論文 95-2-2-10

이더네트에 기초한 실시간 통신을 위한 가변적인 대역폭 할당 기법

正會員 李 政 勳*, 申 玹 植*

TDMA Implementation and Bandwidth Allocation Scheme on Ethernet for Real-Time Communication

Jung Hoon Lee*, Heon Shik Shin* Regular Members

要 約

본 논문에서는 이더네트 상에서 TMDA 구현에 의하여 분산 실시간 시스템을 구현함에 있어 트래픽 특성에 따라 가변적인 대역폭 할당 방식을 제안한다. 트래픽 특성은 메시지 스트림의 주기 및 전송시간, 그리고 TDMA 구현에 따르는 오버헤드 등으로서 슬롯시간 및 프레임 시간을 결정할 때 인자로 사용한다. 슬롯시간이 고정되어 있다면 메시지의 전송시간이 슬롯시간 보다 작은 경우에도 고정된 크기의 슬롯이 할당되므로 대역폭이 낭비되는데 반하여 가변적인 대역폭 할당 방식에서는 이러한 대역폭 낭비를 줄일 수 있다.

ABSTRACT

In this paper, we develop and analyze a variable bandwidth allocation scheme according to the traffic characteristics in constructing distributed real-time systems based on TDMA-implemented Ethernet. The bandwidth of Ethernet can be used efficiently by the variable bandwidth allocation scheme which considers the traffic characteristics which include the period and transmission time of the message stream and the overhead added due to TDMA implementation.

Ⅰ. 서 론

공정제어 시스템, 공장자동화 시스템, 그리고 감시제어 시스템 등과 같은 분산 실시간 시스템은 일반적으로 계산 노드들과 지역 통신망(LAN : Local Area Network)으로 구성되며, 노드간의 메시지 전송은 이 통신망상에서 이루어진다. 실시간 메시지들은 종료시한 내에 전송이 완료되어야 한다는 시간제약 조건을 가지며, 망은 이를 만족시키기 위해 예측가능하고 확장가능한 실시간 통신 메카니즘을 제공하여야 한다. 이더네트는 지난 20여년 동안 안정된 기술로서 다양한 목적으로 널리 사용되어 왔으나, 프로토콜의

* 서울대학교 컴퓨터공학과
論文番號 : 94254
接受日字 : 1994年 9月 22日

속성상 충돌 및 그 해결과정에서 비실시간적 동작 특성을 갖는다(1). 이에 비실시간적인 요소를 상위 프로토콜 계층에 의해 보완할 수 있다면 분산 실시간 시스템을 구성하는데 있어서 널리 사용될 수 있을 것이다. 예를 들어, 분산실시간 시스템 운영체제인 MARS에서는 이더네트에 시분할 접근 방식(TDMA : Time Division Multiple Access)에 의하여 이더네트에 실시간 특성을 부여하려고 시도하였다(2,3).

일반적으로 TDMA는 슬롯으로 구성되고 각 슬롯은 시스템 내의 노드들에게 하나씩 할당되며, 각 노드들에게 할당된 슬롯이 모여 한 프레임을 이룬다. 각 노드는 한 프레임 시간에서 하나의 슬롯을 가지며, 노드는 자신의 슬롯시간에만 메시지를 전송할 수 있으므로 충돌이 발생하지 않는다. 그러므로 이더네트에 실시간 특성을 부여하기 위해서는 TDMA와 같은 방식이 바람직하다. 그런데 TMDA를 구현하는 과정에서 슬롯시간과 프레임 시간은 그 슬롯을 할당받는 노드의 메시지의 특성과 무관하게 일률적인 시간으로 고정된다. 그 결과 각 노드에게 할당된 슬롯 시간이 모두 같게 되어 대역폭의 낭비를 초래하게 된다. 즉, 노드가 전송하는 메시지의 전송시간이 슬롯 시간보다 작은 경우에도 고정된 크기의 슬롯이 할당되므로 그 차이만큼 낭비시간이 생긴다. 한편 TDMA가 이더네트 상에서 구현된다면 슬롯 시간이나 프레임 시간이 고정될 필요가 없다. 또한 실시간 시스템에서는 메시지 스트림의 주기와 전송시간 등이 사전에 알려져 있으므로 이러한 특성들을 슬롯시간을 결정하는데 고려한다면 고정된 슬롯으로 구현된 TDMA 방식에서의 대역폭 낭비시간을 줄일 수 있게 되어 대역폭을 효율적으로 이용할 수 있다.

