濟州大學校 尖端技術硏究所 論文集, 第12卷, 2號, pp.166-173, 2001. 12
Cheju Nat'l Univ. Res. Inst. Adv. Tech. Jour., Vol.12, No.2, pp.166-173, 2001. 12

Design of an Error Control Scheme Capable of Masking Frame Error on Real-Time Data Processing Networks

Junghoon Lee*, Kyung-Youn Kim**

실시간 데이터 처리 네트워크에서 프레임 오차 마스킹을 위한 오차 제어 스킴의 설계

이 정 훈*·김 경 연**

ABSTRACT

This paper proposes an error control scheme for real-time data processing networks. The proposed scheme does not retransmit every erroneously transmitted frame as in the classical error control schemes but minimizes the affect of frame error by duplicated transmissions of the frame in the next period. In order to obviate the intervention of error control message to the normal message transmission, the scheme controls error only when additional network time, which is inevitably derived in the transmission-based real-time communication, is available. Based on the token passing protocol, the receiver knows when to send retransmission request by counting the number of received tokens. The simulation results executed via SMPL show that the proposed scheme is able to enhance the success ratio over the given range of error rate as well as utilization and appropriate to real-time data communication.

Key Words : electric impedance tomography, particle concentration profile, inverse problem, finite element method

I. INTRODUCTION

The goal of hard real-time communication is to guarantee that all messages will meet their deadlines during error-free operation of the Network[1]. To meet their time constraints, hard real-time messages must be properly scheduled for transmission, and scheduling messages in a multiple-access network is the function of the Medium Access Control(MAC) protocol. It is the protocol layer that arbitrates access to the network and determines what message to transmit at any given time on a single communication channel shared by multiple nodes. Existing

* 제주대학교 계산통계학과
Department of Electronic Engineering, Cheju Nat'l Univ.
** 제주대학교 전자공학과, 첨단기술연구소
Department of Electronic Engineering, Research Institute of Advanced Technology, Cheju Nat'l Univ.

approaches for multiple−access networks are classified into two categories, namely, access arbitration based and transmission control based approaches, respectively[1]. The first concentrates on determining when a node can send a message over the shared channel while the second focuses on deciding how long a node can continue to send messages.

As an example of transmission control based schemes, timed token protocol makes each node transmit its message in round−robin fashion. The protocol parameters such as TTRT(Target Token Rotation Time) and capacity vector decide the amount of time for which a node can exclusively access the network[3]. Hence, such parameter values should be properly selected to meet the time constraint of a given message stream set. Each stream has period and transmission time as main traffic parameters and its message should be transmitted within period, in other words, before the next message arrives in the stream. The parameter selection procedure, called as bandwidth allocation, decides the TTRT and capacity vector after the careful analysis of the stream set. This procedure first searches the period which has the minimum available transmission time, that is, which captures the smallest number of tokens. To meet the time constraint for a stream at any time means that the message transmission on such a period can be completed within deadline.

The periods, however, other than that has the minimum transmission time have extra bandwidth which is inevitably overallocated to meet hard real−time guarantee requirement. The original timed token protocol makes this extra network time be used for non−real−time, or asynchronous messages. As contrast, this paper is supposed to use this extra bandwidth for error control purpose for real−time messages. While the real−time protocol can meet the time constraints of messages, transmission error is inevitable which results in missing the deadline of a message. This error is originated from the error rate characteristics of underlying communication media, for example, coaxial cable or optical fiber as well as the temporary disturbance due to external stimulus. Error control functions are responsible for detecting and correcting errors that occur in the transmission of packets.

The currently existing error control schemes such as go−back−N, stop−and−wait, selective repeat and so on do not seem to be applicable to real−time communication as the error control procedure needed for acknowledgement and retransmission of a message prolongs the transmission time, resulting in completion after deadline expiration[4]. Furthermore, the acknowledge−ment messages and the retransmitted messages themselves may interfere normal messages in the shared medium network such as FDDI(Fiber Distributed Data Interface) and so on[5]. In other words, error control messages may defer the transmission of normal messages.

However, in transmission control based real−time communication, sufficient bandwidth is allocated to each stream along with the extra network time as mentioned above. Retransmission via this overallocated bandwidth does not result in missing deadline nor interferes the transmission of normal messages. Additionally, we can assume that buffer is sufficiently available since a message is valid within its deadline and should be kept only by this time, namely, during the interval for which there is possibility of retransmission.

