

석사학위논문

FDDI 기반 실시간 통신에서의
오류제어 기법



115-150
제주대학교 중앙도서관
JEJU NATIONAL UNIVERSITY LIBRARY

제주대학교 대학원
전산통계학과

진 용 문

2000년 12월

FDDI 기반 실시간 통신에서의 오류제어 기법

지도교수 이 정 훈

진 용 문

이 논문을 이학 석사학위 논문으로 제출함.



진용문의 이학 석사학위 논문을 인준함.

심사위원장 _____ (인)

위 원 _____ (인)

위 원 _____ (인)

제주대학교 대학원

2000년 12월

An error control scheme
on FDDI-based real-time communication

Yong-Moon Jin

(Supervised by professor Jung-Hoon Lee)



A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF SCIENCE.

DEPARTMENT OF COMPUTER SCIENCE AND STATISTIC
GRADUATE SCHOOL
CHEJU NATIONAL UNIVERSITY

DECEMBER 2000

목 차

List of Figures	i
Abstract	ii
I. 서 론	1
II. 관련연구	6
1. 오류제어 기법	6
III. FDDI에서의 실시간 통신	9
1. 네트워크 모델 및 메시지 모델	9
1) 네트워크 모델	9
2) 메시지 모델	10
2. 프로토콜 인자	11
3. 대역폭 할당 기법	13
IV. FDDI 상에서의 오류제어 기법	16
1. 오류의 탐지 및 보고	16
2. 오류제어 기법	19
3. 재전송 기법	20

V. 성능평가 ----- 23
1. 기본 설정 ----- 23

VI. 결 론 ----- 27

VII. 참고문헌 ----- 29

초 록 ----- 30



List of Figures

Figure 1. Duplicated transmission of packet in MARS -----	6
Figure 2. Network topology -----	9
Figure 3. Frame formats -----	17
Figure 4. Detection and reporting of frame error -----	19
Figure 5. State diagram of the sender/receiver node -----	20
Figure 6.  log(BER) vs. deadline meet ratio -----	24
Figure 7. Use rate vs. deadline meet ratio -----	25
Figure 8. Asynchronous traffic load vs. deadline meet ratio -----	26

Abstract

This thesis proposes and analyzes an error control scheme for FDDI-based hard real-time communications. Proposed scheme fulfills error control functionality not for all of errors as in the traditional schemes but those that can be recovered within their deadlines according to best-effort strategy. The scheme detects frame errors mainly by the frame error check field dedicated in the FDDI frame. Error report message, including the list of erroneously received frames, is sent by a receiver on respective correct intervals determined by counting the number of received tokens. As our scheme retransmits the frames via overallocated bandwidth in the error control procedure, it does not interfere other normal message transmissions and is able to enhance deadline meet ratio by decreasing message loss. In order to assess the performance of the proposed scheme, we measured the deadline meet ratio according to medium error rate, utilization of stream set, and asynchronous traffic load via simulation using SMPL. The main results are as follows: First, for the given stream set whose utilization is 70 %, the proposed scheme enhances the deadline meet ratio of message up to 29% at medium error rate 10^{-6} . Second, as our scheme uses overallocated bandwidth, the better performance is expected as the utilization decreases. Third, the deadline meet ratio is stable for the varying asynchronous traffic load. As a result, the proposed scheme is able to overcome the transmission errors for the real-time communication based on those protocols which send messages according to the round-robin fashion.

I. 서 론

경성 실시간 통신(hard real-time communication)의 목적은 네트워크의 오류가 발생하지 않는 한 메시지들의 종료시한 내 전송을 보장하는 것으로서 이를 위해 경성 실시간 메시지들의 전송은 정확하게 스케줄 되어야 한다(Arvind 등 1991).

한 네트워크가 분산된 여러 노드들에 의해 공유되어 있는 다중 접근(multiple access) 네트워크에서 메시지 전송의 스케줄은 매체 접근 제어(MAC: Medium Access Control) 계층의 기능에 해당되며 이 계층은 노드들의 네트워크에 대한 접근을 중재하고 각 시점에 전송할 메시지를 결정한다(Malcolm 등 1995).

다중 접근 네트워크에서 경성 실시간 통신을 지원하는 기법은 크게 접근 중재(access arbitration) 방식과 전송 제어(transmission control) 방식으로 구분되는데 접근 중재 방식은 노드가 공유된 채널에 언제 접근할지를 결정하는데 중점을 두고 있는 반면 전송 제어 방식은 노드가 자신의 차례에 전송할 수 있는 양을 결정하는데 중점을 두고 있다. 전송 제어 방식은 FDDI(Fiber Distributed Data Interface)나 토큰 링 등과 같이 각 노드들이 라운드 로빈(round-robin) 형태로 전송하는 프로토콜들에 적합하다(Mirchandani 등 1993).

실시간 메시지들은 주기적으로 도착하고 종료시한(deadline) 이내에 전송이 완료되어야 한다는 시간 제약조건을 가지며 종료시한 이후 전송된 메시지는 손실된 것으로 간주된다. 실시간 메시지들은 일반적으로 트래픽 인자가 사전에 알려지기 때문에 이 분야의 대부분의 연구는 보장(guarantee) 정책을 따른다. 목표는 시스템에 포함된 모든 실시간 메시지 종료시한을 만족시키는 것을 보장하는 것이다. 경성 실시간 통신 이외에 연성 실시간 통신(soft real-time communication) 응용에서는 일부 실시간 메시지들이 종료시한을 놓치는 경우도 허용한다. 그러나 이러한 특성을 수용하는 다중 접근 프로토콜의 설계도 다양한 연구 분야에서 연구되어 왔다.

MAC 프로토콜은 접근 중재와 전송 제어 과정으로 구성되며 메시지 종료시한들을 만족시키기 위해서는 이러한 과정, 즉 대역폭 할당이 주의 깊게 선택되어야 한다.

접근 중재 기반 기법은 적절한 접근 중재 절차의 사용에 초점을 두고 있다. 물론 전송 제어 절차는 이 접근법으로 사용되어지지만 실시간 메시지 종료시한을 만족시키기 위해 접근 중재 절차를 사용하는데 중점을 두고 있다. 접근 중재 기반의 접근법의 가장 좋은 예는 우선 순위 기반의 프로토콜을 사용하는 비율 단조(rate monotonic) 스케줄링이다. 우선 순위 기반의 프로토콜에서는 각 메시지에 우선 순위가 할당되고 접근 중재 절차는 높은 우선 순위를 갖는 메시지가 낮은 우선 순위를 갖는 메시지보다 먼저 전송되도록 한다. 비율 단조 알고리즘은 실시간 메시지 종료시한을 만족시키는 방식으로 주기가 짧은 혹은 종료시한이 촉박한 메시지에 우선 순위를 부여하는 방식이다.

전송 제어 기반 기법은 실시간 메시지 종료시한을 만족시키기 위해서 정적인 전송 제어 절차를 사용하는데 중점을 두고 있으며 전형적으로 시간제한 토큰(timed token) 프로토콜을 사용하는데 초점을 둔다. 시간제한 토큰 프로토콜에서는 전송 제어 절차가 각 노드 i 에 실시간 메시지를 전송할 수 있도록 하는 고정된 시간, 즉 토큰을 받았을 때 배타적으로 네트워크에 접근할 수 있는 시간 H_i 를 인자로 갖는데 H_i 는 실시간 대역폭으로 정의된다. 결국 전송 제어 기반 기법에서 실시간 메시지는 실시간 대역폭을 적절하게 설정함으로써 보장되어질 수 있다.

