

전송제어 실시간 통신에서의 선택적 재전송에 의한 오류제어 기법

正會員 이 정 훈*

An error control scheme based on selective retransmission for transmission control real-time communication

Jung Hoon Lee* *Regular Member*

요 약

본 논문은 TDMA 구현 이더네트를 중심으로 전송제어 기반 실시간 통신에서의 저렴하고 효율적인 오류제어 기법을 제안하고 성능을 평가한다. 이 기법은 선택적 재전송 기법에 기반하고 있으며 실시간 통신 시스템의 대역폭 할당 기법이 불가피하게 생성하는 여분의 전송가용 시간을 이용하여 오류가 발생한 메시지를 재전송하고 재전송 요청 메시지를 위해서만 추가적인 대역폭을 필요로 한다. SMPL을 이용한 모의실험 결과는 제안된 기법이 주어진 스트림 집합에 대해 종료시한 만족도를 최대 28% 향상시킬 수 있으며 프레임 시간이 작을수록 종료시한 만족도가 증가함을 보이고 있다. 이 기법에 의해 추가적인 하드웨어의 도움없이 저비용으로 메시지 오류제어에 의한 종료시한 만족도를 개선할 수 있다.

I. 서 론

경성 실시간 통신(hard real-time communication)의 목적은 네트워크에서 오류가 발생하지 않는 한 모든 실시간 메시지들의 시간 제약조건(time constraint)을 만족시키는 것으로서[1], 종료시한(deadline)과 같은 메시지의 시간 제약조건을 만족시키기 위해서는 메시지들의 전송이 정확하게 스케줄되어야 하는데 이는 네트워크 프로토콜 계층구조상 매체 접근 제어(MAC: Medium Access Control) 계층의 기능에 속한다. 모든 노드가 공유된 네트워크를 통해 연결된 다중접근 네트워크(multiple access network)에서 MAC 계층의 기능은 분산된 노드들로부터의 네트워크 접근을 증대

하여 주어진 시점에 어느 노드가 네트워크를 사용할지를 결정하는 것이다. 다중접근 네트워크에서 사용되는 실시간 통신 방식은 크게 접근중재 방식(access arbitration scheme)과 전송제어 방식(transmission control scheme)의 두 범주로 나눌 수 있는데, 접근중재 방식은 공유된 채널을 통해 노드가 전송할 수 있는 시점을 결정하는데 중점을 두고 있는 반면, 전송제어 방식은 각 노드가 매체를 통해 전송할 수 있는 시간 구간의 길이를 결정하는데 중점을 두고 있다[2]. 즉, 접근중재 방식은 공유된 네트워크에 대한 접근 요구가 동시에 발생했을 때 이를 우선순위(priority)나 여유시간(laxity)에 준하여 해결하는 방식으로서 우선순위에 기반한 비율단조(rate-monotonic) 스케줄링 방식이 이에 포함된다[3]. 이와는 대조적으로 전송제어 방식은 각 노드가 네트워크에 배타적으로 접근할 수 있는 시간의 길이를 노드의 메시지 스트림의 요

* 제주대학교 전산통계학과
論文番號: 96308-0930
接受日字: 1996年 9月 30日

구사함에 따라 결정하여 메시지의 시간 제약조건을 만족시키는 방식으로서 FDDI(Fibre Distributed Data Interface)[4]나 토큰 버스[5], TDMA 구현 이더네트(TDMA-implemented Ethernet)[6]처럼 각 노드가 라운드-로빈(round-robin) 형태로 메시지를 전송하는 프로토콜에 적합하다.

실시간 통신 프로토콜들이 메시지들의 시간 제약조건을 만족시킬 수는 있지만 전송 오류는 불가피하게 발생하며 전송 오류는 메시지의 손실을 의미한다. 전송 오류는 매체 오류율과 외부 자극에 의한 일시적인 교란에 의해 발생하는데 예를 들면 10 Mbps 이더네트의 매체 오류율은 비트당 10⁻⁶이다. 일반적인 네트워크 통신에서 메시지의 전송중 발생한 오류를 탐지하고 보정하는 오류제어 기능은 논리적 링크 제어(LLC: Logical Link Control) 계층의 기능에 해당한다. 그러나 기존의 오류제어 기능은 그 과정이 메시지 전송 시간을 연장하게 되어 메시지들이 시간 제약조건을 갖는 실시간 통신에는 적합하지 못하다.