Each application has specific error control requirement of its own. For example, data transfer such as FTP(File Transfer Protocol) cannot tolerate any message error with little or no regard to the transmission time. On the contrary, typical real−time application requires that as many messages as possible can be recovered within their deadlines, tolerating some loss of messages.

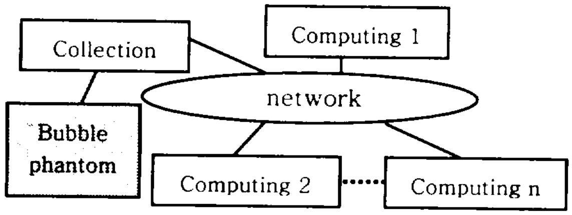


Fig. 1. ET network architecture

On another applications, sequential messages are tightly dependent with one another. In the application, loss of a message strongly needs the error-free transmission of related following messages. For example, ET (Electronic Tomography) data processing network shown in Figure 1 consists of a collection node along with one or more computing nodes[6]. The collection node periodically gathers data from the bubble phantom and controls the collection logic after the basic data analysis. Other computing nodes, having and executing respective algorithms such as Newton-Raphson, genetic and so on, receive from the collection node and exchange messages with one another for further analysis. In this example, real-time network such as FDDI which supports timed token protocol, is desirable as the messages have time constraint.

The data collected at each sampling period on the collection node forms a fixed size message, which is then segmented into FDDI frames and transmitted to the corresponding computing node as shown in Figure 2. All frames belonging to a message should be transmitted within period. Any bit error induces the loss of the entire frame, resulting in degrading the correctness of data analysis at the computing node. That is, on a frame error, a computing node executes its job with imperfect data, containing the error term in analyzing the data. Though the improvement of network performance is able to reduce the network error rate, the growth of message size increases error rate as well as the amount of data loss.

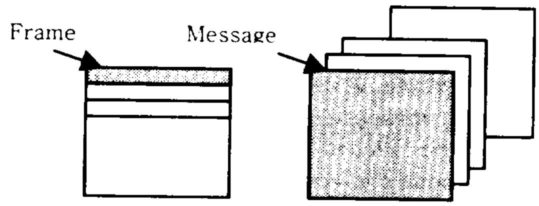


Fig. 2. Frames and Message

The data elements located in the same position or frame of the respective messages have intimate relation with one another as they report the sequential state changes of identical sensor point, called as the electrode, in the bubble phantom. When a specific frame is lost in a message, it is preferable to correctly send the frame of the next message rather than to resend the lost frame. Hence, the computing node can somehow overcome the frame error if the subsequent frames arrive correctly. In short, ET application requires that successive frame errors should be prevented across a certain range of sequential message transmissions.

This paper is consist of as follows: After issuing the problem in Chapter 1, Chapter 2 exhibits some related works on the error control scheme for real-time communication. Chapter 3 describes the proposed error masking procedure as well as the error reporting steps in detail. Then we analyze the characteristics of the proposed scheme briefly in Chapter 4. Finally, Chapter 5 summarizes and concludes this paper.

II. RELATED WORKS

Two kinds of error control schemes may be used for real-time communication, that is, temporal redundancy and spatial redundancy[1]. Temporal redundancy schemes are appropriate when the deadlines of the messages involved in a stream are large enough to permit the use of timeouts. In this scheme, when the message is actually transmitted, the receiver returns an acknowledgment to the sender. If the acknowledgment is not received within the retransmission timeout period, then duplicate of the message is transmitted. As an variation of this scheme, MARS(MAintainable Real-time System) uses temporal redundancy and avoids acknowledgment[7]. As shown in Figure 3, all the duplicates are transmitted, irrespective of whether or not the earlier duplicates suffer transmission errors.

The second scheme is based on spatial redundancy and is appropriate when the deadlines of the messages involved in a stream are too small to permit retransmission based on timeout. This scheme makes multiple copies of each message be transmitted on multiple channels[8]. In spatial redundancy and MARS, bandwidth waste is extremely high, as they always spend at least twice as much bandwidth as the original requirement whether a packet is delivered error-free or not. Beside above-mentioned schemes, error correction mechanism appears to be promising, however, information overhead and processing complexity are not negligible.

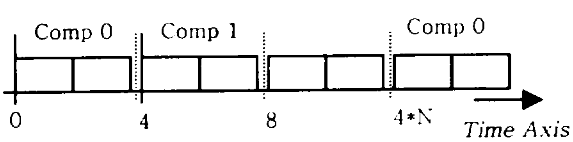


Fig. 3. MARS slot structure

In addition, as an example of error control and QoS(Quality of Service) negotiation in wireless environment, H. Bengston has proposed a protocol based on a scheme of retransmissions done on demand within a given time window[9]. Each retransmission is coded with a varying number of redundant symbols. The set of blocks used for retransmission is controlled by two QoS parameters: deadline for the transmission and the probability that the correct decoded message will reach the recipient before its deadline.

