Chin-Yu Chen

Error Control for Video on Demand Over ATM Network

Key Terms:

- Asynchronous Transfer Mode (ATM)
- Video on Demand (VOD)
- MPEG
- Reed-Solomon Forward Error Correction (RS FEC)
- Header Error Correction (HEC)
- Signal-to-Noise Ratio (SNR)
- Transmission Band Width
- Semaphore Locks

Author

Chin-Yu Davis Chen

Chin-Yu Davis Chen originally gained interest in his project after attending a lecture on Advanced Computer Networks. He values the uniqueness of this project as well as the quality of innovation demanded of him. This project, which focuses on the control and concealment of error in video signals, has also been published in the *Third Annual Multimedia Technology and Applications Conference* (MTAC). Davis suggests that finding a professor whom one likes is the key to a successful research project. ◇

NEXT

Abstract

Video On Demand (VOD) is one of the most promising future multimedia services that allow users to view video programs at a desired time. VOD may be realized using emerging high-speed network infrastructures based on Asynchronous Transfer Mode (ATM) technology. However, due to the novelty of ATM, there is a lack of experience on transport of multimedia traffic. In this paper, ATM performance in transporting compressed video (MPEG) is evaluated through experiment and simulation. A prototype VOD application is developed to investigate design issues such as rate control and error control. Network errors causing data corruption may degrade the quality of the video severely. This paper undertakes a comparison of the two error control schemes, Reed-Solomon Forward Error Correction (RS FEC) and Header Error Concealment (HEC). The metric for the quality measure of the video is the comparison of the signal-to-noise ratio (SNR) of the resulting video to the original video.◇

BACK NEXT

Faculty Mentor

Working with Chin-Yu has been a very rewarding experience for me. Chin-Yu attacked an important research topic of how to provide effective error control for video on demand (VOD) services on an ATM (Asynchronous Transfer Mode) high speed network. In VOD services, it is critical to provide effective error control so that video streams are transmitted to a number of users on a real time basis. Chin-Yu empirically evaluated various error control schemes. The research findings of the project will deepen our knowledge on multimedia high speed networks.◇

BACK NEXT

Tatsuya Suda
Department of Information
and Computer Science

Chin-Yu Chen - Error Control For Video On Demand ... [1] [2] [3] [4] [5] [6]

Back to Journal 1998 Index

Introduction

Video On Demand (VOD) is one of the most promising future multimedia services that allow users to view video programs at a desired time. Video programs are stored in a server with a compression format (such as MPEG[1]) to reduce the necessary disk space and transmission bandwidth. Upon the request of a user, the video server delivers through a network the requested video program. Users are equipped with a TV set-top box that provides interactive operations such as pause and stop, slow and fast forward, and rewind. A good VOD system should handle startup and interactive requests by having low latency, requiring small buffering capability at the user's TV set-top box, and supporting a large number of concurrent users with guaranteed Quality Of Service (QOS).[2]

VOD can be realized using emerging broadcast network infrastructures based on Asynchronous Transfer Mode (ATM) technology.[3] ATM networks provide high speed transport and QOS guarantee, attending the requirements of a good VOD system. However, due to ATM novelty, there is a lack of experience on transport of multimedia traffic. Standardization bodies are still attempting to understand how ATM performs outside the theoretical realm. There are many factors that may not have been properly accounted for in theoretical performance analysis; such factors may only be thoroughly investigated through network experiments and simulations.

In this paper, ATM performance is evaluated through experiment and simulation. A prototype application that distributes MPEG video over an ATM network is developed to investigate the suitability of ATM technology for transporting compressed video as MPEG. This prototype VOD application is based on a video server where the video programs are stored as an MPEG, and users can request and play the video programs in real-time. This prototype is used to investigate design issues such as rate control and error control. In addition, performance measures such as throughput and frame rate are presented to demonstrate the suitability of ATM technology for transporting such multimedia traffic. The performance measures presented in this paper will enhance knowledge of the behaviors, limitations, and possible improvements of ATM network in supporting multimedia applications.