실시간 프로토콜들이 메시지들의 시간 제약조건(time constraint)을 만족시킬 수 있기는 하지만 전송 오류는 불가피하게 발생하며 이는 실시간 메시지의 손실을 초래한다. 전송 오류는 매체 오류율과 외부 자극에 의한 일시적인 교란에 의해 발생하는데 통신 기술의 발달로 인해 매체 오류율이 감소하고 있기는 하지만 완전히 제거되지는 못하고 있다. 또 멀티미디어 응용의 도입은 메시지의 양을 증가시켜 한 비트 오류가 전체 메시지의 오류로 확대되므로 매체 오류율의 영향을 증폭시키고 있다. 더욱이 새롭게 출현하고 있는 무선 통신 환경은 오류의 발생율이 증대되므로 효율적인 오류제어 기능이 필요하다.

프로토콜 내에서 오류제어 기능이 오류를 탐지(detect)하고 보정(correct)하는 기능을

수행하는데 고전적인 오류제어 기법은 go-back-N, 정지 대기(stop-and-wait), 선택적 재전송(selective repeat) 등이지만 이들은 수신 확인(acknowledgement)과 재전송 요구(retransmission request) 및 오류 발생 메시지의 재전송 등 오류 제어를 위한 일련의 과정들이 메시지 전송 시간을 연장하기 때문에 시간 제약조건을 갖는 실시간 메시지의 전송에는 적합하지 못하다. 더욱이 재전송 되는 메시지들이 일반적인 실시간 메시지의 전송에 간섭을 일으켜 이들 메시지의 지연을 초래할 가능성도 포함한다.

전송 제어 방식에 기반한 실시간 통신에서는 대역폭 할당 기법이 각 메시지 스트림에 대하여 최소의 전송시간을 갖는 주기를 기반으로 용량 벡터(capacity vector)를 결정하기 때문에 각 스트림은 여분의 대역폭을 갖게 된다. 또 FDDI 프로토콜에서는 실시간 메시지들이 사용하지 않는 대역폭을 통해 실시간 메시지들이 전송되는데 이와 같이 과할당된 대역폭과 비실시간 메시지들을 위한 대역폭을 오류제어 기능을 위해 사용하면 메시지의 전송에 영향을 주지 않게 되며 전송 오류에 의한 메시지의 손실을 감소시켜 메시지들의 종료시한 만족도를 개선할 수 있다. 시스템에 실시간 스트림만 존재하는 경우에는 과할당된 대역폭을 오류 제어 목적으로 사용할 수 있으며 이 경우에는 실시간 메시지의 사용율이 적을수록 높은 종료시한 만족도를 기대할 수 있으나 네트워크에 실시간 및 비실시간 메시지가 혼재하는 경우에는 적용되기 어렵다(Lee 등 1999). 이와 아울러 비동기 트래픽을 위해 할당된 대역폭을 이용하여 오류 제어를 수행하면 역시 최선 노력(best effort) 방식으로 오류를 복구할 수 있다(정충일 등 1999). 이 때 물론 비동기 트래픽에 간섭을 줄 수 있으나, 즉 비동기 트래픽의 전송시간을 연장할 가능성이 있으나 비동기 트래픽은 시간에 민감하지 않으므로 시스템 전체적인 성능에는 크게 영향을 주지 않는다.

지리적으로 분산된 스테이션(station; 사용자)들이 단일 채널 상에서 메시지들을 전송함으로써 통신하고자 하는 상황을 고려해 보자. 이 채널은 스테이션 사이의 유일한 통신 수단이 되며 채널 상으로 한번에 하나의 메시지만 전송될 수 있다는 특성을 가지고 있다. 두 개 이상의 메시지들이 동시에 채널에 전송된다면 이러한 메시지들은 서로 방

해하게 되어 어떤 것도 원하는 목적지 스테이션에 올바르게 수신되어질 수 없다. 그러한 환경을 다중 접근 환경(multiple-access environment)이라고 불리어진다. 모든 스테이션들이 단일 통신 채널을 감시할 수 있기 때문에 한 스테이션에 의해 전송되어지는 메시지는 모든 다른 스테이션에 의해 발견되거나 감지된다. 이러한 이유로 다중 접근 환경은 방송 환경(broadcast environment)의 하나의 타입으로 알려져 있다.

다중 접근 환경의 수많은 예들이 있다. 일상 생활의 예로는 하나의 토의 집단과 그들 사이의 공기를 들 수 있다. 공기는 사람들이 의사 소통을 하기 위해 사용되어지는 유일한 물리적 매체이다. 모든 사람이 알다시피 두 사람 이상이 동시에 말하면 누구도 그 사람들이 이야기한 것을 알아들을 수 없는 상황이 벌어진다. 위성 채널과 지리적으로 분산된 지상의 스테이션들(ALOHA 시스템)도 다중 접근 환경을 구성한다. 위성 네트워크에서는 지상의 스테이션들이 위성을 향하여 메시지들을 전송하고 그 메시지들은 다시 지상의 스테이션 쪽으로 전송된다. 지상의 스테이션 또는 위성에 의한 동시 전송은 메시지들이 서로 방해받음 받고 지상 스테이션에서 알기 힘든 메시지들을 수신하는 결과를 초래하게 된다. 다중 접근 환경의 다른 타입은 분산된 스테이션들이 단일 라디오 채널을 통해 통신하는 지상 패킷 라디오 네트워크이다. 라디오 전파는 스테이션 사이의 매체를 통하여 전파되고 방해받는 라디오 전파들은 목적지 스테이션에서 불안정한 메시지를 수신하게 하는 결과를 초래한다. 다중 접근 환경에서 가장 자주 인용되는 예는 이더넷(Ethernet)과 같은 지역 네트워크이다. 이더넷은 분산된 스테이션이 유일한 통신 매개체로서 단일 동축케이블이나 광섬유를 공유한다.

하나의 스테이션 만이 주어진 시간에 메시지를 성공적으로 전송할 수 있기 때문에 스테이션은 서로 채널을 공유하기 위해서 채널에 대한 접근을 어떻게든 조정해야 한다. 스테이션들이 채널을 공유하는 분산 알고리즘은 채널 접근 프로토콜(channel-access protocol)로 알려져 있다. 채널 공유의 문제를 복잡하게 하는 몇 가지 논점들이 있다. 첫째, 스테이션들이 분산되어 있기 때문에 채널 공유를 조정하고자 하면 명시적이든 암시적이든 서로 정보를 교환하여야 한다. 그러나 단일 통신 채널이 유일함으로 채널 공유에 관한 사용자들의 조정은 반드시 이루어져야 한다. 둘째, 스테이션들이 분산되어 있기 때문에 다른 스테이션들의 현재 상황을 즉각적으로 알 수가 없다. 다른 스테이션

에 관한 정보는 항상 스테이션들 사이에 메시지들이 전송 시간만큼은 오래 되어야 한다. 이것은 우리가 바라보는 별빛이 실제로는 수백만 년 전의 별에서 발생한 것이라는 상황과 유사하다.

지난 10여 년 동안 다중 접근 문제를 해결하기 위해 수많은 접근법들이 제안되어 왔다.

본 논문은 FDDI에 기반한 경성 실시간 통신을 위한 오류제어 기법을 제안하고 평가함을 목적으로 하며 다음과 같이 구성된다. IV절에서는 기존의 FDDI에서의 실시간 통신과 오류제어 기법을 제안한다. 이에는 오류 제어에 과할당된 대역폭을 이용하는 경우와 비동기 트래픽을 이용하는 경우를 모두 포함한다. V절에서 실험 결과를 보인 후 마지막으로 VI절에서는 논문을 요약하고 추후의 연구 과제를 도출한다.



