SDL MODELED HYBRID ERROR CONTROL SCHEME FOR RELIABLE MULTICAST OVER INTERNET

Bo Rong, Alain Servais, Maria Bennani, Ahmed Elhakeem*, and Michel Kadoch
Department of Electrical Engineering, Ecole de technologie superieure, Universite du Quebec
Montreal, Quebec, Canada, H3C 1K3
**Department of Electrical and Computer Engineering, Concordia University*
Montreal, Quebec, Canada, H3G 1M8

Abstract

With the development of IP multicast and the Internet multicast backbone (Mbone), audio/video conferencing applications become more and more popular. These real-time multicast applications usually have stringent QoS requirements. As one of the most challenging tasks in QoS guaranteed multicast network, the technology of controlling packet loss has attracted more and more attention. In this paper, we introduce a new hybrid error control scheme that integrates interleaving, forward error correction (FEC), and threshold based automatic repeat request (threshold based ARQ) to mitigate the error and loss effects encountered in wire and wireless networks. In particular, the threshold based ARQ is studied carefully to shorten the transmission delay in reliable multicast. Moreover, in order to simplify the description and validation of our scheme, specification and description language (SDL) is used for modeling it. SDL depicts different levels of our scheme, from a broad overview down to detailed designs. In this paper, after devising the basic algorithms of hybrid error control, the corresponding implementations are developed to achieve practical performance of our scheme, and theoretical analysis is employed to explain the experimental results.

Keywords: *Reliable multicast; hybrid error control; threshold based ARQ.*

1. INTRODUCTION

Nowadays, multicast is becoming more and more important to real-time multimedia communication applications, such as interactive video conferencing, video on demand system, and so on. These applications usually have stringent QoS requirements including bandwidth, end-to-end delay, delay jitter and packet loss rate. As one of the most challenging tasks in multicast, the technology of controlling packet loss has attracted more and more attention [1][2][3][4][5][6]. In core networks, congestion, tunneling, and lack of high speed backbones are some reasons leading to occasional data loss. On the wireless side, bursty channel errors combined with packet loss make stronger error and loss control more crucial. Numerous techniques have been suggested for error and loss control over the Internet. In the past, usually mere packet replication was suggested to combat the best effort delivery and subsequent TPDU (transport layer PDU) loss of the Internet, but now this gives way to the powerful erasure codes. These erasure codes, such as Reed Solomon (RS) Codes, have strong inherent erasure correction capability [7].

Different from previous work of employing forward error correction (FEC) and automatic repeat request (ARQ) in multicast [8][9], in this paper, we propose a threshold based ARQ algorithm to shorten the transmission delay. In addition, an actual buildup integrating all the techniques, such as interleaving, FEC, and threshold based ARQ, have been implemented, and the performance of this hybrid error control scheme are achieved by experiments. The experimental results show that our hybrid error control scheme can provide strong multicast error control and excellent QoS guarantee at high user data rate.

The rest of the paper is organized as follows. Firstly, we introduce our hybrid error control scheme in section 2. Then the experimental results are demonstrated and analyzed in section 3. Finally, section 4 summarizes our results.

2. OUR SDL MODELED HYBRID ERROR CONTROL SCHEME

2.1 SDL Modeling

We use specification and description language (SDL) as a formal technique to model our hybrid error control scheme. SDL is an object-oriented formal language

CCECE 2004- CCGEI 2004, Niagara Falls, May/mai 2004
0-7803-8253-6/04/$17.00 ©2004 IEEE

standardized by ITU-T, and it has been developed for modeling telecommunication systems including data communication network. An SDL system contains four main hierachical levels, e.g. system, blocks, processes, and procedures. These four levels give a good discription of the system, from a broad overview down to detailed designs. Although SDL is not intended to be an implementation language, automatic translation of SDL information to a programming language is available.

Our scheme is based on a Client/Server architecture, and the main entities of the new technique are the server and client algorithms which reside on top of the Internet UDP layer. The server routine typically exists at the sender of video/audio multicast or one of the intermediate routers (called domain receiver or DR in multicast terminology). The client routine exists at the end user or the receiver part of DR. To give an outline of our hybrid error control scheme by using SDL, Fig.1 shows the system model. In the rest of paper, SDL is continuously used to describe our scheme.