선택적 재전송(selective retransmission) 방식은 비실시간 데이터 통신에서 널리 사용되는 오류제어 기법으로서 메시지의 전송이 끝난 후 확인(acknowledgment) 혹은 재전송 요청(retransmission request) 메시지에 의해 수신자가 송신자에게 수신 확인이나 재전송 요구를 하며 송신자는 이에 의해 메시지를 재전송한다[7]. 이 기법은 전송중 오류가 발생한 메시지만 선택적으로 재전송하므로 대역폭을 효율적으로 이용할 수는 있지만 메시지의 전송 지연 시간도 메시지 손실의 원인이 되는 실시간 통신 시스템에는 적합하지 못하다. 즉, 확인 혹은 재전송 요청 메시지의 전송과 이에 따르는 메시지의 재전송 과정이 메시지 전송 시간을 연장하여 메시지의 종료시한을 초과하게 할 수 있으며 재전송되는 메시지가 일반적인, 즉 재전송되지 않는 실시간 메시지 전송을 방해할 수 있다. 또한 재전송 요청 메시지가 일반 메시지들과 함께 스케줄되어야 하며 전송된 메시지들을 위해 버퍼가 할당되어야 하므로 실시간 시스템에서는 적용하기 어렵다.

전송제어에 기반한 실시간 통신에서는 대역폭 할당 기법이 최악의 경우를 고려하여 용량 벡터(각 노드가 자신의 전송 시간을 맞이하여 메시지를 전송할 수 있는 시간의 길이)를 결정하기 때문에 각 스트림은 여분

의 대역폭을 갖게 된다. 이 과할당된 대역폭을 통한 재전송은 메시지의 종료시한을 만족시킬 수 있으며 다른 메시지의 전송에 영향을 주지 않는다. 또한 재전송 요청 메시지는 기존의 스트림에 약간의 낭비 시간을 부가하여 대역폭을 사전에 할당할 수 있으므로 일반 메시지들과 함께 스케줄될 수 있다. 마지막으로 메시지는 오직 그 종료시한 내에서만 유효하기 때문에 종료시한 이내에 재전송될 가능성이 있을 때에만 버퍼에 유지하면 되므로 각 노드는 메시지 재전송을 위한 버퍼를 충분히 갖고 있다고 할 수 있다. 결국 선택적 재전송 방식은 전송제어에 기반한 실시간 통신 시스템에 적용이 가능하며 이에 의해 적은 낭비 시간으로 전송 오류를 처리할 수 있다.

본 논문은 전송제어 실시간 통신 기법의 한 예인 TDMA 구현 이더네트에 적합한 선택적 재전송 기법을 제안하고 그 성능을 측정한다. 이를 위해 선택적 재전송 기법을 적용한 실시간 통신의 성능을 설계하고 실시간 메시지와 확인 혹은 재전송 요청 메시지가 결합된 대역폭 할당 기법을 제시한 후 메시지 손실을 측정에 의해 제안된 오류제어 기법의 성능을 분석한다. 본 논문의 구성은 다음과 같다. 2장에서는 실시간 통신 시스템에서의 오류제어 기법에 관련된 기존의 연구를 기술함과 아울러 오류제어 기법을 적용할 네트워크인 TDMA 구현 이더네트에 대해 소개한다. 3장에서는 메시지 모델의 가정에 이어 TDMA 구현 이더네트에서의 선택적 재전송 기법과 대역폭 할당 기법을 설명한다. 4장에서는 모의실험 결과를 보이고 측정된 성능에 대해 분석하며 마지막으로 5장에서는 논문의 결론과 추후 과제에 대해 기술한다.

II. 관련 연구

2.1 오류제어 기법

실시간 통신에서는 두 종류의 오류제어 기법이 널리 사용되는데 이는 시간적 여분(temporal redundancy)과 공간적 여분(spatial redundancy)을 이용하는 방식이다[8]. 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 타임아웃에 뒤따르는 재전송에 의해 오류를 제어한다. 재전송 요청이나 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 여분 방식의 예로서 MARS 시스템을 들 수 있는데, 이 시스템에서는 그림 1에서 보는 바와 같이 오류 발생 여부에 관계없이 모든 메시지들이 이중으로 중복되어 전송된다[9].

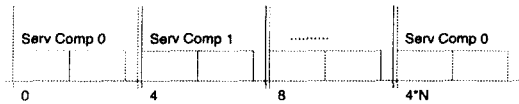


그림 1 MARS 슬롯의 구조
Fig. 1 Structure of MARS slot

반면 공간적 여분을 사용한 방식에서는 중복된 메시지들이 서로 다른 채널이나 네트워크를 통해 전송되며, 메시지의 종료시한이 촉박하여 재전송이나 중복 전송을 허용할 수 없을 경우에 적합하다[10]. 이 방식은 항상 메시지들이 다중으로 중복되어 전송되므로 메시지들에 의해 요구된 대역폭의 중복 배수 만큼의 대역폭을 필요로 하게 되어 상당한 대역폭의 낭비를 초래한다. 이상의 기법들 이외에 오류 보정 코드에 의한 전향 오류제어 방식이 효율적으로 보이지만 정보의 낭비 시간이 크게 되며 처리시간이 길어지게 된다.