II. 관련 연구

1. 오류제어 기법

실시간 데이터 링크 계층에서는 두 종류의 오류제어 기법이 널리 사용되는데 이는 시간적 여분(temporal redundancy)과 공간적 여분(spatial redundancy)을 이용하는 방식이다(Kurose, 1994).

시간적 여분을 이용하는 방식은 메시지의 종료시한이 비교적 긴 분산 응용에 적합한데 이 방식은 물리적인 네트워크의 중복없이 동일한 네트워크 상에서 메시지를 중복 전송하여 오류율을 줄이거나 재전송 요청 혹은 타임아웃을 이용한 방식은 재전송에 관련된 메시지들이 하부 네트워크 프로토콜에 맞추어 스케줄되어야 한다는 문제점을 갖고 있으며 중복 전송에 의한 방식은 대역폭의 낭비를 크게 하는 문제점을 갖고 있다. 시간적 여분의 방식의 예로서 MARS(MAintainable Real-time System) 시스템을 들 수 있는데, 이 시스템에서는 <Figure 1>에서 보는 바와 같이 오류 발생 여부에 관계없이 모든 메시지들이 이중으로 중복되어 전송된다(Kopetz 등 1988).

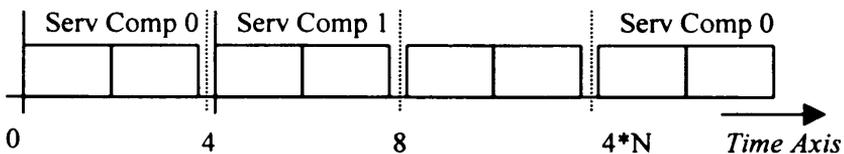


Figure 1. Duplicated transmission of packet in MARS

반면 공간적 여분을 사용한 방식에서는 중복된 메시지들이 서로 다른 채널이나 네트

워크를 통해 전송되며, 메시지의 종료시한이 촉박하여 재전송이나 중복전송을 허용할 수 없을 경우에 적합하다(Garcia 등 1991). 이 방식은 항상 메시지들이 다중으로 중복되어 전송되므로 메시지들에 의해 요구된 대역폭의 중복 배수만큼의 대역폭을 필요로 하게 되어 상당한 대역폭의 낭비를 초래할 뿐 아니라 여러 경로로 다른 시간대에 수신되는 메시지들을 처리하여야 하므로 수신자의 기능이 복잡해진다.

이상의 기법들 이외에 오류 보정 코드에 의한 전향 오류 제어 방식이 효율적으로 보이지만 정보의 낭비가 크게 되며 처리시간이 길어지게 된다. 그 예로 Hamming 코드 방식에서는 메시지의 크기를 m 비트라 할 때 한 비트의 오류를 보정하기 위한 오버헤드 k 는 $2^k \geq m+k+1$ 의 관계식을 만족시켜야 하는데 메시지의 크기가 커질수록 대역폭 낭비 요소의 k 의 길이가 증가한다(Kohavi, 1978). 이 경우에도 2 비트의 동시 오류는 탐지만 가능하며 보정은 할 수 없다. 오류 보정 코드는 오류가 발생하지 않는 경우에도 고정적으로 부가적인 정보가 추가되므로 대역폭의 낭비를 초래하며 송신자와 수신자 측에서 오류를 보정하기 위해 많은 데이터 연산을 수행하여야 한다. 이와 아울러 오류 보정 코드는 메시지의 오류율을 감소시킬 수 있으며 오류제어 기능과 결합한다면 오류 처리율을 개선할 수 있게 되고 무선 통신에서 많이 적용되고 있다.

또 임의의 하부 네트워크를 기반으로 트랜스포트 계층에서 오류를 제어하는 방식의 예로서 사전 재전송 방식은 오류의 발생 여부를 최종적으로 확인하기 위해 기다리지 않고 오류의 가능성이 있다고 판단될 때 미리 재전송을 수행하도록 한다(정충일 등 1999). 이는 오류 복구가 종료시한 이전에 이루어질 수 있는 시간적 여유를 갖도록 함을 목적으로 하고 있다.

그러나 이 방식은 오류의 가능성을 판단하기 위해 사전 재전송 시점을 결정하여야 한다는 추가적인 요구 사항을 갖고 있으며 재전송이 다른 실시간 스트림의 전송을 지연할 수 있다는 문제점을 갖고 있다. 또 MAC 계층보다는 트랜스포트 계층에서의 오류 제어를 대상으로 하고 있기 때문에 다양한 하부 MAC 구조에 적용될 수는 있으나 실시간 MAC의 기능을 이용하지 못한다.

기타 트랜스포트 계층의 오류제어 기법들이 실시간 통신에 일부 적용이 가능하지만 시간제약 조건을 직접적으로 고려하고 있지는 않다(Gong 등 1996; Natravali 등 1990).



III. FDDI에서의 실시간 통신

1. 네트워크 모델 및 메시지 모델

1) 네트워크 모델

네트워크 토폴로지는 점간 연결 방식으로 m 개의 노드들이 원모양, 즉 토큰링으로 이루어져 있다. 토큰이라고 불리는 특정한 비트 패턴은 노드의 수가 m 개일 때 노드 i 부터 노드 $i+1$, 노드 $i+2$, ..., 노드 m 까지 링 토폴로지를 따라 순환하면서 서로 메시지를 전송하려는 노드들 사이의 순서를 결정하는 역할을 한다.

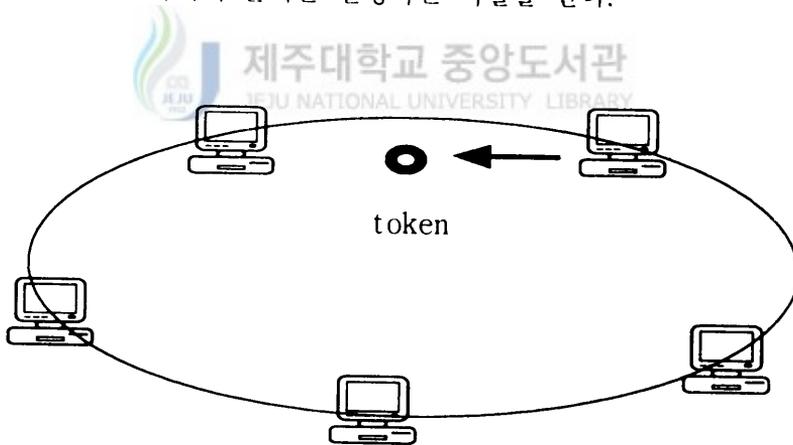


Figure 2. Network topology

θ_i 를 노드 i 와 상향(upstream) 인접노드($i+1$) 사이에 토큰 전송 지연 시간이라 하며 이 오버헤드에는 노드 비트 지연, 노드 대기 버퍼 지연, 미디어 전파 지연 등을 포함한

다. 아무 노드도 전송하지 않을 때 링 상에서의 총오버헤드 Θ 는 토큰이 회전하는데 걸리는 시간으로서 $\Theta = \sum_{i=1}^n \theta_i$ 으로 계산된다. 결국 링 지연 시간(latency) Θ 는 네트워크 상의 노드들이 토큰을 방해하지 않을 때 토큰이 링을 회전하는 시간을 의미한다.