Design

architecture consists of several modules that handle various tasks of the concurrent server. One module listens to the network connection and creates a new thread for each client request received. Another module's responsibility is to copy chunks of the MPEG file from the local disk resident at the server side into a buffer and parse the buffer into MPEG frames. Several of these parsed frames are then loaded into memory; there, they wait for the appropriate transmission time.

There are two reasons to have a number of frames ready and waiting in memory. The first involves a situation where it is time for the server to transmit, but the frames are still in the process of being parsed. The server pauses and waits until the frame is completely parsed to transmit, creating an uneven flow of data to the client. Significant consecutive transmission pauses from the server will create an underflow of data to the decoder at the client. The result is a possible degradation of video quality or the termination of the MPEG decoder altogether. The second reason involves the kernel scheduler. Since the server is concurrent, multiple clients can view different videos simultaneously. This means multiple threads are reading data from a disk, leaving the scheduler to decide which thread's instruction to handle at any one time. Since there is a fixed amount of time to retrieve data from a disk, some of the threads will be blocked until the disk drive is available. This blocking creates a pause in the frame parsing process and delays the server transmission time. Again, if there are no parsed frames ready in memory, the server transmission time may be delayed, which leads to data underflow.

The client architecture consists of four separate buffers and three distinct processes: the producer, the consumer, and the MPEG play process. The producer process receives packets from the server and copies the packet payload into one of three buffers: buf1, buf2, and buf3. The goal of the producer is to keep buf1, buf2, and buf3 filled throughout the packet transmission. Semaphore locks are placed on buf1, buf2, and buf3 to avoid the race condition problem. The consumer reads from one of the three buffers and fills the fourth buffer (temp_file) in packet sequential order. The fourth buffer (temp_file) is the actual MPEG bit stream from the server. The MPEG play process executes the MPEG decoder to decode

The design of the prototype VOD application is based on the client/server architecture, focused to provide rate and error control. The architecture consists of a concurrent server and one or more clients. The server

temp_file. The fourth buffer (temp_file) is necessary because the decoder must always have data to process. Of course, the decoder assumes the end of the MPEG file is reached and terminates otherwise.

BACK NEXT

Page 2

Chin-Yu Chen - Error Control For Video On Demand ... [1] [2] [3] [4] [5] [6]

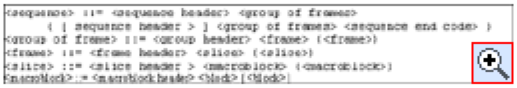
Back to Journal 1998 Index

Rate Control

Since the design involves synchronization between the server and its multiple clients, rate control plays a major issue. To ensure synchronization, a Timer layer, which resides above both the server and the client layer, is implemented. The Timer has two primary duties. One duty is to ensure the proper timing of the server in sending packets to the client. This is implemented by setting a clock resident at the server and transmitting the parsed frame to the client at each clock tick. The clock is implemented by measuring the elapsed time between the last packet sent to the client and the time the next packet is ready to be sent. Each packet that is sent to the client passes through the Transmitter, which passes the packet on to the client. The clock reads the current time when a new packet is sent to the Transmitter and reads the time again when the Transmitter acknowledges that it is ready for the next packet transmission. The time of the next packet transmission is determined by subtracting the latter time from the former time.

The second duty of the Timer is to ensure that neither the server nor the client does indefinite waiting. This is implemented by setting two alarms, one for the server and the other for the client, used for a time-out purpose. The alarms will guarantee that if the server crashes, the client will eventually time-out instead of waiting endlessly for packets from the server. Similarly, if the client crashes, the server will time-out instead of waiting endlessly for client acknowledgment.