본 논문은 TDMA 방식에 의해 실시간 특성이 부여된 이더네트에서 메시지들의 트래픽 특성을 사전분석하여 메시지 스트림의 시간 제약조건을 만족시키도록 슬롯시간 및 프레임 시간을 결정함을 목적으로 한다. 이더네트에 실시간 특성을 부여하기 위하여 TDMA를 구현하고 가변적인 대역폭 할당방식에 의해 슬롯 시간과 프레임 시간을 결정한다. MINIX1.5 운영체제에서 망 관련 태스크의 기능을 확장하여 이더네트에 TDMA를 구현하며, 이 통신 하부구조는 메시지 스트림의 트래픽 특성 분석에 의해 정해진 슬롯시간 및 프레임 시간으로 초기화된다.

본 논문의 구성은 다음과 같다. 2장에서는 이더네트에 실시간성을 부여하는 관련 연구에 대해 고찰하며 3장에서는 이더네트에 TDMA를 구현하는데 있어서 통신 시스템의 구성을 위해 필요한 태스크의 구조를 기술한다. 4장에서는 TDMA에서 주어진 메시지 스트림에 대해 프레임 시간 결정 및 슬롯시간을 할당하는 알고리즘을 제안한다. 5장에서는 다른 방식과의 비교분석에 의해 가변적인 대역폭 할당기법을 평가한다. 마지막으로 6장에서는 논문의 내용을 정리한다.

Ⅱ. 관련연구

이더네트에 실시간 통신을 가능하게 하려는 연구는 다양하게 진행되었다. 대부분은 우선순위 해결에 의해 종료시한 만족도를 향상시키고자 하였으며, 경성 실시간 시스템을 구성하려면 오스트리아 비인 대학에서 Hermann Kopetz가 제안한 실시간 시스템인 MARS에서처럼 TMDA와 같은 방식이 필요하다.

MARS는 하드웨어 시계 동기화 장치를 바탕으로 이더네트에 TDMA 프로토콜을 구현하였다(2,3). 시계 동기화 장치에 의해 제공되는 동기화된 시계 정밀도에 의해 슬롯간격이 결정되는데, 이 슬롯간격 시간에는 메시지의 전송이 일어날 수 없다. MARS에서 슬롯간격 시간은 10㎲로서, 2㎳의 슬롯시간에 비해 상당히 작다. MARS의 TDMA에서는 슬롯시간이 고정적이며, 오프 라인시 CPU의 태스크 스케쥴과 동기화되어 TDMA 통신 시스템이 스케쥴된다(4). 이후 경성 실시간 메시지들은 오프라인 스케쥴시 예약된 슬롯시간에 전송된다. MARS의 슬롯 구성은 그림 1과 같으며, 신뢰성 향상을 위해 각 노드가 연속된 두 개의 슬롯을 할당받는 것이 특징이다. 이더네트에 토큰 버스 방식을 구현하여 실시간성을 부여하는 방식도 고려될 수 있으나(5), 토큰의 유지와 토큰 손실시 복구 등의 과정이 필요하고, 토큰의 회전에 따르는 오버헤드가 크게 되어, TDMA 방식보다 대역폭의 낭비가 많다.

한편 가변적인 슬롯 크기를 결정하는 방식은 FDDI에서 동기 메시지에 대한 대역폭 할당 방식과 비교될 수 있으며 많은 연구가 진행되었다(6,7,8). FDDI에서 TTRT(Target Token Rotation Time)은 TDMA에서 프레임 시간과 같은 특성을 갖고, FDDI의 용량 벡터(capacity vector)의 결정은 각

노드에 할당된 슬롯시간의 결정과 같은 의미를 갖는다. Gopal Agrawal 등의 연구에서는(6) 각 메시지 스트림의 전송률(주기 대 전송시간)에 기초하여 고정된 TTRT 값 내에서 대역폭을 할당한다. Biao Chen 등의 연구에서는(7) 주어진 메시지 스트림의 통신량 특성과 TTRT를 기초로 하여, 용량 벡터가 메시지 스트림의 종료시한을 만족시키기 위해 필요한 제약조건을 구하며, 그 제약조건을 만족시키는 용량 벡터의 값을 결정한다. 대부분의 FDDI 대역폭 할당 방식에서는 TTRT도 시스템 인자로서 고정된 값을 갖고 용량 벡터를 결정하는데 인자로서 사용된다. Malcolm의 연구에서는(8) TTRT값도 메시지의 통신량 특성에 의해 결정하지만 지역 할당 방식에 있어서 최적의 TTRT 값을 구한다.

