	*m*
Bandwidth	3*1.5Mbps	(1+0.000064) *1.5Mbps	1.5Mbps
CPU (*c*)	3*c*	>*c* and <3*c*	*c*
Playing quality	*Intact*	*A little bit worse*	*Not available*

Table 1. Comparison Table for fast operations on Set-top box

3.2 Interactive Program Menu Playing Mechanism without OSD

The set-top box can not provide as many resources as possible to preserve large size of menu pages needed by the VOD service to render users various and diversified program menu pages. Therefore, the following presented innovative interactive program menu mechanism provides not only the high-quality images but also the extensibility to the interactive menu pages with background video for users.

Behind the scene of the ITVOD system, we used software tools to build the menu into an MPEG-1 file by a sequence of bitmap files of menu pages. Besides, the character strings of program menu has already been inserted over the bitmap for each menu page by the tool. Each GOP in the

MPEG-1 menu file represents the starting point of the pictures for a respective menu page. If the MPEG-1 menu file is played out normally, the picture of current menu page will be switched to another in *t* seconds (as shown in Figure 3). The MPEG-1 menu file will be decoded and paused to display corresponding menu page for user's interactive menu operations from the set-top box. For example, user can scroll the highlighted indexing bar in menu page via pressing the up or down button of the remote controller. Besides, user can display the next or previous menu page via pressing the page-down or page-up button of remote controller. For the set-top box has already preserved buffer resources to decode the MPEG-1 requested video files, this innovative mechanism will cost no additional resources on the set-top box.

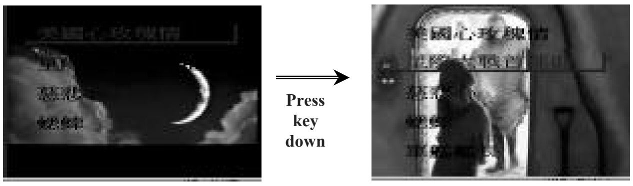

Figure 3. Playing program menu page (i.e. it appears that user pressing the "down" key)

The atomic play-and-pause operation is issued at the set-top box when the current menu page needs to be switched to another one. The play operation can help decoder to drain out the picture of the previous menu page in the decoding buffer and continue to play the next menu page until the pause operation issued. Therefore, the value of *t* should be large enough to help eliminating the possibility of displaying wrong menu page from the play-and-pause operation for the variable video frame sizes in MPEG-1. In Appendix A, we will present the algorithm of this program menu playing mechanism in more details. Then we will describe the performance analysis of the program menu playing mechanism in the section 5-1.

Considering the extension to deploy the interactive program menu playing mechanism with video background, we can merge the character strings of menu pages with the corresponding pictures in the background videos or the program previews to create the MPEG menu file. Then, the pause and play operation could be omitted at set-top box. However, video server now needs to continue to deliver the video streams of the same menu page in a loop. While user press another interactive key for program menu, server will leave the current loop and search the file index pointing to the starting position of the other menu page in the videos and continue to read in a new loop and to deliver them to the set-top box.

4. ERROR CONTROL

For the reliable transmission of compressed multimedia data on the ADSL links of variant qualities, we implement retransmission and rate control mechanisms to furnish the effective and efficient error control for the true VOD systems with bounded buffers on set-top boxes. The error control mechanism is also accomplished by two atomic functions: retransmission and rate control. Retransmission mechanism will send out the retransmission request due to the packet lost if the retransmitted packet can be restored to the buffer in time by the delay estimation from buffer occupancy. The rate control timely sends out the request for adjusting the sending rate of ITVOD server by the instance of buffer occupancy of set-top box to prevent the possibility of buffer underflow and overflow (as shown in Figure 4) while the decoder continuously decodes the data in the buffer queue.

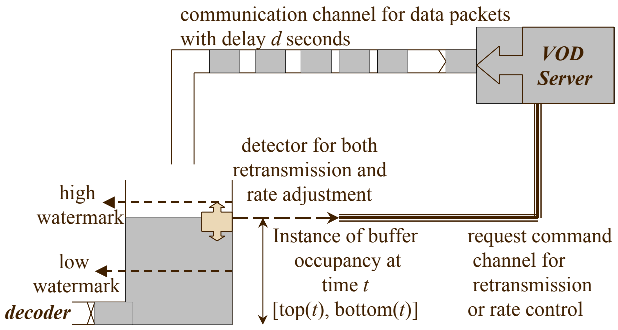

Figure 4. Error control model for the True VOD system

We execute the performance test cases of the error control mechanism by the simulation tool dummynet [7]. The dummynet tool can approximate the networking model to simulate the effects of finite queue in router, limitation of bandwidth, communication delay and the drop rate of data packets. These parameters all effect the QoS of the networking multimedia applications. And we will examine the implementation of the proposed error control by elaborating the values of the parameters controlled by the dummynet simulation tool for the ADSL model. The performance results of error control are listed in section 5-2. The details of retransmission and rate control in error control are described in the following sections 4.1 and 4.2:

4.1 Retransmission:

In the retransmission mechanism, we assume the decoding rate is 1.5Mbps, the size of a buffer unit is 1024 bytes and totally we have *s* buffer units. Besides, we define that a retransmitted packet is valid only if the retransmitted packet can be recovered to decode without losing the original order sent by the server. For example, while the sequence number *i* of the incoming packet header is larger than the sequence number *j* of packet header of the previous one by 2 or more, there are $j - i - 1$ packets are lost[2] during the transmission between server and client. Then, we will stuff the current arrival packet with sequence number *i* into the tail of bounded buffer $j - i$ times and check if the outgoing retransmissions for the $j - i - 1$ lost packets will be valid or not (as shown in Figure 5).

The possibility of valid retransmissions will be indicated by the size *b* of the current running buffer occupancy. Since the decoder will continuously decode to remove the data from the bottom of the buffer, we must verify if the

[2] On the dedicated ADSL link, we simply assume that no packets will arrive out of order.

retransmitted packets will be put back to the buffer in time before the packet with sequence number j has been decoded. After the communication delay u seconds for the retransmitted packet, the decoder would consume approximately $u/0.005$ (i.e. 8Kbits/1.5Mbps $\cong$ 0.005 second) buffer units. In Figure 5, b is 8 buffer units at time t. We assume the communication delay is longer than 25 milli-seconds (i.e. u > 25ms). So the first retransmission packet with sequence number j – 1 won't be valid to be recovered at time $u+t$. To check the possibility of valid retransmission before we start to send the request of retransmission to the server will save the bandwidth cost for the unnecessary retransmissions. For the running round trip delay on the communication link can approximate the consuming buffer units at clients for the valid retransmission, the retransmission validation threshold is timely estimated to reset by the periodically probed round trip delay. The pseudo code for packet retransmission and its validation is listed in Appendix B-1.

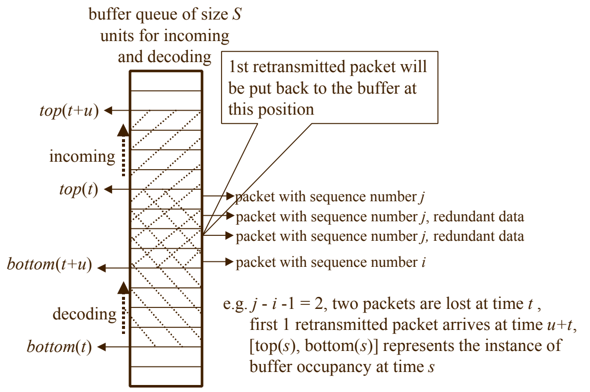

Figure 5. Example of retransmission model for the True VOD system

4.2 Rate Control:

The rate control mechanism of error control will play a key role to help to raise the possibility of valid retransmission mentioned in the previous paragraph. The proposed rate control mechanism will control the sending rate from server by the feedback of command request from client side. While the size of buffer occupancy (i.e. [$top(t)$, $bottom(t)$]) is running above the high watermark threshold or under the low watermark threshold, the set-top box will send out the command request for the sending rate adjustment on VOD server (as shown in Figure 4). The non-stop non-decreasing packet loss rate could cause the buffer occupancy always running under the low watermark. In the mean time, the corresponding request for the increasing sending rate will push VOD server to spend more and more bandwidth to deliver the data stream. However, the limited communication bandwidth on the ADSL may not sustain the traffic loading and the side effect of the packet loss will get worse. To preventing the chain reaction of the consuming bandwidth for the sending rate of VOD server exceeding the actual bandwidth of the communication link for a long time period, we must define a maximum sending rate R_{max} of server to limit the sending rate. The pseudo codes of rate control for set-top box and VOD server are shown in Appendix B-2.