To obtain control over the transmission rate, and to avoid overflow or underflow, one must understand the structure of the MPEG. A full-motion video consists of a set of pictures to be displayed sequentially. These pictures (frames) are MPEG-compressed, producing bit streams that represent the pictures as well as control information for the MPEG decoder. The bit streams are arranged into hierarchical structures from frames, down to individual pixels. One reason for such arrangement is to provide random access to the video sequence. The following BNF notation describes the MPEG bit stream[1]:



```
<sequence> ::= <sequence header> <group of frames>
      { [ sequence header > ] <group of frames> <sequence end code> }
<group of frame> ::= <group header> <frame> {<frame>}
<frame> ::= <frame header> <slice> {<slice>}
<slice> ::= <slice header > <macroblock> {<macroblock>}
<macroblock> ::= <macroblock header> <block> {<block>}
```

Control information is stored in the header of each group and provides information about the specific group needed by the decoder. All group headers start with a unique 32-bit code to distinguish one group from another. Thus a frame header start code will be different from the slice header start code. Fields in the MPEG sequence header also indicate the frame rate in which the video program should be displayed. Therefore, based on the MPEG structure, transmission rate can be controlled using any of the MPEG sub-structures (i.e. frame, slide, and macroblock) as a rate control unit. These structures could allow varying levels of coarse to fine granularity. Fine grain control would require a large processing overhead to parse the MPEG bit stream. In the experiment, MPEG files are parsed according to the frame header start code and sent to the client one frame at a time. It is uncertain that sending frame by frame is optimal in comparison to sending by slice or by macroblock. Further experimentation is necessary. However, sending by frame instead of by slice or by macroblock will reduce the overhead processing time at the client side since a frame is much larger than a slice and macroblock. To clarify, the client need not process as many packet headers. Shorter overhead processing time is favored because the VOD system must satisfy a real-time constraint of 30 frames/s. Furthermore, with a shorter overhead processing time, a single server can handle more clients.

Error Control

Network conditions are not always optimal. Random network errors and network burst errors (i.e. due to congestion) may degrade the quality of the transmitted MPEG video stream severely and to an unacceptable level. Error control methods that alleviate this situation are achieved either by correcting the errors or by concealing the errors. Error control methods can be categorized into two main categories: forward error correction (FEC) and selective retransmission. Selective retransmission, however, is not feasible due to the possible network latency delay in the retransmission, which violates the requirement of a good VOD system. Because the VOD system is a real-time application, selective retransmission methods may lead to rate control complexity and cause problems such as data under-flow. In either case, the retransmitted data may arrive too late to be of any use. Thus, FEC methods are favored for real-time applications such as VOD.

For example, the slice header provides slice information such as the position of the slice in a frame as well as the quantifier scale.

Two forward error correction methods, the Reed-Solomon Forward Error Correction (RS FEC) and the Header Error Concealment (HEC), are implemented and investigated. Reed-Solomon FEC encodes each packet with RS codes on the server side before transmission, decodes the packets, corrects errors within range, and maintains data integrity on the client side. Reed-Solomon codes are non-binary codes capable of

BACK NEXT

Page 3

Chin-Yu Chen - Error Control For Video On Demand ... [1] [2] [3] [4] [5] [6]

Back to Journal 1998 Index

correcting errors by appending redundancies, especially if they occur in bursts. The motivation of RS FEC method is in its ability to maintain data integrity without retransmission and its ability to handle network burst errors. There are various parameters that may be adjusted when implementing RS codes to obtain different levels of error correction. The number of bits used to represent a data symbol may be changed to focus on the correction of different distributions of errors (such as scattered or in burst). The number of data symbols in each encoded block may also be changed to vary the size of burst errors it can handle. Finally, the symbol block size may be changed in relation to the ratio of the number of errors it can handle to the amount of overhead redundancies imposed. Reed-Solomon codes of 4-, 8-, and 16-bit symbols were implemented and experimented with to reveal their actual error correction ability and their feasibility to provide error control for VOD application.