본 논문에서 구현하는 통신 시스템은 이더네트 상에서 시계동기화에 의해 TDMA를 구현하여 실시간 통신을 지원한다는 점은 MARS의 접근 방식과 같지만, 메시지의 트래픽 특성에 의해 슬롯시간과 프레임 시간을 결정하는 특징을 갖는다. 또 FDDI의 대역폭 할당 방식은 TTRT를 시스템 인자로서 용량 벡터 결정에 이용하였으나, 본 논문에서는 프레임 시간도 메시지 스트림의 트래픽 특성에 의해 결정한다.

Ⅲ. TDMA의 구현

이더네트에서 TDMA 방식을 구현하려면 동기화된 시계를 바탕으로 하여 각 노드는 자신에게 지정된 시간에만 메시지를 전송하도록 하여야 한다. 이때 시계동기화에 있어서 최대편차가 슬롯 간격시간을 결정하는데, 슬롯 간격시간에는 메시지 전송을 불가하게 하여 노드들의 시계 편차에 의한 메시지 충돌을 방지한다. 이미 발표되어 있는 시계동기화 알고리즘들은 각각 동기화에 따른 최대편차를 제시하는데, TDMA를 구현함에 있어서 이들이 제공하는 최대편차를 슬롯 간격시간으로 설정한다(9,10,11). 이와같이 TDMA가 구현된 이더네트에서는 한 노드가 자신에게 지정된 슬롯시간까지 기다리는 시간이 예측가능하다. 즉, 계산 가능한 한계시간 이내에 노드에게 슬롯이 도착하므로, 그 시간 이상을 기다리지는 않는다. 반면 일반 이더네트에서는 노드가 메시지를 전송하려면 다른 노드의 전송이 끝나 캐리어를 감지할 때까지 기다려야 하므로 대기시간이 예측가능하지 못하다. 또 TDMA에서 모든 노드들은 자신의 슬롯시간에만 전송을 하므로 충돌이 발생하지 않기 때문에 통신에 있어서 비실시간적 특성이 제거된다. 본 절에서는 이더네트에 TDMA를 구성함에 있어서 프레임과 슬롯의 구조 및 구현된 통신 하부구조의 구성과 기능에 대해 기술한다.

3.1 프레임과 슬롯의 구조

TDMA에서 타임 슬롯은 그림 2와 같이 구성되며, 각 노드는 자신의 슬롯시간 동안 메시지를 전송할 수 있다. 프레임은 제어 슬롯 T_c와 각 노드에게 할당된

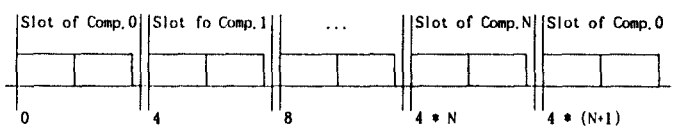

그림 1. MARS 슬롯의 구조
Fig. 1. Structure of MARS slot

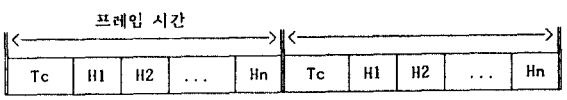

그림 2. TDMA 슬롯의 구성
Fig. 2. Structure of TDMA slot

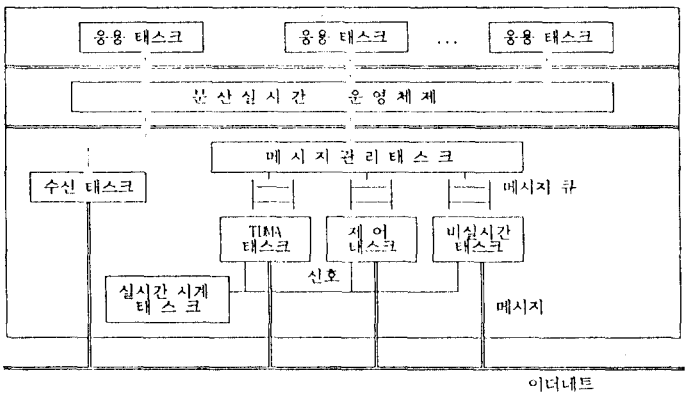

그림 3. 통신 시스템의 구성
Fig. 3. Structure of communication system

슬롯 H_i(i=1, n)로 구성된다. 제어 슬롯 T_c는 메시지 스트림의 하나로서 시스템을 유지하고 관리하는데 필요한 제어 메시지가 교환되는 슬롯이다. 이 슬롯은 주로 조정자(coordinator) 노드에 의해 사용되며, 다른 노드가 이 슬롯을 사용하고자 하는 경우에는 조정자 노드의 허가를 얻은 후 가능하다. 제어 슬롯에 대한 사용을 요청은 H_i 슬롯을 통해 이루어진다.