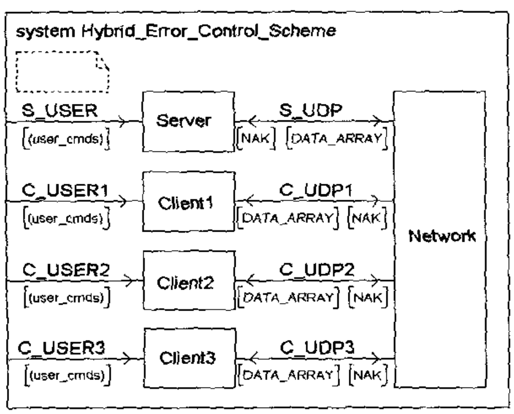

Fig.1. The system model of our hybrid error control scheme

2.2 TPDU Format Used in Our Scheme

The format of frames sent from the server to the clients is shown in Fig.2. From this figure, we can know clearly how the application data are encapsulated within the frame. Each block of data, 218 bytes, is given a sequence number (NO) in the range from 0 to 65535. A one byte data length (DL) field is also added to indicate the end of receiving because the data may be less than 218 bytes. If the data are less than 218 bytes, the remainder of the 218 bytes is filled with padding. A two byte Cyclic Redundancy Check (CRC) field is appended at the end of the word, and this CRC operates over the whole word. The resulting 223 bytes are fed to the (255,223) Reed Solomon encoder routine [1] which outputs a corresponding 255 bytes that we call an encoded word. The RS code is assumed to be systematic, in which case the 32 parity bytes are added by the encoder at the end of the word. A one byte of dummy data is added to obtain a 256 bytes encoded word.

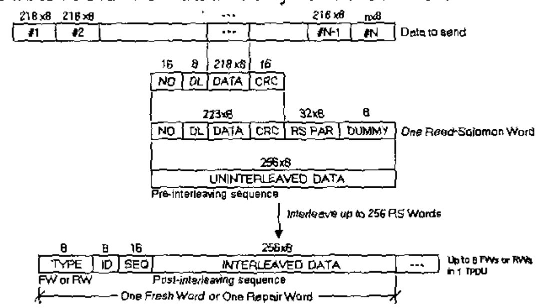

Fig. 2. TPDU format

Interleaving is next applied to a record of 256 such encoded RS words to yield a corresponding interleaved record. The first interleaved word of such record consists of the first byte of each pre-interleaving word. The 256th interleaved word is the concatenation of the last byte of each pre-interleaving word. It is also possible to have a smaller interleaving depth such as 128, in which case the first interleaved word will encompass the first byte and the second byte of each pre-interleaving word. The first 128 bytes of the 128th interleaved word are bytes numbered 255 in the original pre-interleaving words, and the second 128 bytes of the 128th interleaved word are bytes numbered 256 in the original pre-interleaving words.

If the interleaving depth is 256, it is easy to know that up to 32 consecutive interleaved words can be lost without any retransmission request from the receiver. Due to interleaving, if any 32 words are lost, only 32 out of 255 bytes in each corresponding pre-interleaving word would be lost. Because RS decoder can conceal up to 255-223= 32 lost bytes, such loss is not detrimental and will not result in retransmission requests. For multicast error concealment over the Internet, minimizing the need of retransmissions is an essential asset which subsequently leads to minimizing NAK and repair packet implosions [2]. As shown in Fig.2, the server routine groups up 8 interleaved words into one UDP data unit (TPDU). Upon request of the client, retransmitted words are also included in some TPDUs. A one byte type filed (TYPE) is given to each TPDU to indicate this UDP packet carries fresh words or retransmitted words (11111111 for fresh words and 10101010 for retransmitted words). Furthermore, a one byte ID filed is also added to denote the position of this TPDU inside the interleaved record. Although each original RS encoded word has a sequence number field (NO), this information is scrambled by the interleaving afterward. Therefore, a new sequence number field (SEQ) is added to each TPDU. From the above discussion we know, if an interleaving depth of 256 is used, each TPDU contains 8 interleaved words, then 4 consecutive TPDUs can be lost without any retransmission.