The second method, HEC, focuses on concealing the errors in the MPEG sequence headers and on maintaining video stream's format integrity. Researchers have observed that if the error resides on the headers, the video will play with incorrect frame rate and order. If excessive error resides on the headers, the MPEG decoder terminates the decoding process, causing the MPEG player to quit prematurely. The HEC error control method is implemented as follows. On the server side, each packet is parsed to separate out the sequence headers from the image data. The sequence headers are then encoded with RS codes before transmission. The client decodes to correct errors in the sequence headers and then reinserts the headers back into the image data in the proper position. The advantage of this method is that it imposes a low overhead on the data transmission while maintaining the integrity of the video format. Recall that maintaining the integrity of the video format ensures the success of playing the whole video with correct frame rate throughout. However, the tradeoff is that the quality of the image is not guaranteed.

Results

A 2.5 Mb MPEG file was used as the requested video program throughout the experiment. The video server rate controls the transmission at 4 Mb/s, satisfying the real-time playing constraint of 30 frames/s. It was observed that as the number of clients increased, the average throughput decreased, since a single server has to serve an increasing number of clients. As throughput decreased, the frame rate at which the server provided also decreased because such application requires a large amount of CPU processing. Thus, the number of concurrent clients a single server can handle is limited by the CPU's processing power. The measurements showed that an UltraSparc 2, UltraSparc 1, and Sparc 5 could only support up to 18, 10, and 5 concurrent clients, respectively, while maintaining the transmission rate at 30 frames/s.

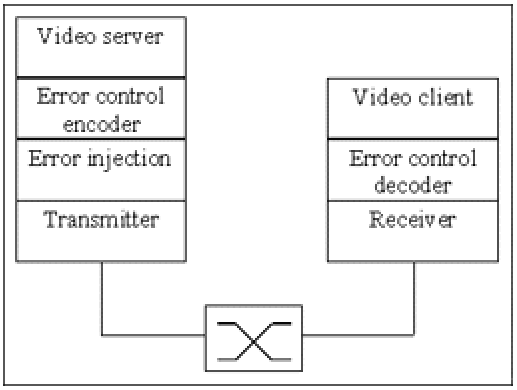


Figure 1
Simulation Model for Error Control Experiment

Random network errors and network burst errors (i.e. example, due to congestion) were simulated for the error control experiments. Random network error rates from 10^{-8} to 10^{-2} (in units of bits) were simulated using a random number generator. Each experiment used an error rate with a constant burst error size in the range of 1 to 10 bits. The errors were simulated and inserted into the packet on the server side before transmission, but after the error control decoder. On the client side, the error control decoder processed the packet when it was first received, before any processing by the client. Figure 1 shows the simulation model used for error control experiments.

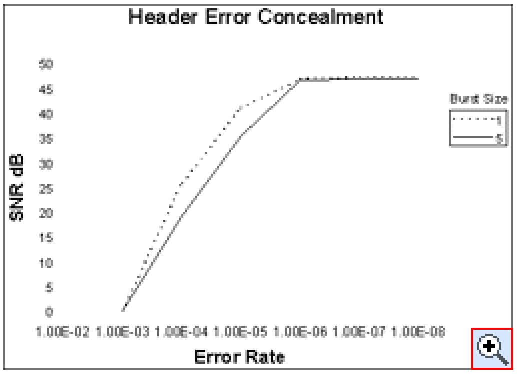


Figure 2
Error Rate and SNR with no error control.

BACK NEXT

Page 4

Chin-Yu Chen - Error Control For Video On Demand ... [1] [2] [3] [4] [5] [6]

Back to Journal 1998 Index

The error control experiment comprised of five measurements, with the measurement of no error control being the control case of the experiment. The metric for measuring the quality of the video is the comparison of the signal-to-noise (SNR) ratio of the resulting video to the original video. The measurements were no error control (the control case), HEC, and 4-, 8-, and 16-bit symbol RS FEC. Figures 2 through 4 show the results of the experiments for each error control method with error rate as a function of the SNR. Higher SNR ratio means better image quality. A SNR ratio of 47 is a perfect duplicate of the original video, and an SNR of 0 indicates a premature termination of MPEG player due to decoder error.