2.2 TDMA 구현 이더네트

이더네트는 전세계적으로 널리 사용되고 있는 네트워크이기는 하지만 프로토콜의 특성상 메시지의 충돌 가능성을 갖고 있으며 충돌의 해결과정이 예측가능하지 못하므로 실시간 응용에는 적합하지 못하다. TDMA 구현 이더네트는 이더네트로 하여금 실시간 통신을 지원할 수 있도록 하는 기법의 하나로서 이더네트에 TDMA 프로토콜을 구현하여 실시간 통신의 필수적인 특성인 예측가능성을 지원하도록 한다. 즉 전역적으로 동기화된 시계를 기반으로 하여 각 노드

로 하여금 정해진 시간에 정해진 슬롯 시간 만큼 메시지를 전송하도록 한다. 그림 2에서 보는 바와 같이 TDMA 구현 이더네트의 시간축은 프레임 시간 F 로 균등하게 분할되며, 노드의 수 n 이라 할 때 하나의 프레임 시간은 제어 슬롯 T_c 와 각 노드에 할당된 슬롯 시간 H_i ($1 \leq i \leq n$)로 구성된다. 슬롯간격 시간은 TDMA 프로토콜 구현에 따른 낭비 시간으로서 시계 동기화 오류, 전파지연 시간(propagation delay), 패킷화 낭비 시간 등을 포함한다. 프레임내 총 낭비 시간 γ 는 n 과 슬롯간격 시간의 곱으로 계산된다.

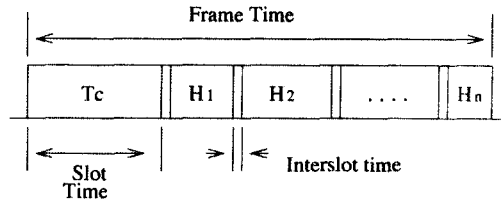


그림 2. TDMA 구현 이더네트 프레임의 구조
Fig. 2 Frame structure of TDMA-implemented Ethernet

H_i 의 크기는 각 메시지 스트림의 시간 제약조건을 만족시키도록 가변적인 대역폭 할당 기법(variable bandwidth allocation scheme)에 의해 결정되며 대역폭 할당은 오프라인 시나 새로운 실시간 스트림이 설정될 때 수행된다[11]. 실시간 연결 설정은 제어 슬롯 T_c 를 통해 수행되는데 이 제어 슬롯은 대역폭을 재할당하거나 시계의 동기화 등 네트워크의 관리 및 운영을 위해 설정된 슬롯이다[12]. 결국 대역폭 할당 기법은 각 메시지 스트림에 대해 주기 P_i , 전송시간 C_i 와 같은 트래픽 특성을 기반으로 하여 프레임 시간 F 와 슬롯 시간 H 를 결정한다. 그러나 대역폭 할당 기법은 오류 제어를 고려하지 않았으며 오류제어를 반영하기 위해서는 확인 혹은 재전송 요청 메시지나 재전송 메시지를 위한 대역폭을 추가적으로 할당하여야 한다.

대역폭 할당 기법은 최악의 상황을 고려하여 H 를 결정하기 때문에 각 스트림은 주기내에서 여분의 대역폭을 갖게 된다. 최악의 상황이란 그림 3에서 보는 바와 같이 스트림에 속한 메시지가 슬롯 시간중 도착

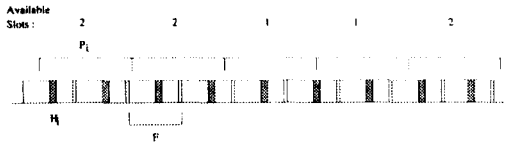


그림 3 대역폭 할당과 전송가용시간
Fig. 3 Bandwidth allocation and available transmission time

한 경우로서 스트림은 그 슬롯을 사용할 수 없게 되므로 주기내에서 맞이하는 슬롯의 수가 하나 감소된다. 그러나 위의 상황을 제외하고는 각 스트림은 하나의 슬롯을 여분으로 갖게 되며 이 슬롯은 낭비되거나 비실시간 메시지의 전송에 사용된다. 본 논문은 이 여분의 슬롯을 바탕으로 한 오류제어 기법을 제시하고 대역폭 할당 기법을 수정하여 오류제어 기능을 반영하도록 한다.