2.3 Threshold based AR

In order to shorten the transmission delay in reliable multicast, we propose a threshold based AR algorithm to improve the performance of traditional AR algorithm. Video/audio communication applications usually don t need 100 reliable data transmission, because users cannot feel an obvious quality decrease if only a small number of packets are in error or lost. Instead, these applications strongly require the packet loss rate in transmission must be lower than a certain threshold X_{app} . To accelerate the data transmission with this sort of packet loss requirement, the threshold based AR algorithm is devised. Briefly, our threshold based AR algorithm can be explained as follows. During the period when the packet loss rate in the network is less than X_{app} , no AR is employed. On the other hand, when the packet loss rate in the network is higher than X_{app} , AR must be employed to mitigate some packet loss so that the real packet loss rate of video/audio applications is guaranteed to be under the threshold X_{app} .

To implement this algorithm, once the client detects the n' lost packet at time $t_{oss}\ n$, it will calculate the size of data correctly received from time $t_{oss}\ n$ to time $t_{oss}\ n$. We define the size of the n' lost packet as $S_{oss}\ n$, and the size of data correctly received from time $t_{oss}\ n$ to time $t_{oss}\ n$ as $S_{rec}\ n\ \ ,n$. In general, $S_{oss}\ n$ is a constant which has been known by the client beforehand. If $S_{oss}\ n\ /S_{rec}\ n\ \ ,n < X_{app}$, no AR is needed; otherwise, AR mechanism must be employed to retransmit the lost packet. Usually, any error control scheme has unwanted residual errors. Therefore, to prevent residual errors from damaging the real packet loss rate our scheme can guarantee, a new parameter X_{ARQ} ($X_{ARQ} < X_{app}$) is often chosen to replace X_{app} in practice. By this way, the packet loss threshold X_{app} required by video/audio applications can be really met.

The format of No Acknowledgment (NAK) packet in our scheme is shown in Fig.3, and it is the only packet type that the client can send. However, for video conferencing system, data may also be sent from the client to the server. In Fig.3, we use 01010101 as the value of TYPE field to denote NAK messages. A one byte field DL denoting the length of NAK message is also added. Moreover, each packet is given a sequence number (SE) in the range from 0 to 65535. After this, the subsequent field ID LIST gives ID information of the lost words that need to be retransmitted from the server to the client. At last, one byte CRC field is added for error checking at the server side.

The algorithms running on the server side and client side are outlined by FSMs shown in Fig.4 and Fig.5 respectively.

8	8	16	8xN	8
TYPE	DL	SEQ	ID_LIST	CRC

NAK

Fig. 3. NAK packet format

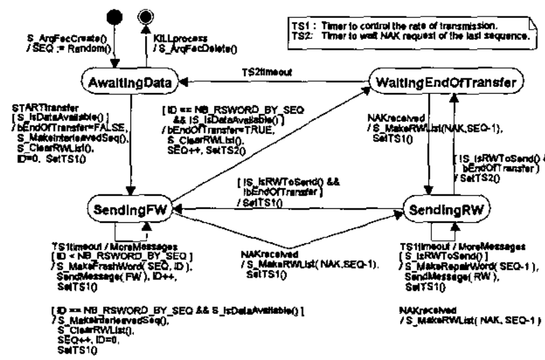

Fig. 4. Server FSM

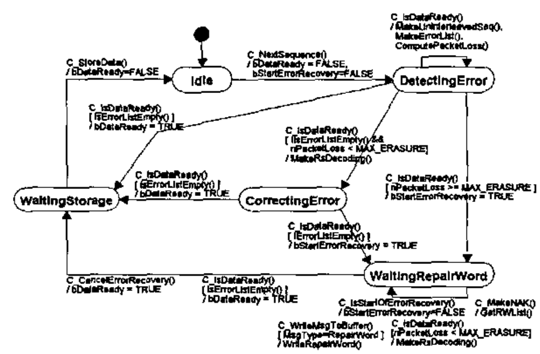

Fig. 5. Client FSM

2.4 Algorithms on Server and Client

In Fig.6, the process model of sending fresh word and repair word at server side is shown clearly. Usually, the whole video/audio file is FEC encoded, interleaved, and stored before starting the real-time multicast. However, the procedure of sending repair words has to be executed in real time after the multicast session commences.