5. PERFORMANCE ANALYSIS

5.1 Delay bound of interactive operations

Let the size of control message be S_c. The bandwidths of ADSL upstream and downstream are B_u and B_d, respectively. As the transmission model shown in Figure 1, the ITVOD server will receive the interactive control messages including fast and program menu operations from the user in S_c/B_u seconds. After VOD server receives these interactive control messages, the interactive mechanisms presented in the section 3 for fast operations (i.e. fast-forward/fast-backward) and program menu all need to seek the file to a new starting position to continue to deliver the MPEG-1 file. Denoted T_s as the average seek time in the hard disk and B_c as the transmission bandwidth of the hard disk controller. The server will start immediately to send out the media stream after seeking to the media file in T_s seconds and reading the first block of data of size S_b in the hard disk in S_b/B_c seconds. If the VOD server has to serve N users for reading the data block, the total disk service time is $N*(T_s + S_b/B_c)$ seconds. After the size of media stream received reaches M Kbytes (it takes M/B_d seconds), the set-top box will be able to display the video for the interactive operations of fast operations and program menu. Notably, as the average response time of external events in Windows NT [6] is less than tens of microseconds, we can omit the delay of threading task switching. Therefore, a user will get the requested service for interactive operations roughly in ($S_c/B_u + N*(T_s + S_b/B_c) + M/B_d$) seconds.

CPU & MEMORY	**NETWORK CARDS**	**STORAGE DEVICES**
1. Intel Pentium III 550MHz	1. Intel 82557-base 10/100Mbps Ethernet Adapter.	1. Adaptec AHA-2940U2W Controller
2. 512M SDRAM	2. D-Link DFE-570TX, Quad Channel Server Card (4 ports), 10/100Mbps	2. IBM 18G, 10,000 RPM Hard Disk with average seek time 4.9ms

Table 2. Specification of main hardware components in a VOD server

For example, in our ITVOD system, S_c is 60 bytes of raw data on the link, B_u is 16Kbps, B_d is 1.5Mbps, S_b is 512 Kbytes and M is up to 50 Kbytes. We also know that the value of M ranges from Kbytes to tens of Kbytes for the variable frame sizes in MPEG streams. Referring to Table 2, B_c is 80MBps (Bytes per second) for the wide ultra SCSI hard disk controller and T_s is 0.0049 second for the hard disk applied. It was mentioned in section 2.2 that totally up to K*4 users can be served for one ITVOD server. In the worse case, when a user sends its interactive control request and at the same time there are other K*4-1 users simultaneously request file I/O, N will be equal to 16 if we expect the delay of interactive operation is less than 0.5 second. Therefore, in the Table 2, there are four (i.e K = 4) network cards installed to deliver the streams for delay constrains of the interactive operations. The average estimated delay for the formula (S_c/B_u + $\frac{1}{16}\sum_{N=1}^{16} N(T_s + S_b / B_c)$ + M/B_d) in this example is 0.4 second and it's very close to most of the observed delays in the ADSL trial. Although B_d will vary according to the signal-to-noise ratio [5] for the length and condition of the ADSL link, it usually won't affect the delay of the interactive functions for fast operations and program menu playing too much.

Besides, for the interactive menu, the total delay estimated can help us to measure the values of *p* and *t* in advance. The playing time *p* of the play-pause operation should be preset to ($S_c/B_u + N$*$(T_s + S_b/B_c) + M/B_d$) for the required QoS. The preset value of *t* (as mentioned in section 3.3) between the consequent menu pages in the menu video file should be relatively larger than *p* while building the MPEG program menu video file. It also may depend on the playing time of commercials or program previews given by the VOD service provider if the background of the menu pages is video.