III. ERROR CONTROL SCHEME

3.1 Error Detection

As regard to error detection on the FDDI network, each of FDDI frames has FCS(Frame Check Sequence) and FS(Frame Status) fields. While FCS field contains an information based on a 32-bit CRC(Cyclic Redundancy Check) used to detect errors within the frame, FS field includes ED(Error Detection) symbol which is aiming at indicating error occurrence during frame transmission. As the FDDI frame proceeds along the stations following the ring topology, each station checks whether the frame is corrupted and sets ED when an error is detected. Since all frames return to the source station to be absorbed, source station can decide whether a frame has experienced a network error or not.

However, for the errors on the upper layers, additional explicit messages should be adopted for retransmission request or error report, which demands supplementary bandwidth allocation[10]. It is desirable to send this message via asynchronous bandwidth in order not to interfere the transmission of normal real-time messages. To construct the error report message, the receiver initializes the error frame list when the first frame arrives. If the sequence number of this frame is not 1 but k, the receiver adds the numbers from 1 to (k-1) into the list. From then, the receiver appends the erroneously received frame number. As the receiver also meets tokens and knows the total number of frames transmitted in a period, it can decide when to send error report message to the transmitter.

Figure 4 shows the example, where a message is transmitted with u frames and the receiver knows it from the connection establishment. At the initial state, a frame numbered as 3 (not 1) arrives. Then the receiver initializes and adds 1 and 2 to error list as well as sets counter to 3. The counter increases each time the receiver captures token. When the counter reaches to u, receiver sends error report to the sender only if it has asynchronous bandwidth, that is, when a token arrives earlier than TTRT. After all, error list is reported to the sender periodically as long as there is additional token time as in SNR error control scheme[11].

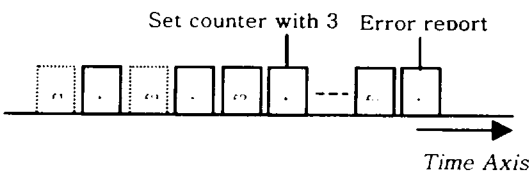


Fig. 4. Example of error reporting

3.2 Error Control

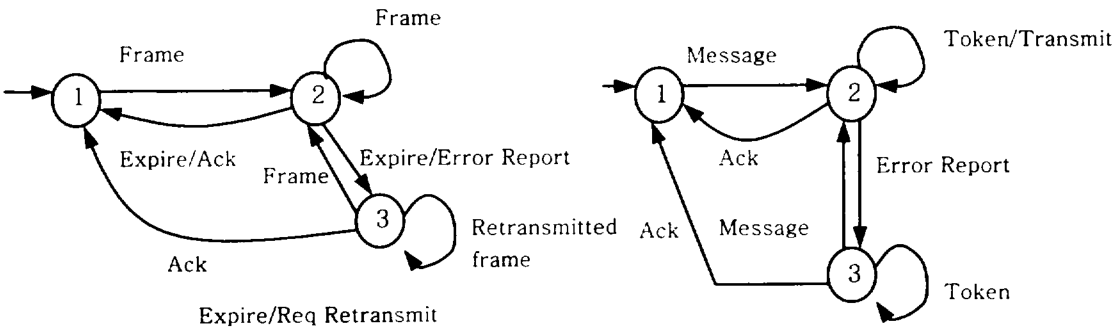
The networking community has explored a broad spectrum of solutions to deal with error control. They range from local solutions that decrease the link error rate observed by the upper layer protocols or applications to transport protocol modifications and proxies inside the network that modify the behavior of the higher layer protocols. Local error control allows users to obtain a high percentage of the available link bandwidth, even when using standard protocols as TCP in harsh error conditions. With local approach, error control overhead is paid only when needed, and designed in such a way that it does not interfere with higher layer protocols.

The receiver node should be able to reassemble the frames which has been delivered out of order due to retransmission[12]. The receiver requires reassemble buffer as large as Ci for a stream. The sender also requires buffer as large as Ci for retransmission. If a new message arrives when not all of frame errors are recovered, the sender abandons the error control procedure and overwrites the message to the buffer. By sequence numbers, we assume that the receiver can cope with the out of order delivery of frames or duplicated arrivals of the same frame.