3.2 통신 시스템의 구성

각 노드의 통신 시스템은 메시지 관리태스크, 수신 태스크, TDMA 태스크, 실시간 시계 태스크, 제어 태스크, 비실시간 메시지 관리 태스크 등으로 구성된다. 이들의 전체적인 구조는 그림 3과 같으며, 이들은 MINIX1.5 운영체제 상에서 구현되었다.

3.2.1 메시지 관리 태스크

메시지 관리자는 분산 실시간 운영체제(Distributed Real-Time Operating System : DROS)에서 발생한 메시지들을 메시지의 특성(실시간 메시지, 비실시간 메시지, 제어 메시지 등)에 따라 각각의 큐에 넣는다. 비실시간 메시지는 전송할 실시간 메시지가 없을 경우에만 비실시간 태스크에 의해 전송된다.

3.2.2 실시간 시계(Real-Time Clock) 태스크

실시간 제어 태스크는 실시간 시계의 틱킹 신호에 의해 시간을 유지하고 있으며 제어 슬롯의 시간과 노드의 슬롯시간이 되면 해당관리자에게 슬롯시간이 되었음을 알려준다. 각 노드가 유지하는 실시간 시계는 시스템 초기화 과정에서 조정자 노드의 시계값으로 초기화되며, 이후에는 시계동기화 과정에 의해 주기적으로 재동기화된다.

3.2.3 제어 태스크

시스템의 관리 및 유지를 위한 데이타, 즉 제어 데이타가 제어 슬롯을 통해 전송되며, 제어 슬롯은 조정자 노드에 의해 관리된다. 조정자 노드의 제어 태스크는 제어 슬롯을 통해 노드들에 시계 동기화를 제공하고, 각 노드의 메시지 스트림들의 트래픽 변화를 보고받아 대역폭 할당을 재조정하여 각 노드에게 전송한다. 각 노드의 제어 태스크는 제어 슬롯을 통해 조정자 노드의 주도로 시계 동기화 메시지 및 트래픽 변화를 전송한다.

3.2.4 TDMA 태스크

TDMA 태스크는 메시지 관리 태스크로부터 전달된 메시지를 이더네트 상에서 TDMA 방식에 의해 이더네트에 전송해주는 기능을 수행한다. 실시간 시계 태스크가 슬롯 시간의 시작 신호를 발생시키면 TDMA 태스크는 큐내의 메시지를 전송한다.

3.2.5 수신 태스크

수신 태스크는 각 통신 시스템을 통해 들어오는 메시지들을 DROS를 통해 해당되는 응용 태스크에 전달한다. 수신 태스크가 DROS에 메시지를 전달하면, DROS는 이들을 큐에 넣으며, 메시지를 기다리고 있는 응용 태스크에게 전달한다.

3.3 시스템의 동작

본 논문의 통신시스템은 초기화 과정을 거쳐 TDMA 동작 모드로 동작하며, 메시지 스트림의 트래픽 특성이나 구조 변화시 재구성 과정을 갖는다.

3.3.1 초기화 과정

초기화 과정에서 조정자 노드의 주도로 초기화 과정이 수행되며, 초기화 과정은 시스템 내 모든 노드들의 시계를 일치시키고, 대역폭 할당에 의해 프레임 시간과 각 노드의 슬롯시간을 결정하여 이를 각 노드들에게 알린다. 초기화 과정에서 일반 노드들은 조정자 노드의 요청에 의해서만 데이타를 전송한다. 조정자 노드는 시스템 노드들을 차례로 폴링하여 분산 실시간 시스템 구성 노드 수와 각 노드가 전송할 메시지 스트림의 트래픽 특성을 알아낸다. 조정자 노드는 수집한 메시지 스트림의 트래픽 특성에 의해 각 노드에게 대역폭을 할당한다. 시스템의 시계들을 동기화시키기 위해 자신의 시계값을 방송하고 모든 노드들로부터 확인을 받는다. 이후 대역폭 할당 정보를 방송하며, 역시 모든 노드들로부터 확인을 받는다. 그리고 조정자 노드가 TDMA 모드의 시작을 방송하면 시스템은 TDMA 모드로 동작한다.