2) 메시지 모델

실행 시간에 시스템 상에 발생하는 메시지들은 실시간 메시지들 또는 비동기 메시지들이다.

n 개의 실시간 메시지 스트림이 있다고 가정하면 실시간 메시지 스트림의 집합, M 은 S_1, S_2, \dots, S_n 으로 설정된다. 즉, $M = \{S_1, S_2, \dots, S_n\}$ 이다. 반면 비동기 메시지들은 동적으로 임의의 노드에서 발생한다.

이러한 메시지들의 특징은 다음과 같다.

1. 실시간 메시지들은 주기적이다. 즉, 실시간 메시지 스트림 상의 메시지에는 일정한 간격의 도착 시간이 있다. 스트림 $S_i (i=1, 2, \dots, n)$ 의 주기는 P_i 이다.
2. 각 실시간 메시지의 종료시한은 실시간 메시지가 도착하는 주기로 설정된다. 만일 스트림 S_i 의 메시지가 시간 t 에 도착한다면 종료시한은 시간 $t+P_i$ 이다.
3. 메시지들의 도착들이 다른 메시지들의 전송 요구들의 개시 또는 완료에 의존하지 않는다는 점에서 메시지들은 서로 독립적이다.
4. 스트림 S_i 의 각 메시지의 길이는 이 메시지를 보내기 위해 필요로 하게 되는 시간의 최대값 C_i 이다.

5. 비동기 메시지들은 비주기적이고 경성 실시간 종료시한이 요구되지 않는다.

실시간 메시지 집합의 사용 인자 $U(M)$ 은 실시간 메시지들의 전송에서 네트워크에 의해 쓰여지는 시간의 일부분으로서 정의된다. 즉, n 을 실시간 메시지 스트림의 수라 할 때

$$U(M) = \sum_{i=1}^n \frac{C_i}{P_i} \quad (1)$$

와 같이 계산된다.

각 노드마다 오직 하나의 스트림만이 존재한다고 단순화할 수 있는데 이러한 가정은 가상 노드(virtual node) 개념에 의해 한 노드에 다수의 스트림이 포함된 경우로 확장될 수 있다. 즉 가상의 노드에 각각 속한 스트림에 대해 용량 벡터를 할당하고 물리적으로 같은 노드에 속한 스트림들의 용량 벡터 값을 더하면 실제 용량 벡터가 산출된다 (Agrawal 등 1992).

2. 프로토콜 인자

대역폭 할당에 관련된 시간제한 토큰 프로토콜의 2개의 네트워크 인자들은 다음과 같다.

- Target Token Rotation Time(TTRT)

네트워크가 초기화될 때 TTRT의 값이 결정된다. TTRT는 토큰 회전 시간을 결정 짓는 요소로서 주어진 TTRT에 대해 한 노드에서 토큰을 맞이하는 간격의 최대값은 $2 \times TTRT$ 이다. 따라서 만약 주기가 $2 \times TTRT$ 보다 작다면 노드는 특정 주기 중에 토큰을 전혀 맞이하지 못할 수도 있다(Sevcik 등 1987).

실시간 메시지들이 그들의 주기의 끝으로서 그들의 종료시한들을 갖기 때문에 메시지 종료시한을 만족시키기 위해서는 TTRT 값은 다음과 같은 필요조건을 갖는다. $1 \leq i \leq n$ 에서

$$TTRT \leq \frac{P_i}{2} \quad (2)$$

이고 여기서 P_i 는 실시간 메시지 스트림 S_i 의 주기이다. 그러므로 어떤 P_i 도 TTRT의 선형 함수로 표현되어 진다. 즉,

$$P_i = q_i \cdot TTRT + \gamma_i \quad (3)$$

이고 여기서 q_i 는 스트림 S_i 의 한 주기 내에서 토큰을 맞이하는 횟수로서 $q_i = \left\lfloor \frac{P_i}{TTRT} \right\rfloor$ 와 같이 계산되고 γ_i 는 마지막 토큰을 방출한 후 주기가 종료할 때까지의 시간으로서 다음과 같이 계산된다.

$$\gamma_i = P_i - q_i \cdot TTRT = P_i - \left\lfloor \frac{P_i}{TTRT} \right\rfloor \cdot TTRT \quad (4)$$

P_i 에 대한 수식은 뒤에 나오게 되는 여러 가지 이론에 사용되는 것에 따라 나타내었다. 수식 (3)은 이 논문 전체에서 유지된다고 가정한다.

- 노드 i 의 실시간 대역폭(H_i)

이 인자는 스테이션이 토큰을 받을 때마다 실시간 메시지를 전송하도록 허용하는 최대 시간을 나타낸다. 각 스테이션은 갖기 다른 H_i 값을 할당받을 수 있으며 실시간 대역폭 할당을 지원하는 H_i 값을 할당한다. 용량벡터 \vec{H} 는 다음과 같이 표현된다.

$$\vec{H} = (H_1, H_2, \dots, H_n) \quad (5)$$

앞에서 언급한 바와 같이 실시간 대역폭 할당은 실시간 메시지 종료시한들을 보장하는 과정에 있어서 중요한 역할을 한다.

3. 대역폭 할당 기법



기존에 FDDI 네트워크에서 실시간 스트림들의 시간 제약조건을 만족시키기 위해 연구된 대역폭 할당 기법들은 다음과 같다(Agrawal 등 1992).

- 총 대역폭 할당 기법(Full length scheme)

이 기법에서는 노드에 할당된 실시간 대역폭이 실시간 메시지들을 전송하는데 필요한 전체 시간과 동일하다. 즉,

$$H_i = C_i \quad (6)$$

이다. 이 기법은 노드에 도착하는 실시간 메시지를 분리해서 각 주기 P_i 마다 전송하는 것이 아니라 한 번에 전송하려는 시도이다.

- 비례 할당 기법(Proportional scheme)

이 기법에서는 노드 i 에 할당된 실시간 대역폭이 C_i 와 P_i 의 비율에 비례한다. 즉,

$$H_i = \frac{C_i}{P_i} (TTRT - \gamma) \quad (7)$$

이다.

- 균등 분할 기법(Equal partition scheme)

이 기법에서는 TTRT의 사용 가능한 시간이 n 개의 노드에 균등하게 분배된다. 즉,

$$H_i = \frac{TTRT - \gamma}{n} \quad (8)$$

이고 여기서 n 은 시스템 노드들의 수이다.

- 정규화한 비례 기법(Normalized proportional scheme)

이 기법에서는 실시간 대역폭이 노드에서 실시간 메시지의 정규화된 부하에 따라 할당된다. 즉,

$$H_i = \frac{C_i/P_i}{U} (TTRT - \gamma) \quad (9)$$

이고 여기서 U 는 사용율로서 (1)에서 보인 바와 같다. 이외에 Zhao 등은 이상의 제약 조건들을 만족시키는 값을 찾는 과정에서 복잡한 계산 과정을 필요로 하기는 하지만 최적의 대역폭 할당 기법을 제시하였다.

이상의 대역폭 할당 기법들은 조정자(coordinator) 노드에서 수행되어 관리 메시지를

통해 각 노드로 할당된 대역폭의 정보가 전달되는데 반해 지역 할당 기법은 각 분산된 노드들이 스스로 대역폭을 결정하고 트래픽 특성이 변화한 경우에도 동적으로 H_i 값을 수정할 수 있도록 한다(Chen 등 1992).