Figure 5 shows the state diagrams of sender and receiver, respectively. As shown in Figure 5(a), when the receiver receives the first frame in the initial state (State 1), state is changed into State 2, setting the counter with the number of frame as well as initializing the retransmission list. State 2 continuously receives token or frame, updating the counter and retransmission list when needed. When the counter expires and all of frames are correctly received, state is changed to State 1. Otherwise, retransmission request message is generated and the receiver goes to State 3. The receiver sends the message only when it has asynchronous bandwidth. In State 3, the receiver waits for the retransmitted message until all of frame errors are recovered or the first frame of a new message arrives.

Figure 5(b) shows the state diagram of sender side. A new message arrival makes state transition from State 1 to State 2. In State 2, the sender transmits the frames one by one on each receipt of a token. If a Ack message arrives, the sender goes to State 1 and wait for the new message. Arrival of a retransmission request message changes the sender to State 3, where the sender retransmits the requested frame via overallocated bandwidth. If there is no additional token for a sender in a period, the sender is not able to retransmit the message.

With the error reporting procedure, the sender knows which frame has been lost during the previous period. To enhance the probability of the successful transmission of the frame at the next period, the sender



(a) receiver side (b) sender side

Fig. 5. State diagrams

duplicates the frame and sends the duplicate when it meets an additional token after transmitting all frames belonging to a message. This scheme reduces the possibility of sequential two frame errors across the periods by half. If there is no additional token for a sender in a period, the sender is not able to transmit such duplicated messages. As the average number of received tokens in a period is dependent on the network load, we can expect that more error can be overcome as the utilization of message set goes smaller.

IV. ANALYSIS

For the analysis of the proposed error control scheme, we define symbols first. By convention, a message stream S_i has period P_i and transmission time C_i. In addition, capacity vector is denoted as $\{H_i\}$. As mentioned above, the proposed scheme uses the network bandwidth remaining from real-time traffic for error control. For a time interval T, the bandwidth for error control $B(T)$ for all of nodes can be calculated as equation (1).

$$B(T)=\left(1-\sum_{i=1}^{n}\frac{C_i}{P_i}-\frac{\gamma}{TTRT}\right)\cdot bT \qquad (1)$$

where γ is token rotation time of TTRT and b is network bandwidth. If we define e_i as the number of frames which have met any error during transmission (The average of e_i is $\rho \cdot C_i$, where ρ is network error rate), R_i, the minimum amount of extra bandwidth for S_i's error control, can be calculated as shown in equation (2).

$$R_i = B(P_i)-\sum_{k=1}^{n} C_k \cdot e_i \qquad (2)$$

As a result, for S_i's period, packet error can be recovered when inequality (3) holds.

$$e_i \cdot C_i \le R_i \qquad (3)$$

V. SIMULATION RESULTS

We have measured the performance of the proposed scheme via simulation using SMPL[13]. In the experiment, we decide TTRT and H_i according to Malcomn's work and normalized proportional scheme, respectively[2]. Namely,

$$TTRT = \frac{P_{min}}{\left\lceil \dfrac{-3+\sqrt{9+8P_{min}/\gamma}}{2} \right\rceil} \qquad (4)$$

$$H_i = \frac{C_i \cdot P_i}{U}(TTRT - \gamma) \tag{5}$$

where $U = \sum \frac{C_i}{P_i}$ and γ is the protocol overhead including token rotation time. The selection is for simplicity, as we are mainly concerned on the performance of the proposed error control function. As natural, it can be applied to other allocation schemes.

This section first exhibits success ratio according to medium error rate. In the experiment, we say that a frame is successfully transmitted when the frame error is masked out by the successful transmission of the next frame in the subsequent period as well as, not to mention, when the frame is sent correctly at the first trial. For the experiment, we have generated 40 stream sets which have arbitrary numbers of streams individually with their utilizations ranging from 70% to 79%. This experiment measures the success ratio for the medium error rate from 10^{-5} to 10^{-10}. As expected, the message transmission supported by the proposed error control scheme outperforms that without error control scheme for all error rates as shown in the Figure 6. The graph shows that the proposed scheme enhances success ratio up to as much as 20% at maximum when error rate is 10^{-6}. The resolution of the EIT system depends on various variables, such as the conductivity contrast and its distribution, injected current patterns, and the errors in voltage measurements. This calls for a verification test to assess the appropriateness of the present EIT technique. A series of simulation has been carried out in this regard.