At client side, after a record of data is read from the UDP layer, blocks of 256 bytes are sampled from it. The algorithm for processing fresh words is shown in Fig.7, from which we can know that all error control techniques including interleaving, FEC, and threshold based AR are utilized to conquer the packet loss. The algorithm for processing repair words is similar to Fig.7, and in our scheme the retransmission times of one lost packet is limited to 2.

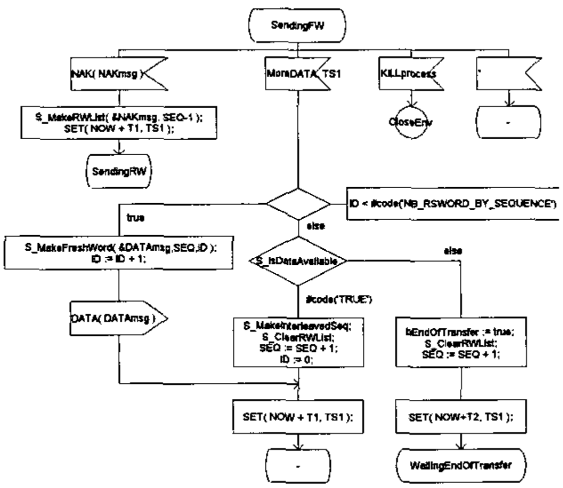

Fig. 6. Process model of sending fresh word and repair word at server side

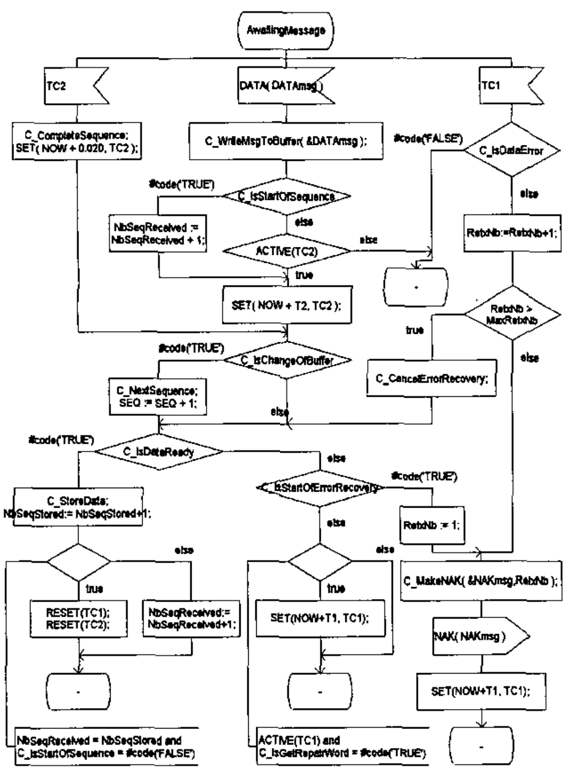

Fig.7. Process model of receiving fresh words and sending NAK packets at client side (TC1 and TC2 are two timers used in the algorithm)

3. EXPERIMENTAL RESULTS

Fig.8 shows a simple outline of our experimental environment. Although there are 7 PCs involved in the experiment to build up multicast, only two PCs (Pentium II 600Mhz with 256MB memory), one as server and another as client, are mainly addressed in Fig.8. All PCs are connected by 100Base-T Ethernet virtual LANs, while Linux real-time platform is running on them to evaluate the performance of our hybrid error control scheme. The IP Wave Network Impairment Emulator (IP Wave) on Windows NT platform is configured as the default router between different virtual LANs to resemble the error and loss effects between server and clients. The IP Wave can simulate effects that may be encountered by any application communicating across a large network such as the Internet. The IP Wave includes a comprehensive set of applicable, real-world impairments such as packet delay, duplication, reordering, fragmentation, loss, and errors that can affect IP packets coming from distant locations or across distributed networks. In our experiment, the IP Wave is mainly employed to achieve various packet loss and error conditions without having to set up distributed endpoints geographically for testing.