III. 오류제어 기법

3.1 메시지 모델

본 논문에서는 동기(synchronous)와 비동기(asynchronous)의 두 메시지 계층을 가정한다[13]. 동기 메시지는 특정 스트림에 속한 메시지들로서 주기적으로 도착하며 다음 메시지가 도착하기 전에 전송이 완료되어야 하는데 이는 각 메시지의 종료시한이 해당 스트림의 주기가 됨을 의미한다. 반면 비동기 메시지는 시간 제약조건을 갖고 있지는 않으며 동기 메시지가 없을 때에만 슬롯을 통해 전송될 수 있다. 실시간 통신 시스템에서 MAC과 대역폭 할당 기법의 기능은 동기 메시지의 시간 제약조건을 만족시키는 것으로서 본 논문에서는 모든 노드 i 가 주기 P_i 와 전송시간 C_i^{max} 를 갖는 동기 메시지 스트림 S_i 를 하나 갖고 있다고 가정하며 이 가정은 가상 노드 개념을 도입하여 일반적인 경우, 즉 한 노드가 다수의 스트림을 갖는 경우로 확장될 수 있다[13]. 메시지는 H_i 크기의 서브패킷으로 나뉘어 전송되는데 이때 DAT(m)을 주기내에서 m 번째 가용 슬롯을 통해 전송되는 서브패킷으로 정의한다. 또한 선택적 재전송을 적용하기 위한 추가적인 메시지로 SEL(m)과 RET(m)을 정의하는데 SEL(m)은 수신자 노드에 의해 전송되어 DAT(m)의

재전송을 요청하는 기능을 갖고 있으며 RET(m)은 DAT(m)의 재전송 메시지이다. 송신자가 한 주기 내의 메시지 전송을 완료하면 REQ를 보내 수신자로 하여금 송신자의 슬롯 시간을 통해 SEL(m)을 보낼 수 있도록 한다.

3.2 오류제어 기법

DAT(m), SEL(m) 및 REQ는 예약된 대역폭을 이용하여 전송되는 반면 RET(m)은 여분의 슬롯 시간이 있을 때에만 전송된다. 오류제어는 스트림의 각 주기마다 수행되며 그 과정은 다음과 같다. 송신자 노드는 주기내의 모든 서브패킷의 전송을 완료하면 REQ를 수신자에게 보낸다. 이 메시지는 자신의 슬롯 시간내에서 네트워크에 대한 접근 권한을 수신자에게 넘겨주는 기능을 갖는다. 수신 노드는 f 번째 서브패킷에 오류가 발생한 경우 SEL(f)를 보내며 오류가 발생하지 않은 경우는 SEL 메시지를 보내지 않는다. 송신자는 만약 여분의 슬롯이 있으면 RET(m)을 보내 한 주기를 완료한다. 이 과정에서 주기내에 두 개 이상의 서브패킷에 오류가 발생한 경우나 하나의 서브패킷에 오류가 발생했으나 여분의 슬롯이 없는 경우에는 오류를 처리할 수 없다. 이는 메시지를 재전송한다 하여도 종료시한을 초과하기 때문에 오류제어에 의미를 갖지 못하기 때문이다. 송신 노드와 수신 노드는 모두 송신자의 주기내에서 전송되는 메시지의 양에 해당하는 송수신 버퍼를 유지하여야 한다. 이러한 과정은 그림 4에 나타나 있다.

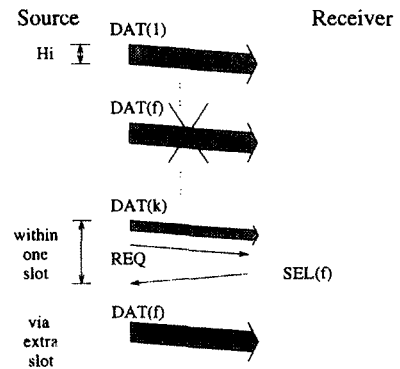


그림 4 오류제어 기법의 메시지 흐름
Fig. 4 The message flow of error control scheme

3.3 대역폭 할당

각 스트림에 대해 오류제어 기법을 위해 추가적인 대역폭이 할당되어야 한다. 추가적으로 요구되는 대역폭은 REQ 및 SEL(m) 메시지를 위한 네트워크 시간과 이들 메시지의 전파지연 시간을 포함한다. 결국 각 메시지 스트림의 최종 대역폭 요구는 d 를 전파지연 시간이라 할 때, 식(1)과 같이 계산될 수 있다.

$$C_i = C_i^{orig} + REQ + d + SEL(n) + d \quad (1)$$

이와 같은 전송시간 보정에 의해 가변적인 대역폭 할당 기법이 적용되며 이 기법은 프레임 시간의 범위와 제약조건들로 구성된다[10].