IV. FDDI 상에서의 오류제어 기법

실시간 메시지들은 주기적으로 발생되며 이들은 주기 내에 전송이 완료되어야 하며 FDDI 네트워크에서 메시지들의 시간 제약조건을 만족시키기 위해서는 TTRT나 용량 벡터와 같은 네트워크 인자들이 정확하게 설정되어야 한다. TTRT는 토큰이 회전하는 기대치로서 각 노드들은 자신의 메시지 특성에 따라 TTRT 중 일부를 할당받는데 이 할당받는 양이 용량 벡터이다. 각 노드는 토큰을 받으면 최대 용량 벡터만큼 실시간 메시지를 전송할 수 있으며 TTRT 보다 토큰이 일찍 도착했을 때에만 비실시간 메시지를 전송할 수 있다.

실시간 시스템에 있어서 모든 실시간 메시지 스트림에 대한 정보는 시스템의 운영 이전에 알려지며 메시지 스트림 S_i 는 주기 P_i 와 전송 시간 C_i 로 특성화되므로 통신 스케줄러는 주어진 $\{S_i\}$ 집합에 대해 실시간 제약조건을 만족시키도록 TTRT와 용량 벡터 $\{H_i\}$ 를 결정한다.

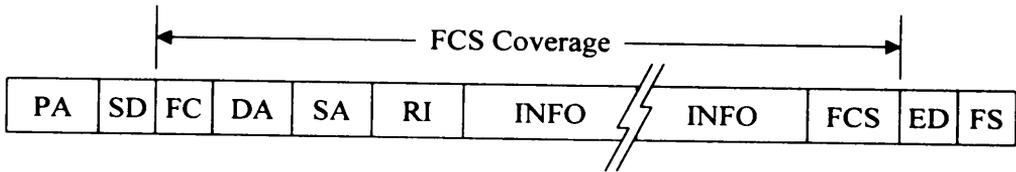
FDDI에서 대역폭을 결정하는 여러 방식들이 제안되고 평가된 바 있으나 이들은 모두 경성 실시간 제약조건을 만족시키기 위하여 최소의 전송 시간을 갖는 주기를 기반으로 용량 벡터를 할당하기 때문에 각 노드는 충분한 대역폭을 할당받게 된다.

1. 오류의 탐지 및 보고

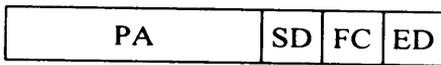
오류 제어를 위해서는 수신자 노드는 수신된 메시지의 오류를 탐지하고 이를 송신자에게 보고하여야 한다. FDDI 네트워크에서 전송 오류의 탐지는 <Figure 3>에서와 같이 프레임에 포함된 FCS(Frame Check Sequence)와 FS(Frame Status) 필드를 이용하는데 FCS는 프레임의 오류를 탐지하기 위한 32비트 CRC(Cycle Redundancy Check)를 포함하고 있으며 FS는 프레임 전송 중에 발생한 오류를 나타내기 위한 ED(Error

Detection) 심볼을 포함하고 있다.

FRAME



TOKEN



PA-Preamble(16 or more symbols)

RI-Routing Field(0 to 30 symbol pairs)

SD-Starting Delimiter(2 symbols)

INFO-Information(0 or more symbol pairs)

FC-Frame Control(2 symbols)

FCS-Frame Check Sequence(8 symbols)

DA-Destination Address(4 or 12 symbols)

ED-Ending Delimiter(1 or 2 symbols)

SA-Source Address(4 or 12 symbols)

FS-Frame Status(3 or more symbols)

Figure 3. Frame formats

FDDI 프레임이 링 토폴로지를 따라 전송될 때 각 노드는 수신된 프레임을 검사하고 오류가 탐지되면 ED 필드를 세팅하는데 모든 프레임은 자신을 전송한 노드로 환원되어 삭제되기 때문에 전송 노드는 프레임에 오류가 발생했는지 판단할 수 있다. FCS를 이용한 오류의 탐지와 보고는 메시지의 손상에 기인한 오류를 탐지하고 보고할 수 있는 하지만 수신 버퍼 초과에 기인한 기각과 같이 수신자 내부 원인에 기인한 오류는 보고할 수 없으며 링 구조상 수신 노드 이후에서 발생한 오류에 대해 송신자와 수신자

사이의 불일치가 발생할 가능성도 내포하고 있다.

이를 극복하기 위해서는 추가적인 재전송 요청 메시지를 통해 송신자에게 오류 발생을 보고하여야 하며 이 재전송 요청 메시지는 실시간 메시지의 전송에 지연을 주지 않도록 비동기 대역폭을 이용하여 전송하여야 하는데 재전송 요청은 SNR(Signal to Noise Ratio) 방식과 같이 재전송 리스트를 주기적으로 전송하는 방안이 바람직하다(Natravali 등 1990). 즉, 수신자는 각 주기마다 재전송 리스트를 초기화하고 프레임 수신시에 오류가 발생하면 오류 프레임을 리스트에 추가하여 재전송 리스트를 주기적으로 송신자에게 보고한다. 재전송 시점의 결정은 FDDI 네트워크에서 효율적으로 계산될 수 있으므로 수신자는 메시지의 송신 시작 시점부터 자신이 수신하는 토큰의 개수를 세고 있으면 한 메시지에 속한 프레임의 전송이 모두 끝났는지 알 수 있게 된다.

<Figure 4>에서는 그 예를 보이고 있는데 한 메시지가 u 개의 프레임으로 나뉘어 전송된다면 수신자는 자신이 수신한 토큰의 수에 의해 송신자의 전송이 완료되었는지 판단할 수 있다.

처음으로 수신된 프레임이 2라고 한다면 카운터를 2로 설정하고(1번 프레임은 재전송 리스트에 추가) 이후부터 토큰의 수신 회수를 측정하여 최종적으로 u 가 된다면 재전송 리스트를 전송한다. 이 방식에 의해 임의의 프레임에 손상이 발생하더라도 토큰만 정확하게 수신이 된다면 수신자는 쉽게 재전송 시점을 결정할 수 있다. 물론 한 메시지에 속한 모든 프레임이 손상되면 재전송 요구를 할 수 없는데 이 경우는 어차피 종료 시한 이내에 오류 복구를 할 가능성이 거의 없다. 재전송되는 프레임의 경우 임의의 순서번호를 갖기 때문에 최초 수신 프레임 번호 인식에 오류를 발생시킬 수 있으므로 일반적인 전송과 재전송되는 프레임을 구분하기 위해 MAC 프레임에 포함된 예약 비트를 사용한다.

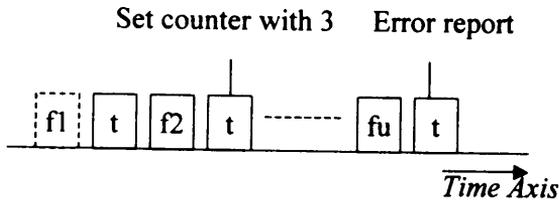


Figure 4. Detection and reporting of frame error

2. 오류제어 기법

송신자 노드는 주기적으로 메시지가 도착하면 이를 다수의 FDDI 프레임으로 분할하여 목적지로 전송하는데 프레임이 돌아오면 오류의 발생 여부를 판단할 수 있으며 송신자는 이러한 정보를 기록한 후 오류 발생 여부와 관계없이 나머지 프레임을 모두 전송한다. 메시지에 속한 모든 프레임의 전송이 완료된 후 주기 내에서 또다른 토큰을 맞이하거나 TTRT 보다 일찍 토큰이 도착하는 경우 비동기 대역폭을 이용해 오류가 발생한 프레임을 재전송한다.

수신자 노드는 오류 때문에 순서에 어긋나게 수신된 노드를 재조립 할 수 있어야 하며 이를 위해 C_i 크기의 버퍼가 필요하다. 만약 오류가 발생하였으나 여분의 대역폭이 없는 경우에는 재전송을 할 수 없게 되며 이때 메시지는 불가피하게 손실로 처리된다.