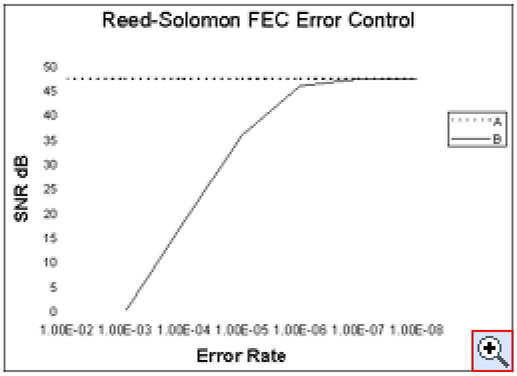


Figure 3
Error Rate and SNR Using RS FEC.

Generally the quality of the video decreases as error rate and/or burst error size increases. It is observed from the experiment that an error rate less than or equal to 10^{-6}, with small burst size (less than 10 bits), yields nearly perfect image quality. Visually, errors are not perceived. However, this is subjective. With error rates greater than 10^{-6}, video quality starts to degrade severely without error control. Especially with a burst error size larger than or equal to 5-bit, the MPEG decoder encounters errors and thereby terminates, causing the MPEG player to quit prematurely. Furthermore, the data does not show that with an error rate greater than 10^{-6} and a small burst error size, the video will not be played with the correct frame rate without error control. It pauses every now and then and usually skips frames. Thus error control must be provided to guaranteed quality of service.

The RS FEC error control method provided superb error control. It corrected all errors under all experimented conditions, except the 4-bit symbol RS FEC when the burst error size was larger than 10 bits. The video quality degraded slightly in this case. It was found that a larger code symbol size of RS code is favored to handle larger burst size error; a smaller code symbol size is favored to handle scattered bit errors. Furthermore, by increasing the number of redundancies appended, the number of burst errors it can handle also increases. However, this is a tradeoff, which causes a larger overhead processing time in the encoding and decoding process. Despite the superb error correction ability of RS FEC, it imposed a large overhead redundancy and overhead processing time. The redundancies imposed by the 4-, 8-, and 16-bit symbol RS FEC were 25, 14.4, and 32 percent, respectively. The imposed redundancy by the 16-bit symbol RS FEC was unusually large because its block size is relatively large compared to the packet size; therefore extra padding was required. Large overhead redundancy and overhead processing time resulted in lower throughput and higher latency delay. In either case, the number of concurrent clients a single server can handle decreased. The performance of the system was decreased to maintain relatively perfect image quality.

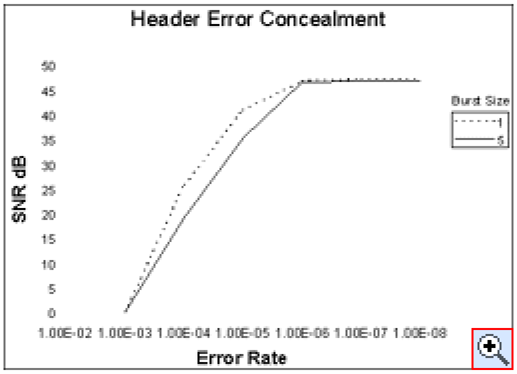


Figure 4
Error Rate and SNR using HEC.

The HEC error control method provided a relatively perfect image quality with error rate less than or equal to 10^{-6}, an acceptable image quality with error rate of 10^{-5}, and ensured the continuing playing of the video with error rate of 10^{-4}. The overhead redundancy was very small compared to the RS FEC error control method, 1.34 percent. However, the overhead processing time is about the same as with RS FEC because of the parsing and the rearrangement of sequence headers. The software implementations for both proposed error control methods are not practical because of the large overhead processing time imposed. Thus hardware implementation is recommended for such tasks.