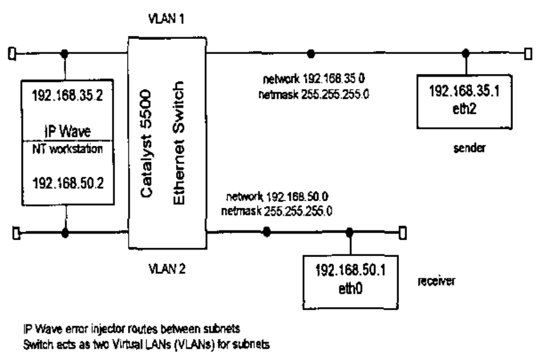

Fig.8. Block diagram of our experimental testbed

Many tests have been conducted for different packet loss and byte errors. Fig.9 to Fig.12 shows the main results obtained from our experiment, which reflect the average outcome at all clients in our multicast testbed. From these figures, the performances of different error control schemes including P-AR , P-AR /RS/Int, and Th-AR /RS/Int (our scheme), are compared carefully. The acronyms and parameter setup used in our experiment are listed as follows.

Acronyms in ig. to ig.

P-AR : Pure AR ; Th-AR : Threshold based AR ;

RS: Reed Solomon Codes; Int: Interleaving.

Parameter set p in ig. to ig.

Data transmission rate: 1.5Mbps;

Byte errors in the network: 5 ;

Packet loss rate: 0 ~16 in Fig.9, 0 ~50 in Fig.10~Fig.12;
Words in one TPDU: 8;
Interleaving depth: 256, if interleaving is involved;
Limit on retransmission times: 2, if P-AR or Th-AR is involved;
X_{ARQ} : 8 , if Th-AR is involved;
X_{app} : 12 , if Th-AR is involved.

Fig.9 shows the percentage of lost packets corrected by RS decoder in all packet loss. Because pure AR doesn t use RS coding, its value keeps 0. As to P-AR /RS/Int and Th-AR /RS/Int, when packet loss rate is low, RS decoder can counteract most packet loss and byte errors. However, under heavy packet loss condition, the RS decoder becomes impotent to conceal all the packet loss and byte errors, so P-AR or Th-AR must be employed. From Fig.10, we can see that the number of NAK messages generated by our Th-AR /RS/Int scheme is less than that of others, due to the capability of threshold based AR and RS codes. Fig.11 shows the percentage of residue errors in all transmitted data. Although P-AR and P-AR /RS/Int have better performance than our Th-AR /RS/Int scheme in terms of residue errors, our scheme still can promise a packet loss rate lower than certain threshold X_{app} . Fig.12 demonstrates the transmission delays caused by different error control schemes. Obviously, our scheme has the shortest delay, which makes it more suitable for real-time multimedia communications.

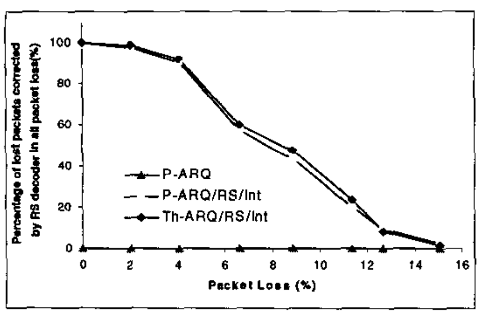

Fig.9. Percentage of lost packets corrected by RS decoder in all packet loss VS. packet loss rate in multicast network

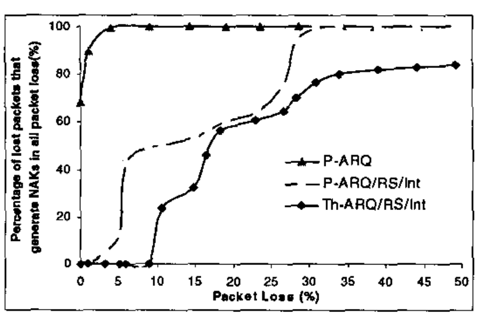

Fig.10. Percentage of lost packets that generate NAKs in all packet loss VS. packet loss rate in multicast network