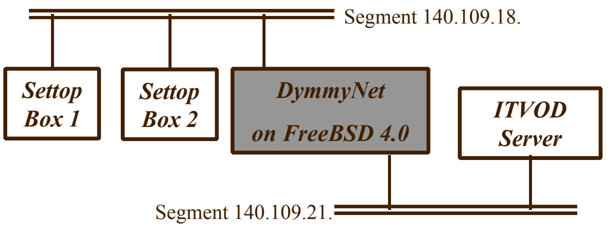

Figure 6. Testbed for error control

5.2 Performance of Error Control

Our testbed for error control is shown on Figure 6. The testing video is a 30-minutes long MPEG-1 video from Public TV Company in Taiwan. The file size *L* is 291,873,836 bytes. The buffer size *s* at the set-top box is 200 data units and each unit is 1024 bytes. In this sub-section, we raise two test cases to validate the performance under different network circumstances simulated by the dummynet tool [7]. Set-top box 1 perform first test case with no packet retransmission and the rate control in which the low watermark is *s*/8 and the high watermark is *s*/2. Set-top box 2 performs the second test case with the aggressive watermarks: the low mark is *s*/2 and the high watermark is 7*s*/8. Besides, the retransmission threshold is dynamic in the second test case and it is timely reset by the round trip delay that is periodically probed and can approximate the buffer units consumed after valid retransmission. Table 3 shows the testing results for the numbers of total packets loss and valid retransmission under the packet drop rates (i.e. *d* = 0.001, 0.01) and different network delay (from 0ms to 500ms) controlled by the dummynet. The maximum sending rate R_{max} of VOD server is defined as 183024bytes/s measured from the 1.5Mbps bandwidth of the ADSL downstream link and the bandwidth overhead (i.e. 76bytes) of the data packet and the protocol headers including TCP/IP and Ethernet headers.

The performance result for the implementation of error control presents that more than 50% of the lost packets can be recovered at variant network delays for the rational packet drop rate 0.001. While the packet drop rate is tenfold increased to 0.01, the result shows very poor valid retransmission rate. However, the unlisted experimental results show that the valid retransmission ratio can be highly improved if altering the simulation setting to provide more bandwidth for retransmission packets. Higher packet drop ratio will need more bandwidth to accomplish the valid retransmission. If the transmission link can not sustain the retransmission loading, the valid retransmission ratio will nosedive and it surely demonstrates the side effect mentioned in section 4.2.

Delay *d*	Drop Rate = 0.001				Drop Rate = 0.01			
	Test case 1		Test case 2		Test case 1		Test case 2	
	Lost Packets	Valid Retrans.	Lost Packets	Valid Retrans.	Lost Packets	Valid Retrans.	Lost Packets	Valid Retrans.
0ms	314	0	391	258	5125	0	5104	29
50ms	335	0	425	280	5201	0	5174	19
100ms	296	0	359	245	5079	0	5087	72
200ms	267	0	505	254	4896	0	5261	0
500ms	273	0	554	277	5115	0	5080	2

Table 3. Testing results for the error control

6. CONCLUSIONS

In this paper, we present the implementation of an ITVOD system on ADSL with scarce resources at the set-top box. The proposed mechanism for the fast-forward and fast-backward gives economical and effective solution in the cost of bandwidth and storage for the ITVOD system in the ADSL trail as compared to other methods proposed in literature. The mechanism proposed for interactive program menu playing can provide menu background pages with not only images but also videos of the commercials or content previews. It is beyond the capability of the original OSD device at the set-top box, and can provide high-quality and diversified program menu pages for multimedia services. As ADSL has provided a dedicated broadband channel, the delay of interactive service for fast operations and program menu playing is bounded (less than 500 ms in our experiments). The QoS obtained is acceptable for providing real-world applications. Additionally, due to the tightly coupled packet retransmission and rate control mechanisms, our system is cost-effective in error control and can perform reliable QoS for transmitting multimedia streams over an unstable communication link. As these proposed mechanisms cost no additional resources on the client, they can be also deployed on hand-held devices (such as cellular phone and PDA) to support the services of networking multimedia with scarce system resources.

ACKNOWLEDGEMENT:

The authors would like to thank the helps from Dr. Chin-Ming Chen and Mr. Yin-Hwa Huang, the project manager and a researcher of Chunghwa Telecom. Co. to deploy our proposed ITVOD server in the ADSL trial of Taipei City.

REFERENCE:

[1] Chih-Shoung Huang; Wen-Deh Wang; Liang-Wei Lee; Yinyi Lin; Huang-Cheng Huang; Yin-Hwa Huang; Chang-Chi Lai; Bing-Shan Chien; Yu-Chyuan Chang; Chi-Shi Liu; Bor-Sheng Jeng, "Design of video service engine with SCSI for an ADSL-based near video on demand system", IEEE Transactions on Consumer Electronics, Vol.43 #3 , pp. 386 –393, Aug. 1997.

[2] Liang-Wei Lee; Yeou-Chyuan Chang; Yin-Hwa Huang; Sing I.S. Young; Huang-Cheng Huang; Wen-Deh Wang; Chien-Hsing Wu; Jei-Shoung Huang; Bor-Shenn Jeng, "A PC-Based Video Server for True VOD and Near VOD Systems", Proceedings of the second Workshop on Real-time and Media Systems, pp. 335-340, 1996.

[3] Doh Sang-Yoon; Min Jang, "Fast forward and fast rewind play system based on the MPEG system stream with new concept", IEEE International Conference on Multimedia Computing and Systems, Vol.2, pp. 846 – 850, 1999.