[프레임 시간의 범위] 주어진 스트림 집합에 대해 모든 스트림의 시간 제약조건을 만족시키려면 사용율을 U , 주기의 최소값을 P_{min} 라 할 때, 프레임 시간 F 는 부등식 (2)를 만족시켜야 한다.

$$\frac{\gamma}{1-U} \leq F \leq \frac{P_{min}}{2}, \quad \text{단 } U = \sum \frac{C_i}{P_i} \quad (2)$$

주어진 스트림 집합이 실시간 요구사항을 맞추려면 위에서 계산된 프레임 시간의 범위내에서 슬롯 시간의 벡터 H 는 다음의 제약 조건에 따라 결정된다.

[프로토콜 제약 조건] 각 노드에게 할당된 슬롯 시간의 합은 메시지 프레임 시간의 가용 부분보다 작아야 한다.

$$\sum_{i=1}^n H_i \leq F - \gamma \quad (3)$$

[사용율 제약 조건] 사용율 U , 프레임 시간 설정에 따른 낭비 시간 O_f , 슬롯간격 시간에 의한 낭비율의 합은 1.0보다 작아야 한다.

$$O_f + U + \frac{\gamma}{F} \leq 1.0 \quad (4)$$

부등식 (4)에서 O_f 는 스트림의 주기를 프레임 시간으로 나눈 나머지 시간이 메시지의 전송에 사용될 수 없기 때문에 발생하는 낭비 시간으로서 다음과 같이 계산된다.

$$O_f = \sum_{i=1}^n \frac{P_i - \left\lfloor \frac{P_i}{F} \right\rfloor \cdot F}{P_i}$$

[중요시한 제약 조건] 동기 메시지들의 종료시한을 만족시킬 수 있도록 슬롯 시간의 크기가 결정되어야 한다. 슬롯 시간의 크기에 의해 각 메시지 스트림이 주기내에서 사용할 수 있는 네트워크 시간의 크기가 결정되는데 스트림 i 의 주기내 전송가용 시간 X_i 는 메시지 전송시간 C_i 보다 커야 한다.

$$X_i \leq C_i \quad (5)$$

부등식 (5)에서 한 주기 내에서 각 스트림이 자신의 슬롯을 맞이하는 최소 횟수는

$$\left\lfloor \frac{P_i}{F} \right\rfloor - 1$$

이므로 한 주기내 최소 전송 가용시간 X_i 는 다음과 같이 계산된다.

$$X_i = \left(\left\lfloor \frac{P_i}{F} \right\rfloor - 1 \right) \cdot H_i$$

[슬롯의 크기] 스트림 S_i 의 슬롯의 크기 H_i 가 부등식 (6)과 같이 결정된다면 S_i 의 시간 제약조건을 만족시킬 수 있다.

$$H_i \geq \frac{C_i}{\left\lfloor \frac{P_i}{F} \right\rfloor - 1} \quad (6)$$

이상의 기존 제약조건 이외에 단일 슬롯 제약조건이 추가되며, 이는 재전송 요청 과정, 즉 REQ 전송과 SEL(f)의 전송이 한 슬롯 내에서 완수되도록 한다.

[단일 슬롯 제약 조건] REQ와 SEL(m)을 전송하는 프로시저는 하나의 슬롯내에서 이루어져야 한다. R_i 를 각 주기의 메시지 C_i (재전송 요청 과정 포함) 중 마지막 슬롯을 통해 전송되는 서브패킷의 크기라고 할 때 (7)의 제약조건을 만족시켜야 한다.

$$R_i \geq REQ + SEL(f) + 2 \cdot d \quad (7)$$

[중명] 주기의 마지막 슬롯을 통해 재전송 과정이 일어나는데 이 과정은 하나 이상의 슬롯에 분산될 수 없다. 마지막 슬롯은 메시지의 남은 부분과 재전송 요청 과정으로 구성되므로 이들의 합 R_i 가 재전송 과정에 필요한 네트워크 시간보다 커야 한다. 재전송 과정은 REQ 메시지와 SEL(f) 메시지로 구성되므로 이 때

```

procedure frame.slot()
input :
  { $C_i$ } : the transmission times of message set { $S_i$ }
  { $P_i$ } : the periods of message set { $S_i$ }

   $F_{min} \leftarrow \frac{1}{(1-\gamma)}$ 
   $F_{max} \leftarrow \frac{1}{\gamma}$ 
  compute GCM of { $P_i$ }
  /* first multiplier of GCM which is larger than  $F_{min}$  */
   $F \leftarrow \lceil \frac{F_{min}}{GCM} \rceil \cdot GCM$ 
  while ( $F \leq F_{max}$ ) {
    if (nalloc( $C_i, P_i$ ) == success) exit
    else  $F \leftarrow F + GCM$ 
  }
  return (no schedule)
end procedure

procedure nalloc()
input : { $C_i$ }, { $P_i$ }
  /* utilization constraint */
  if ( $O_f + U + \frac{\gamma}{1-\gamma} \geq 1.0$ ) return (fail)
  for all  $i$ 
     $H_i \leftarrow \frac{C_i}{(F-\gamma)}$ 
    /* one slot constraint */
    if ( $R_i \leq 2d + SEL(m_i) + REQ$ ) return (fail)
    /* protocol constraint */
    if ( $\sum H_i \leq F - \gamma$ ) return (success)
  else return (fail)
end procedure
    
```