또 비동기 트래픽을 오류 제어 목적으로 사용하는 경우에는 비실시간 메시지의 전송에 영향을 줄 수 있으나 실시간 메시지의 우선 순위가 높으므로 시스템의 실시간 성능은 향상된다.

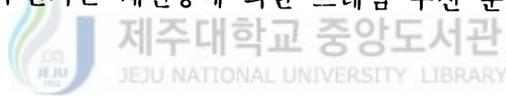
결국 제안된 오류제어 기법은 메시지 전송 오류가 발생한 경우에도 메시지 손실의 가능성을 감소시켜 메시지의 종료시한 만족도를 개선할 수 있는데 여분의 대역폭을 사용하기 때문에 재전송되는 메시지들이 다른 실시간 메시지의 전송에 간섭을 일으키지

않을 뿐 아니라 실시간 통신 시스템의 스케줄 가능성(schedulability)에도 영향을 주지 않는다.

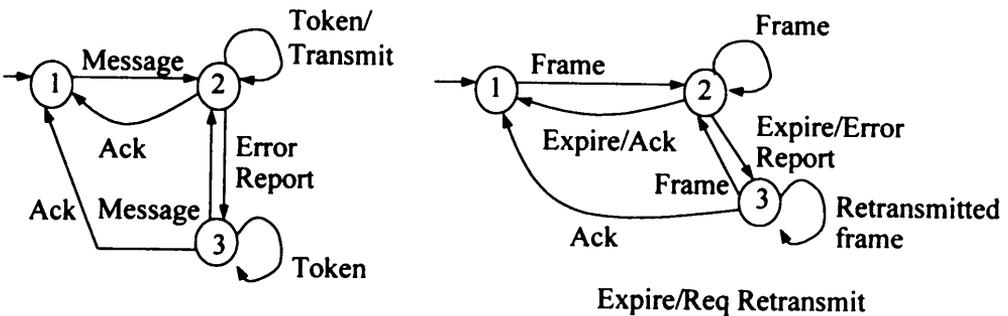
3. 재전송 기법

수신자 노드는 오류 때문에 순서에 어긋나게 수신된 노드를 재조립할 수 있어야 하며 이를 위해 C_i 크기의 버퍼가 필요하다.

송신자 역시 추후의 재전송을 위해 C_i 크기의 버퍼를 할당받는데 재전송 가능성은 주기 이내에서만 발생 가능하며 일반적으로 송신 버퍼는 새로운 메시지가 발생할 때마다 자신의 버퍼에 덮어쓰게 된다. 다른 오류 제어 기법과 마찬가지로 각 프레임마다 순서번호를 추가하며 수신자는 재전송에 의한 프레임 수신 순서의 전복을 복구할 수 있다고 가정한다.



<Figure 5>는 송신자와 수신자의 오류 제어를 위한 상태도를 보여주고 있다.



(a) Sender node

(b) Receiver node

Figure 5. State diagram of the sender/receiver node

<Figure 5(a)>는 송신자 측의 상태를 보여주고 있는데 상태 1은 초기 상태로서 노드 내부에서 전송할 메시지가 생성되면 상태 2로 천이하며 상태 2에서는 토큰을 받을 때 마다 한 프레임씩 전송한다. 상태 2에서 수신자로부터 재전송 요청과 확인을 받을 수 있으며 이들 메시지가 손실된 경우에는 다른 메시지가 도착한다. 확인을 받으면 상태 1로 천이하여 다음 메시지의 도착을 기다리게 되며 재전송 요청을 받으면 상태 2로 천이하여 토큰을 맞이할 때마다 재전송 리스트에 포함된 프레임을 하나씩 전송한다. 상태 2에서 재전송 중 새로운 메시지가 도착하면 재전송 과정을 기각하고 상태 2로 천이하여 일반 프레임을 전송한다.

<Figure 5(b)>는 수신자 측의 상태를 보여주고 있는데 수신자는 상태 1에서 일반 프레임을 처음으로 수신하면 상태 2로 천이하며 상태 2에서 토큰을 맞이할 때 수신 계수기를 최초로 수신된 프레임의 번호로 설정함과 아울러 재전송 리스트를 초기화한다. 상태 2에서는 계속해서 프레임을 수신하면서 토큰을 맞이할 때마다 계수기를 증가시키는데 수신되지 못한 프레임의 번호를 재전송 리스트에 기록한다. 계수기의 값이 목적치가 되면 메시지의 오류 발생 여부에 따라 확인 혹은 재전송 리스트를 송신자에게 보내고 각각 상태 1과 3으로 천이한다. 상태 3은 재전송되는 프레임들을 수신하는 상태로서 이 상태에서 일반 프레임이 수신되면 이전 메시지의 프레임들에 대한 수신을 포기하고 상태 1로 천이한다. 물론 계수기와 재전송 리스트를 수신된 메시지에 따라 초기화한다. 상태 3에서 재전송에 의해 오류가 복구되었을 때는 상태 1로 천이하여 메시지의 수신을 완료한다.

오류의 보고는 수신자가 송신자에게 재전송 리스트를 전송함으로써 수행되는데 만약 수신자가 토큰을 받았을 때 충분한 비동기 대역폭이 없다면 보고를 할 수 없으며 오류 제어가 불가능하다.

송신자는 한 메시지의 전송이 모두 완료된 후 재전송 요청을 받으면 주기 내에서 이후 토큰을 맞이할 때만 재전송을 수행하고 또 다시 새로운 메시지가 도착하면 이전 재

전송 요청을 무시한다.

이 경우 지난 주기에서의 메시지 전송은 실패로 처리된다.



V. 성능평가

1. 기본 설정

본 절에서는 SMPL을 이용한 모의 실험에 의하여 제안된 오류제어 기법의 성능을 평가하였다(MacDougall, 1987). 실험에서 TTRT와 $\{H_i\}$ 는 Malcomn의 연구와 정규화된 비례 기법(normalized proportional scheme)에 따라 식 (10)과 식 (11)과 같이 설정되었으며 다른 대역폭 할당 기법을 적용할 수도 있다(Chen 등 1992).

$$TTRT = \frac{P_{\min}}{\left| \frac{-3 + \sqrt{9 + \frac{8P_{\min}}{\gamma}}}{2} \right|} \quad (10)$$

$$H_i = \frac{C_i/P_i}{U} (TTRT - \gamma) \quad (11)$$

앞에서 언급한 바와 같이 식 (11)에서 U 는 사용율이며 γ 는 토큰 회전에 따르는 낭비시간이다.

이 경우는 과할당된 대역폭을 이용하여 오류 제어를 수행하는 경우로서 주 측정 인자는 매체 오류율에 따른 종료시한 만족도와 메시지 스트림 집합의 사용율에 따른 종료시한 만족도 등이다. 첫 번째 측정을 위해 사용율 60~69%를 갖는 실시간 스트림 집합 20개를 생성하였다. 종료시한 만족도는 종료시한 내 전송이 완료된 실시간 메시지의 개수를 생성된 전체 실시간 메시지의 수로 나누어 계산된다. 각 스트림 집합은 임의 개수의 메시지 스트림을 가지며 각 스트림의 주기와 전송시간도 임의 분포를 따른다.

비실시간 트래픽의 부하를 0.1로 고정시키고 매체 오류율을 10^{-4} 부터 10^{-9} 까지 변화시켜 가며 종료시한 만족도를 측정하였는데 <Figure 6>에서 보는 바와 같이 모든 매체 오류율에서 제안된 방식이 높은 종료시한 만족도를 보이고 있다. 여기서 매체 오류율은 매체의 비트 오류율(BER: Bit Error Rate)을 의미한다.