BACK NEXT

Page 5

Chin-Yu Chen - Error Control For Video On Demand ... [1] [2] [3] [4] [5] [6]

Back to Journal 1998 Index

Conclusion

In this paper, a prototype VOD application that distributes MPEG video over ATM network was developed to investigate design issues such as rate control and error control. Performance measurements, such as the achieved frame rate and the number of concurrent clients a single server can handle, indicate that the ATM networks can effectively transport compressed video (MPEG) for VOD application. Experiments also show that random network errors and network burst errors (i.e. due to congestion) degrade the video quality severely and to an unacceptable level. Thus error control must be provided. The two proposed error control methods�RS FEC and HEC�have their own advantages and disadvantages. The RS FEC error control method maintained perfect image quality, but imposed a large overhead redundancy. HEC had a small overhead redundancy and maintained the integrity of video format. However, in the latter image quality was not guaranteed. A hybrid error control method involving both is probably optimal. The VOD application may start by providing the high overhead redundancy RS FEC error control and switch to HEC error control as the number of concurrent clients handled increases. Recall that fewer concurrent clients can be supported if the overhead redundancy is large.

Possible future work could be done to implement and verify the performance of the hybrid error control method described above. Also, researchers could investigate further the error control issues for VOD application. Another possible future endeavor might be to investigate further the rate control issues to provide interactive operations such as pause and stop, slow and fast forward, and rewind.

Endnotes

[1] ISO/IEC JTC 1/SC 29. *Revised Text for ITU-T Recommendation H.262* (ISO/IEC: Japan, 1995).

[2] S. Sengodan and V. Li, "A Quasi-Static Retrival Scheme for Interative VOD Servers." In *IEEE IC3N 96* (Washington DC, October 1996).

[3] R. Handel, M. Huber, and S. Schroder, *ATM Networks� Conepts, Protocols Applications* (New York: Addison-Wesley, 1995).

Works Cited

Hagenauer, Joachim. "Rate-Compatible Punctured Convolutional Codes (RCPC codes) and Their Applications." *IEEE Transactions on Communica-tions* 36:4 (1998): 389-400.

Han, T., and L. Orozco-Barbosa. "Performance Requirements for the Transport of MPEG Video Streams over ATM Networks." *Proc. of IEEE ICC 95 Conference* 1 (1995): 221-5.

Karlsson, G. "Asynchronous Transfer of Video." *IEEE Communications Magazine* 34:8 (1996): 118-26.

Karn, Phil. *Reed-Solomon Coding/Decoding Package*. Vers. 1.0. September 1996.

Lin, S., and D. J. Costello. *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice Hall, 1983.

Noronha, C., and F. Jia. "Latency and Error Control in Live MPEG Video Communication over Computer Networks." *Computer Communication* 20 (1997).

Oliveira, C., J. Kongkitisupchai, S. Gaffney, J. Kim, and T. Suda. "Experiments on Transport of Compressed Video (MPEG) over ATM Network." *Proc. of Multimedia Technology and Applications 97 Conference, Session 5*, Paper 2 (1997).

Peterson, W., and D. Brown. "Cyclic Codes for Error Detection." *Proceedings of The IRE Conference* 1 (1961): 228-235.

Rasheed, Y., and A. Leon-Garcia. "AAL1 with FEC for the Transport of CBR MPEG2 Video over ATM Networks." *Proc. of IEEE Infocom 96 Conference* 2 (1996): 529-36.

Raychaudhuri, D., H. Sun, and R. Girons. "ATM Transport and Cell-Loss Concealment Techniques for MPEG Video." *Proc. of IEEE International Conference on Acoustics* 1 (1993): 117-20.

Stevens. W. R. *UNIX Network Programming*. Upper Saddle River, NJ: Prentice-Hall Inc., 1994.

Todd, H., and J. Meditch. "Encapsulation Protocols for MPEG Video in ATM Networks." *Proc. of IEEE Infocom 96 Conference* 3 (1996): 1072-9.

Fore Systems. *ATM Programming Reference Manual*. Pittsburg: Fore Systems, 1994.

BACK

Page 6

Chin-Yu Chen - Error Control For Video On Demand ... [1] [2] [3] [4] [5] [6]

Back to Journal 1998 Index