그림 5 대역폭 할당 기법
Fig. 5 The bandwidth allocation scheme

시지가 전송되는데 필요한 시간은 이들의 크기와 전파 지연 시간으로 구성된다.

이상의 제약 조건들을 기반으로 하여 대역폭은 그림 5와 같이 할당될 수 있다.

[예] 표 1에서 보이는 바와 같은 전송 특성을 갖는 5개의 스트림이 있다고 하고 재전송 요청 과정을 위해 필요한 대역폭, 즉 REQ 메시지와 SEL(f)의 전송을 위해 필요한 대역폭이 총 30 비트 시간이라고 한다면 각 스트림이 필요로 하는 전송시간은 30씩 증가한다. 이 스트림 집합에 대해 부등식 (2)에 의해 계산된 프레임 시간의 최소값과 최대값은 각각 138과 12130이

표 1. 스트림 집합의 예
Table 1. Example stream set

	τ_1	τ_2	τ_3	τ_4
P_i	35400	79410	24240	96860
C_i^{REQ}	7020	960	1200	14960
C_i	7050	990	1230	14990

표 2 할당된 슬롯 시간(프레임 시간 = 2200)
Table 2 Allocated slot time(Frame time = 2200)

	τ_1	τ_2	τ_3	τ_4
H_i	470	30	123	349
X_i	7050	1020	1230	15007

다. 대역폭 할당 기법은 프레임 시간 2200에서 가능한 대역폭 할당을 찾을 수 있으며 이에 의해 결정된 슬롯 시간 H_i 와 전송가용 시간 X_i 는 표 2에 나타나 있다.

N. 성능 평가 및 분석

SMPL[14]을 이용한 모의실험에 의해 제안된 오류 제어 기법의 성능을 두 가지 측면에서 측정하였다. 첫번째 실험은 오류율에 대비한 종료시한 만족도를 측정하는 것으로 그림 6에 그 결과가 나타나 있다. 이 실험에서 사용율 60에서 80%까지의 40개의 스트림 집합을 생성하였는데 이들은 각각 임의의 스트림 갓

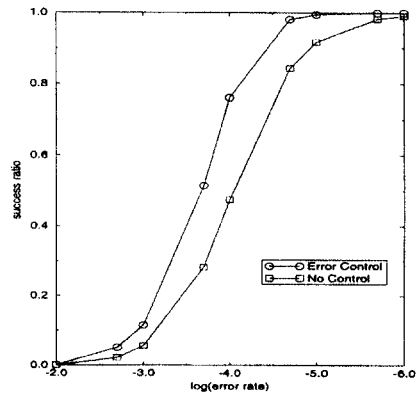


그림 6 매체 오류율에 대비한 종료시한 만족도
Fig. 6 Deadline meet ratio according to error rate

수를 가지며 각 스트림의 주기는 1000에서 10000, 전송 시간은 10에서 2000 까지 임의의 값을 갖는다. 종료시한 만족도는 주기내에 전송 완료된 메시지의 수를 모의실험 시간 내에 생성된 메시지의 수로 나누어 계산된다. 제안된 오류제어 기법은 매체 오류율 10^{-4} 에서 최대 28%의 만족율 증가를 보이고 있다.

오류율에 따른 종료시한 만족도 측정에 관련된 또 하나의 실험으로서 이중 링크를 사용한 오류제어와 비교하였는데 이 방식은 매체의 오류율을 이중 전송에 의해 감소시킨 효과를 갖게 되어 단일 네트워크의 오류율이 e 라면 이중화된 네트워크의 매체 오류율은 e^2 이 되므로 전체 오류율이 크게 감소하기는 하지만 네트워크의 설치 비용이 2배로 증가하고 대역폭의 낭비가 심하게 된다. 이중 링크와의 비교 결과는 그림 7에 나타나 있으며 매체의 오류율이 10^{-5} 보다 작아진다면 두 방식의 이는 거의 없어지는데 이는 제안된 오류제어 방식이 실시간 통신에서 저비용으로 오류를 제어할 수 있음을 보여준다.