실험에서는 C_i 의 크기를 평균 20,000 비트로 설정하였는데 매체 오류율이 무척 높으면(10^{-4} 이상) 어느 방식도 오류를 완벽하게 복구하지 못하며 매체 오류율이 무척 낮으면(10^{-9} 이하) 오류 제어를 하지 않아도 충분한 종료시한 만족도를 기대할 수 있다. 그러나 메시지의 크기가 커지면 한 비트 오류의 영향이 전체 프레임에 파급되므로 오류의 영향이 커지게 된다. 따라서 <Figure 6>의 그래프는 매체 오류율에 대해 우측으로 이동한 형태로 표현될 것이다.

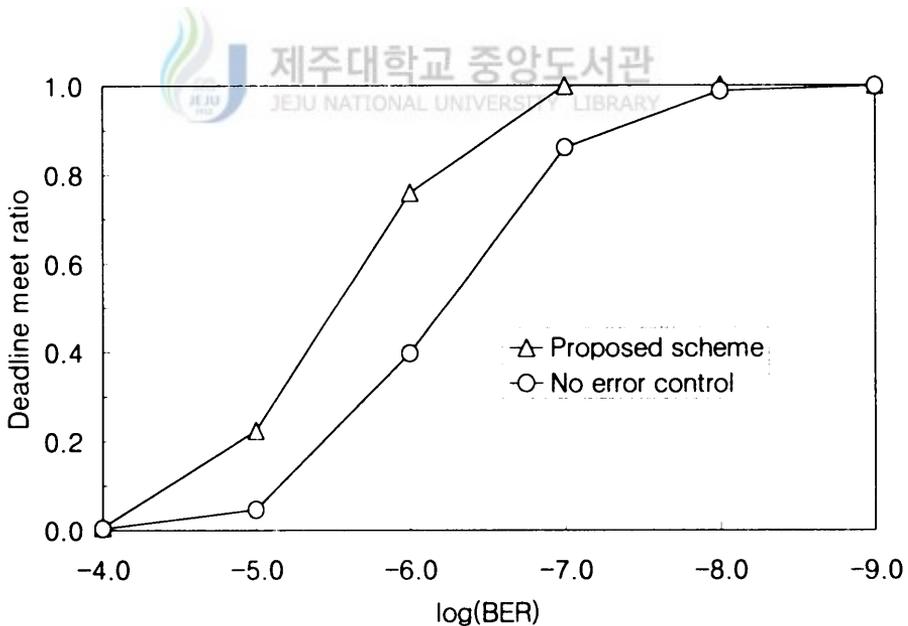


Figure 6. log(BER) vs. deadline meet ratio

<Figure 7>은 매체 오류율을 10^{-6} 으로 고정시킨 채 스트림 집합의 사용율을 0.5에서 0.8로 변화시켜 가며 종료시한 만족도를 측정한 결과이다. 오류 처리를 하지 않는 경우에는 종료시한 만족도가 큰 영향을 받지 않고 다만 집합 내에서 각 스트림의 부하가 균등하게 분리되어 있는지에 조금씩 영향을 받는다. 또 사용율이 높아지면 집합 전체적으로 메시지의 양이 많아지므로 이에 따라 오류 가능성이 증가한다. 반면 제안된 방식은 사용율의 증가에 따라 오류 처리를 위한 대역폭이 줄게 된다. 따라서 종료시한 만족도가 감소한다.

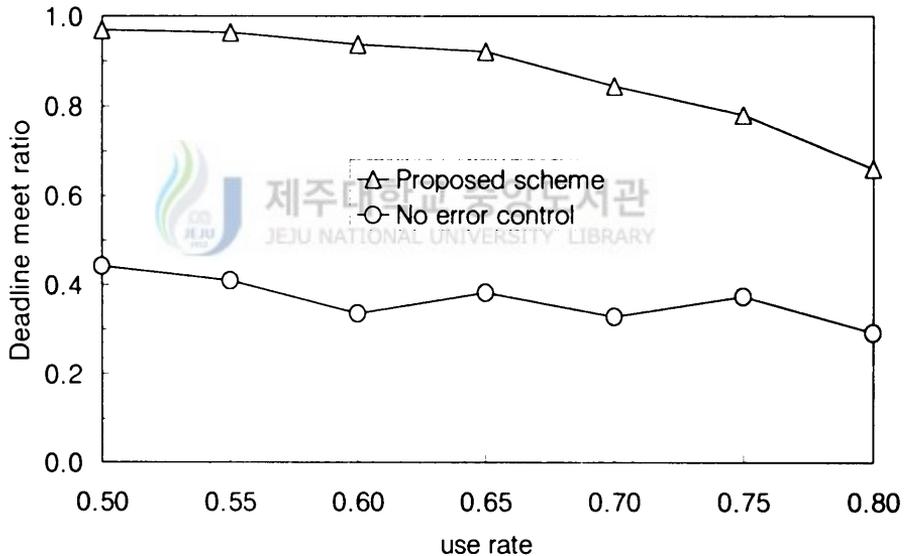


Figure 7. Use rate vs. deadline meet ratio

세 번째 실험은 비실시간 메시지의 부하가 오류 제어에 주는 영향을 측정하기 위하여 비실시간 메시지의 부하를 0에서 0.5까지 변화시켜 가며 종료시한 만족도를 측정하였는데 <Figure 8>에서 나타난 바와 같이 모든 범위에서 제안된 기법이 네트워크의

실시간 성능, 즉 종료시한 만족도를 개선할 수 있으며 전체 네트워크의 부하가 0.8 이상이면 오류 제어를 위한 여분의 대역폭이 감소하기 때문에 개선도는 크게 변하지 않는다.

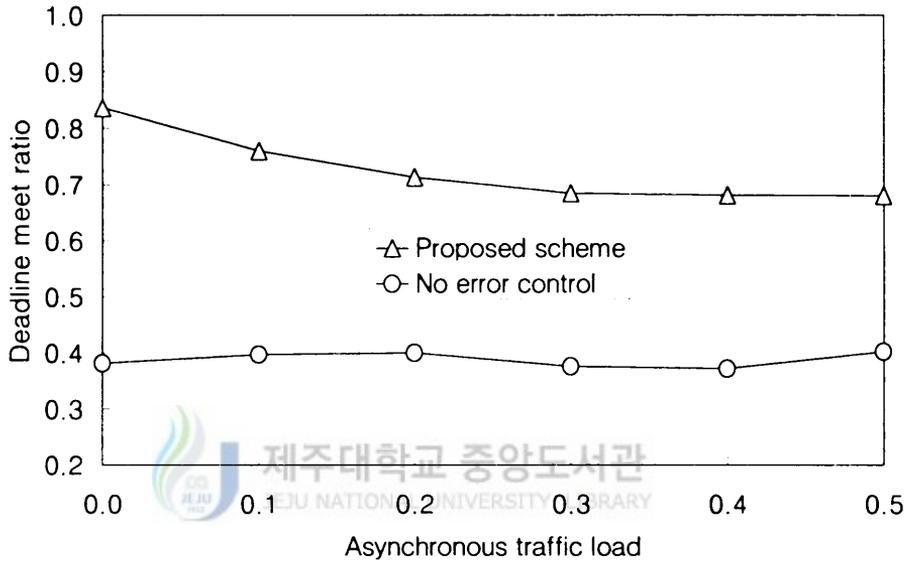


Figure 8. Asynchronous traffic load vs. Deadline meet ratio

VI. 결 론

본 논문은 FDDI 네트워크에 기반한 경성 실시간 통신에서 여분의 대역폭을 이용한 최선 노력 오류제어 기법을 제안하고 SMPL을 이용한 모의실험에 의해 성능을 평가하였다.