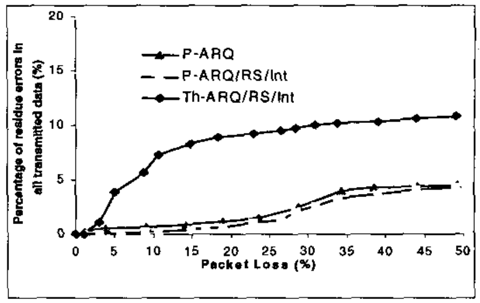

Fig.11. Percentage of residue errors in all transmitted data VS. packet loss rate in multicast network

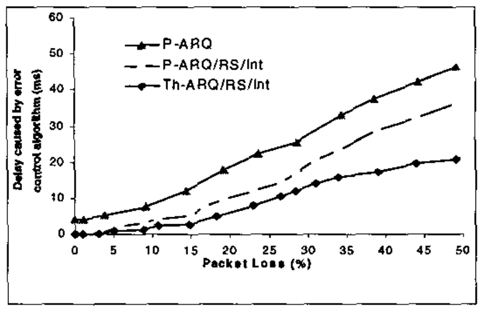

Fig.12. Delay caused by error control algorithm VS. packet loss rate in multicast network

4. CONCLUSIONS

In this paper, a hybrid error control scheme is presented to provide good oS experimental results when applied to multicast transmissions. While some theoretical results have been recently documented for miscellaneous techniques of reliable multicast, our laboratory work shows the practical feasibility of integrating all these techniques. According to the results achieved by our testbed, one can multicast video steam over the Internet with short transmission delay in the condition of heavy loss and errors. Our experimental results indicate that the receivers can be guaranteed a packet loss rate lower than certain threshold value while operating at reasonable data rate, i.e. 1.5Mbps video transmission. Currently, the experiments are being extended to teleconferencing applications.

References

[1] S. B. Vicker, *ERR R TR L SYSTEMS R IGITAL MM I ATI S A ST RAGE.* Prentice-Hall, 1995.

[2] J. Nonnenmacher, E. W. Biersack, and D. Towsley, "Parity-Based Loss Recovery for Reliable Multicast

Transmission," *IEEE A M Transactions on et orking*, Vol. 6, No.4, pp. 349-361, August 1998.

[3] Rizzo L. and Vicisano L., "Effective Erasure Codes for Reliable Computer Communication Protocols," *A M SIG MM omp ter omm nication Revie* , Vol.27, No.2, pp. 24-36, Apr. 1997.

[4] S. Floyd, V. Jacobson, C. Liu, S. McCanne, L. Zhang, A reliable multicast framework for light-weight sessions and application level framing, *IEEE A M Trans. on et orking*, Volume: 5, Issue: 6, pp. 784 803, Dec. 1997.

[5] Christos Pappadopoulos, Guru Parulkar and George Varghese, An Error Control Scheme For Large-Scale Multicast Applications, *I M'* , Volume: 3, pp. 1188 1196, 29 March-2 April 1998.

[6] J. Lin and S. Paul, RMTP: A Reliable Multicast Transport Protocol, *in Proceeding of t e IEEE Infocom* , pp. 1414-1424, March 1996.

[7] J.H. Jeng and T.K. Truong, "On Decoding of both Errors and Erasures of a Reed-Solomon Code using an Inverse-Free Berlekamp-Massey Algorithm," *IEEE Transactions on omm nication*, Volume: 47, Issue: 10, pp. 1488 1494, Oct. 1999.

[8] Yu J.X., Yuan Li, Murata H., Yoshida S., Hybrid-AR scheme using different TCM for retransmission, *IEEE Transactions on omm nications*, Volume: 48, Issue: 10, pp. 1609-1613, Oct. 2000.

[9] Kousa M.A., Elhakeem A.K., Yang H., Performance of ATM networks under hybrid AR /FEC error control scheme, *IEEE A M Transactions on et orking*, Volume: 7, Issue: 6, pp. 917-925, Dec. 1999.