[4] ISO/IEC JTC1/SC29/WG11 N1100, "Digital Storage Media Command & Control", Nov. 1995.

[5] Al Jackson, "ADSL for high-speed broadband data service", Aerospace Conference, IEEE, 1998.

[6] Brian Catlin, "Design of a TCP/IP Router using Windows NT", the 1995 Digital Communications Design Conference.

[7] Luigi Rizzo, "Dummynet: a simple approach to the evaluation of network protocols", Computer Communication Review, Vol.27#1, pp. 31-41, 1997.

[8] Luigi Rizzo, "Effective erasure coded for reliable computer communication protocols", Computer Communication Review, Vol.27#2, pp. 24-36, 1997.

[9] Chi-Chien Hsueh; Wen-Cheung Cheng; Chi-Shi Liu, "An Efficient Near-Video-on-Demand System for Broadband Residential ADSL Network", Proceedings of IEEE International Conference on Multimedia and Expo, 2000.

[10] Jaon L. Mitchell, William B.Pennebaker, Chad E. Fogg and Didier J. LeGall, "MPEG Video Compression Standard", International Thomas Publishing, 1996.

APPENDIX A.

Pseudo Codes for Interactive program menu playing method:

VOD server

Declaration:

Key : "up", "down", "page up", "page down", "enter" and "quit" keys from remote controller.

GOP_i : GOPs' locations of the MPEG-1 menu file indexed by integer *i*.

V$File_k$: the character string of full path and name of the requested video file indexed by *k*.

$Action_{key,j}$: integer action list indexed by the received *key* and integer *j* for interactive operations from user, =0: quit; >0: index *i* to GOP location; <0: index *k* to Vfile.

Program

i = 1. /* *i represents the index of menu pages in order* */

if the received control is the initial request of service {

spawn the following threading task.

read the MPEG-1 menu file and send the video stream packets to the set-top box.

do{

wait until controls arrived.

assign received control to the variable *key.*

if ($Action_{key,i}$ > 0) {

j = $Action_{key,i}$.

seek the current reading index of the MPEG-1 menu file to the position GOP_j .

}

else if ($Action_{key,i}$ <0){

k = 0 - $Action_{key,i}$.

/* *to be positive index* */

begin to delivering the video content file indicated by $VFile_k$ after immediately terminating transmission of MPEG-1 menu file.

} /* *else if* */

}*while* ($Action_{key,i}$!=0). /* *do-while* */

/* *Action list should be carefully initialized in advance* */

leave the threading task.

} /* *if* */

Set-top Box

Declaration:

p : the period between the atomic play and pause operation.

Program

send the initial request of ITVOD service and wait for MPEG-1 menu page.

if the MPEG-1 menu file is received at the first time{

decoded to play the MPEG-1 video stream for less than *p* seconds and pause immediately.

/* *paused at first page of program menu* */

}

do{

wait until user press interactive request control keys from remote controller.

send out the request control.

if the key is "up", "down","page up"or"page down"{

wait until the next page coming and continue to play for *p* sec and pause immediately.

/* *paused at the next page of program menu* */

} /* *if* */

else if the pressed key is "enter"{

wait until the normal video content stream comes and continue decoding to play without pausing.

/* *playing the video content stream* */

} /* *else if* */

}*while* (the pressed key is not "quit").

leave the ITVOD service.

/* *after receiving "quit" key* */

APPENDIX B.

Pseudo Codes for Error Control:

B-1. Retransmission Pseudo Codes:

Set-top Box

if (*current_sequence_no* != *last_recv_no* + *1*){

/*detect packet lost */

stuffing the current packet (*current_sequence_no* - *last_recv_no -1*) times into the buffer.

if (the [*current_sequence_no* - *last_recv_no -1*] lost packets can be completed to retransmit in time before the buffer occupancy running underflow){

send out the retransmission request for (*current_sequence_no* -*last_recv_no* - *1*) lost packets.

}

}

B-2. Rate Control Pseudo Codes:

Set-top Box

if (*current_sequence_no* - *last_cmd_seq*) > *N* {

/* i.e. *N* = 1000 */

last_cmd_seq = *current_sequence_no*

if (the buffer occupancy is less than the low watermark)

send request for increasing sending rate.

else if (the buffer occupancy is larger than high watermark) {

send request for decreasing sending rate.

}

}

VOD server

if (receiving request for increasing sending rate){
if (current rate is smaller than rate R_{max})
add a unit to current rate
else
deduct the current rate by a unit
}
if (receiving request for decreasing sending rate)
deduct the current rate by a unit