제안된 기법은 오류 탐지를 위하여 기본적으로 FDDI 프레임에서 제공하는 오류보고 필드를 사용하며 수신자는 정확한 오류보고 시점을 결정하기 위해 링을 따라 자신이 맞이하는 토큰의 수를 계수한다. 오류의 보고는 비동기 대역폭을 이용하여 송신자에게 보고되며 이후 송신자는 여분의 대역폭, 즉 토큰을 맞이할 때마다 오류 발생 프레임을 재전송한다.

제안된 기법의 성능은 SMPL을 이용한 모의실험에 의해 측정하였으며 이 실험은 각각 매체 오류율, 사용율, 비동기 트래픽에 대한 종료시한 만족도를 포함한다.

이 기법은 오류 제어를 위하여 과할당된 대역폭 혹은 비동기 대역폭을 사용하기 때문에 다른 실시간 메시지의 전송에 간섭을 일으키거나 통신 시스템의 스케줄 가능성에 영향을 주지 않는다. 주어진 메시지 스트림 집합에 대해 최대 29%의 종료시한 만족도 향상을 보이고 있다.

VII. 참고 문헌

- Agrawal, G., B. Chen, W. Zhao and S. Davarn. 1992. Guaranteeing synchronous message deadlines with timed token protocol. *Proc. of IEEE International Conference on Distributed Computing Systems*.
- Arvind, K., Krithi Ramamritham and John A. Stankovic. 1991. A local area network architecture for communication in distributed real-time systems. *Journal of Real-Time Systems*, 3 : 115~117.
- Chen, Biao, Gopal Agrawal and Wei Zhao. 1992. Optimal synchronous capacity allocation for hard real-time communications with the timed token protocol. *Proc. Real-time Systems Symposium*, 198~207.
- Garcia-Molina, H., B. Gao, D. Barbara. 1991. Aggressive transmissions over redundant paths. *Proc. Distributed Computing Systems*, 198-207.
- Gong, F., G. Parulkar. 1996. An application-oriented error control scheme for high-speed networks. *IEEE Trans. Networking*, 4(5). 669~683.
- Kohavi, Z. 1978. *Switching And Finite Automata Theory*, McGraw-Hill.
- Kopetz, H., A. Damm, J. Reisinger, W. Schwabl. 1988. The real-time operating systems of MARS. *Proc. Operating Systems Review*, 141~157.
- Kurose, J.. 1994. Real-time communication on packet-switched networks. *Proc. of IEEE*, 82(1). 115~147.
- Lee, Junghoon and Choelmin Kim. 1999. Design of an error control scheme for hard real-time communication on FDDI networks. *IEEE TENCON*.
- MacDougall, M. H. 1987. *Simulating Computer Systems : Techniques and Tools*, MIT Press.

- Malcolm, Nicholas and Wei Zhao. 1995. Hard real-time communication in multiple-access networks. *Journal of Real-Time Systems*, 37~77.
- Mirchandani, Sonu and Raman Khanna. 1993. *FDDI Technology and Applications*, John Wiley and Sons, Inc..
- Natravali, A. N., W. D. Roome, K. Sabnani. 1990. Design and implementation of a high-speed transport protocol. *IEEE Trans. Communication*, 38(11). 2010~2024.
- Sevcik, K. C. and M. J. Johnson. 1987. Cycle time properties of the FDDI token ring protocol. *IEEE trans. Software Eng.*, SE-13(3). 376~385.
- 정충일, 권도한, 박창윤. 1999. 멀티미디어 통신에서 적시 재전송에 기반한 선택적인 오류 제어 방법. *정보과학회논문지(A)*, 10(26). 1225~1236.



초 록

본 논문은 FDDI에 기반한 경성 실시간 통신을 위한 오류제어 기법을 제안하고 성능을 평가한다. 제안된 기법은 모든 오류에 대해 처리하는 기존 방식들과는 달리 종료시한 이내에 처리 가능한 오류에 대해서 최선 노력 방식으로 수행된다. 오류 탐지를 위해 FDDI 프레임에 포함된 프레임 검사 기능을 이용하였다. 오류 보고는 한 메시지 내에 오류가 발생한 프레임의 리스트를 송신자에게 보내는데 수신자는 토큰의 수신 횟수를 계수하여 보고 시점을 결정하며 오류 보고 메시지는 비동기 트래픽을 이용하여 수행된다. 송신자의 재전송 과정에서 과할당된 대역폭을 이용하기 때문에 다른 실시간 메시지 전송에 간섭을 일으키지 않고 전송 오류에 의한 메시지 손실을 감소시켜 종료시한 만족도를 개선한다. 성능 평가를 위해 SMPL에 기반한 모의 실험을 통해 매체 오류율, 사용률 및 비동기 트래픽 부하 등에 따른 종료시한 만족도를 측정하였으며 그 결과는 다음과 같다. 첫째, 사용률 70 %로 주어진 메시지 집합에 대해 실시간 메시지의 종료시한 만족도를 매체 오류율 10^{-6} 에서 최대 29% 개선하였다. 둘째, 제안된 기법은 사용률이 적을수록 여분의 대역폭이 많으므로 적은 사용률에서 더 큰 종료시한 만족도를 보인다. 끝으로, 비동기 트래픽의 부하에 대해서 종료시한 만족도는 안정적이다. 결국 제안된 방식은 FDDI 뿐만 아니라 라운드 로빈 형태로 매체 접근 제어를 수행하는 다양한 프로토콜을 이용한 실시간 통신에서 전송 오류를 극복할 수 있다.

감사의 글

나의 보족함을 채워 보자는 막연한 생각으로 대학원 문턱을 들어선 게 엇그제 같은데 벌써 이렇게 논문을 마치고 졸업을 하게 됐다고 생각을 하니 참으로 많은 것을 돌아보게 합니다.

우선 이 논문이 있기까지 때로는 격려를, 때로는 질책을 아끼지 않으며 저를 이끌고 지도해 주신 이정훈 교수님께 깊은 감사를 드립니다. 그리고 제가 조교 업무와 학업 사이에서 힘들어 하고 있을 때마다 저를 이해해 주시고 힘이 되어 주신 김철수 교수님과 형님처럼 격려를 아끼지 않아 주신 이봉규 교수님께도 지면을 빌어 감사를 드립니다. 또한 항상 가까이에서 저를 지켜봐 주신 김익찬 교수님과 박경린 교수님의 은혜에도 감사를 드립니다.

많은 시간을 저와 함께 고생해 주었던 미경이와 형장이, 항상 형님같이 많은 조언과 힘이 되어 주신 김봉모 선생님께도 고마움을 전하고 같은 사무실에서 근무하면서 도와 주었던 후배 인혜에게도 늦게나마 고맙다는 말을 전합니다.

그리고 나에게 마음의 격려가 되어 주었던 형과 누나와 동생 용석이, 내가 힘들 때마다 위로가 될 수 있었던 친구들, 항상 가까이에서 나를 지켜봐 주신 많은 분들께 지면을 빌어 고마움을 전합니다.

끝으로 항상 저를 지켜보고 나의 든든한 힘이 되어 주신 아버지, 어머니. 지난 시간 동안 내가 하고자 하는 일에 믿음으로 큰 격려가 되어 주신 부모님께 감사하다는 말과 함께 이 글을 드립니다.