In addition, Figure 7 show the success ratio according to the utilization of message stream set. As Eq.(1) indicates, the more error can be coped with, the smaller the utilization is. As we have generated stream set with equal numbers of stream irrespective of the number of streams, the average message size increases according to the growth of utilization. Hence, in the case of no control, the success ratio slightly decreases according to the increase of utilization. As contrary, the proposed error control scheme is more affected by the utilization, as natural.

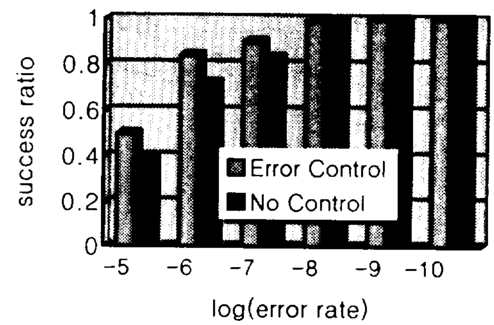


Fig. 6. Success ratio vs. error rate

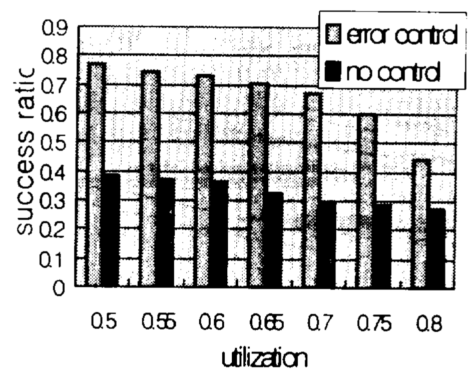


Fig. 7. Success ratio vs. utilization

VI. CONCLUDING REMARKS

In this paper, we have proposed and analyzed an error control scheme for a real-time application which requires forward error masking functionality. The proposed scheme duplicately transmits the frame which has met network error in the previous period, aiming at minimizing the probability of successive frame errors across multiple period, rather than retransmits the frame itself for fast reaction at the computing nodes. We are currently developing an analytical model of the proposed error control scheme as well as measuring the performance via simulation including such parameters as utilization and so on. Furthermore, when the multiple frame errors occur over several connections, the collection node should determine the control priority for the frames. The priority should reflect the value of frames in the individual transmissions.

ACKNOWLEDGEMENTS

This study was sponsored by Nuclear Academic Research Program by Ministry of Science & Technology (MOST).

REFERENCES

1) K. Arvind, Krithi Ramamritham and John A. Stankovic. 1991. A local area network architecture for communication in distributed real-time systems. *J. Real-Time Systems*, Vol. 3, pp.115-147.
2) N. Malcolm and W. Zhao. 1995. Hard real-time communication in multiple-access networks,'" *J. Real-Time Systems*, pp. 37-77.
3) B. Chen, G. Agrawal and W. Zhao. 1992. Optimal synchronous capacity allocation for hard real-time communications with the timed token protocol. *Proc. Real-Time Systems Symposium*, pp.198-207.
4) M. Schwartz. 1987. *Telecommunication Networks: Protocols, Modeling and Analysis*. Addison Wesley Publishing Company.
5) S. Mirchandani and R. Khanna. 1993. *FDDI Technology and Applications*, John Wiley and Sons, Inc.
6) K. Cho, S. Kim. 1999. Adaptive mesh grouping in electrical impedance tomography for bubble visualization. *KSME Int. J.*, Vol. 13, pp. 504-515, 1999.
7) H. Kopetz, A. Damm, J. Reisinger, W. Schwabl. 1988. The real-time operating systems of MARS. *Proc. Operating Systems Review*, pp.141-157.
8) H. Garcia-Molina, B. Gao, D. Barbara. 1991. Aggressive transmissions over redundant paths. *Proc. Distributed Computing Systems*, pp.198-207, 1991.
9) H. Bengtsson, et al., 1999. Protocol for Wireless Real-Time Systems. Euromicro conf. on RTS '99.
10) J. Lee, C. Kim. 1999. Design of an error control scheme for hard real-time communication on FDDI networks. *IEEE TENCON*, pp.828-831.
11) A. N. Natravali, W. D. Roome, K. Sabnani. 1990. Design and implementation of a high-speed transport protocol. *IEEE Trans. Communication*, Vol. 38, No. 11, pp.2010-2024.
12) J. Lee. 1996. An error control scheme for Ethernet-based real-time communication. *Proc. Real-Time Computing Systems and Applications*, pp.214-219.
13) M. H. MacDougall. 1987. *Simulating Computer Systems: Techniques and Tools*. MIT Press.