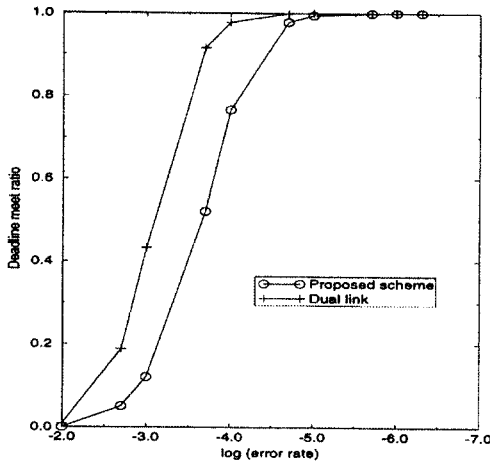


그림 7 이중 링크와의 비교
Fig. 7 Comparison with dual link communication

스트림 집합의 사용율이 낮을 때에는 여러 가지의 대역폭 할당이 가능하고 이들은 각각 다른 프레임 시간을 갖는다. 따라서 두번째 실험에서는 프레임 시간이 종료시한 만족율에 주는 영향을 측정하였으며 이 실험은 3장의 할당 예에 있는 스트림 집합을 가정하

였는데 이 결과는 그림 8에 나타나 있다. 그림 8은 프레임 시간이 작게 설정될 때 메시지의 종료시한 만족율이 증가함을 보이고 있다.

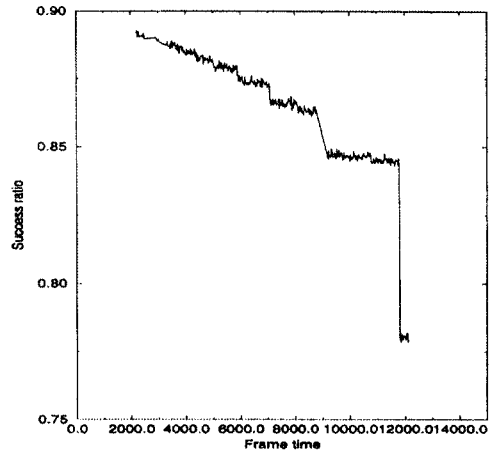


그림 8 프레임 시간에 따른 종료시한 만족도
Fig. 8 Deadline meet ratio according to frame time

이를 분석하기 위해 스트림 S_i 가 주기내에서 k 개의 서브패킷으로 나뉘어 전송되고 이 서브패킷의 크기를 l , 오류율을 e , 여분의 슬롯이 있을 확률을 라고 한다면 $k \cdot l$ 은 상수이며 이때의 종료시한 만족율 $R_{i,0}$ 는 식(8)과 같이 계산된다.

$$R_{i,0} = C_0(1-le)^k + C_1le(1-le)^{k-1} \cdot \delta = (1-le)^{k-1}(1-le + kle\delta) \quad (8)$$

$R_{i,0}$ 은 k 가 상대적으로 작을 때 극소값을 갖게 되는데 이는 $R_{i,0}$ 가 k 의 증가에 따라 같이 증가함을 의미하므로 결국 프레임 시간이 작을수록 종료시한 만족도가 증가한다. 또, G_i 를 스트림 S_i 가 요구하는 최소 오류율이라 할 때 만약 G_i 가 $R_{i,0}$ 보다 작다면 오류제어를 위해 추가적인 대역폭이 할당되어야 한다. 이는 C_i 가 증가되어야 함을 의미하며 c 개의 슬롯이 추가로 할당된다면 C_i 는 $c \cdot H_i$ 만큼 증가하는데 S_i 가 c 개의 슬롯을 추가로 얻었다고 한다면 이때의 오류율 $R_{i,c}$ 는 식(9)와 같이 계산된다.

$$R_{i,c} = \sum_{j=0}^c ({}_k C_j)(le)^j (1-le)^{c-j} + {}_k C_{c+1}(le)^{c+1} (1-le)^{k-c-1} \cdot \delta \quad (9)$$

따라서 S_i 의 오류 요구사항을 만족시키려면 부등식 (10)을 만족시키는 c 값을 결정하여야 한다.

$$R_{i,c} \geq G_i \quad (10)$$

V. 결 론

본 논문에서는 전송제어에 기반한 실시간 통신에서의 오류제어 기법을 제안하고 성능을 평가하였다. 제안된 오류제어 기법은 선택적 재전송 기법에 기반하고 있으며 전송 제어 실시간 통신 시스템에 적용이 가능한데 이는 종료시한, 즉 주기 내에서 메시지의 재전송이 이루어질 수 있기 때문이다. 전송제어 기반 실시간 통신에서 대역폭 할당 방식에 의해 불가피하게 발생한 과할당 슬롯 시간을 통해 서브패킷의 재전송이 이루어지므로 이 오류제어 기법은 오로지 재전송 요청 메시지의 전송에 필요한 대역폭만을 추가적으로 요구한다. 따라서 시간적 혹은 공간적 여분에 의한 오류제어 기법에 비해 대역폭의 낭비가 상대적으로 작게 되며 이 추가 대역폭은 가변적인 대역폭 할당 기법에 의해 쉽게 할당될 수 있다. 이 기법은 FDDI나 다른 시간제약 토큰 프로토콜(timed token protocol)과 같이 대역폭 할당 기법이 용량 벡터를 과할당하는 실시간 통신 기법에 적용될 수 있다.

메시지 스트림이 요구하는 오류율이 네트워크가 제공할 수 있는 오류율보다 작은 경우 추가적인 대역폭이 할당되어야 하는데 대역폭은 슬롯 시간 단위로 추가 할당되며 요구된 오류율을 만족시키는 추가 대역폭의 양을 계산할 수 있다. 추후의 연구 과제로는 다중 전송 스트림에 대한 효율적인 오류제어 기법과 이중 혹은 다중의 TDMA 구현 이더네트를 이용한 대역폭 할당과 오류제어 기법에 대한 연구가 진행될 것이다.

참 고 문 헌

1. James F. Kurose et. al., "Real-time communication in packet-switched networks," *Proc. of the IEEE*, Vol.

82, No. 1, pp.122-139, Jan. 1994.

2. Malcolm and Wei Zhao, "Hard real-time communication in multiple-access networks," *Journal of Real-Time Systems*, pp.37-77, 1995.
3. J. K Strosnider, T. E. Marchok, "Responsive, deterministic IEEE 802.5 token ring scheduling," *Journal of Real-Time Systems*, pp.133-158, 1989.
4. Nicholas Malcolm and Wei Zhao, "Guaranteeing synchronous messages with arbitrary deadline constraints in an FDDI networks," *IEEE Proc. of the Conference on Local Computer Networks*, pp.186-195, 1993.
5. Chitra Venkatramani and Tzi-cker Chiueh, "Supporting real-time traffic on Ethernet," *Proc. of IEEE Real-Time Systems Symposium*, pp.282-286, Dec. 1994.
6. 이정훈, 신현식, "이더네트에 기초한 실시간 통신을 위한 가변적인 대역폭 할당 기법," 한국통신학회논문지, 제 20 권 2 호, pp.383-394, 1995년 2월.
7. Mischa Schwartz, *Telecommunication Networks: Protocols, Modeling and Analysis*, Addison Wesley Publishing Company, 1987.
8. K. Arvind, Krithi Ramamritham and John A. Stankovic, "A local area network architecture for communication in distributed real-time systems," *Journal of Real-Time Systems*, Vol. 3, pp.115-147, May 1991.
9. Hermann Kopetz, A. Damm, J. Reisinger and W. Schwabl, "The real-time operating systems of MARS," *Proc. of Operating Systems Review*, pp.141-157, 1988.
10. H. Garcia-Monila, B. Bao, and D. Barbara, "Aggressive transmissions over redundant paths," *Proc. of IEEE International Conference on Distributed Computing Systems*, pp.198-207, 1991.
11. Junghoon Lee and Heonshik Shin, "A variable bandwidth allocation scheme for Ethernet-based real-time communication," *Proc. of the First International Workshop on Real-Time Computing Systems and Applications*, pp.28-32, Dec. 1994.

12. Junghoon Lee, Heonshik Shin, "A bandwidth reallocation scheme for Ethernet-based real-time communication," *Proc. of Second International Workshop on Real-Time Computing Systems and Applications*, pp.28-33, Oct. 1995.
13. S. Mukherjee, D Saha, M. C. Sakena and S. K. Tripathi, "A bandwidth allocation scheme for time constrained message transmission on a slotted ring LAN," *Proc. of Real-Time Systems Symposium*, pp.44-53, 1993.
14. M. H. MacDougall, *Simulating Computer Systems: Techniques and Tools*, MIT Press, 1987.



이 정 훈(Jung Hoon Lee) 정회원

1988년:서울대학교 컴퓨터공학과
공학사

1990년:서울대학교 컴퓨터공학과
석사

1996년:서울대학교 컴퓨터공학과
박사

1990 ~ 1991, 1996:대우통신 근무

1997년~현재:제주대학교 자연대학 전산통계학과 교수

e-mail:jhlee@venus2.cheju.ac.kr