3.3.2 TDMA 모드

각 노드들은 시스템 초기화시 조정자 노드에 의해 결정되어진 대로 정해진 시간에 자신의 슬롯시간 만큼 메시지를 전송한다. 또한 제어 슬롯을 통해 주기적으로 시계 재동기화에 필요한 메시지가 전송된다. 그리고 노드의 메시지 스트림 트래픽 특성에 변화가 제어 슬롯을 통해 조정자에게 보고되면 조정자 노드는 대역폭 할당을 재조정하고 다른 노드들에게 이를 알린다.

Ⅳ. 대역폭 할당

일반적으로 TDMA와 같은 시간 영역 다중화 기법에서 프레임 시간을 결정할 때 (12)에서와 같이 메시지 스트림 주기들의 최대 공약수를 선택한다. 주기를 프레임 시간으로 나눌 때 나머지로 주어지는 시간은 전송에 이용할 수 없어 대역폭의 낭비가 발생하므로, 이 대역폭 낭비를 줄이기 위함이다. 그러므로 메시지 스트림의 주기들이 서로 소이거나 메시지 스트림이 많아질수록 프레임 시간을 결정하기 어렵고, 이에 대처하는 유연성이 부족하다. 반면 본 논문에서 제안하는 대역폭 할당 방식은 프레임 시간과 슬롯시간의 모든 가능한 할당을 검사하므로 이러한 유연성 부족을 해결할 수 있다.

4.1 시스템 모델과 트래픽 모델

본 논문에서는 실시간 슬롯 링(12)이나 FDDI(13)에서와 같이 동기 메시지 및 비동기 메시지의 두 메시지 계층을 고려한다. 동기 메시지는 주기적으로 도착하며 반드시 종료시한 내에 통신이 완료되어야 한다. 비동기 메시지는 비주기적으로 도착하며, 실시간 제약조건을 갖지는 않는다. 동기 메시지의 예로서는 실시간 시스템에서 감시 데이타와 같이 주기적으로 센서로부터 입력되는 메시지를 들 수 있으며, 비동기 메시지로는 사용자나 운영자가 필요에 따라 노드들에 요청하는 데이타를 들 수 있다. 이 비동기 메시지는 동기 메시지의 전송에 간섭을 일으켜서는 안된다. 결국 실시간 시스템은 주기적인 메시지에 대해 종료시한을 만족시켜야 하며, 비주기적인 메시지에 대해서는 빠른 시간내에 처리해 주어야 한다. 본 논문에서는 n개의 동기 메시지 스트림 S_1, S_2, …, S_n 이 시스템에 존재하며, 이들은 각각 하나의 노드에 의해 처리된다고 가정한다. 물론 여러 개의 메시지 스트림

이 동일한 노드에 있을 수 있으나, 하나의 메시지 스트림을 갖는 가상 노드들로 구성된 것으로 볼 수 있다. 메시지 스트림 S_i는 주기 P_i와 최대전송시간 C_i를 갖는다. S_i는 주기의 시작시간에 도착하며 주기시간 이전에 전송이 완료되어야 한다. 즉 메시지 스트림의 주기가 종료시한이 된다. 주어진 메시지 스트림의 집합 $\{S_i\}$에 대해 모든 동기화 메시지에 대해 종료시한 내 전송이 가능한 대역폭 할당이 존재할 때 메시지 스트림의 집합 $\{S_i\}$는 스케쥴 가능하다.

4.2 대역폭 할당

TDMA 방식에서 각 노드들은 일정시간 마다 자신이 전송할 수 있는 슬롯을 가지며, 그 슬롯시간 동안 자신의 메시지를 전송할 수 있다. 그런데 이더네트 위에 구현된 TDMA 방식에서는 일반적인 TDMA 방식과는 다르게 각 노드의 슬롯 시간이 고정될 필요가 없다. 즉 전송할 메시지가 많고 주기가 짧은 노드의 슬롯에 더 많은 대역폭을 할당하여 종료시한 만족도를 늘릴 수 있다. 그러므로 본 논문의 할당기법에서는 메시지 스트림의 종료시한 만족도를 높이기 위해서 시스템 초기화 과정에 전역적으로 각 노드의 메시지의 주기(종료시한), 전송시간 등을 분석하여, 각 노드에 대역폭을 할당한다. 이후 각 노드는 자신의 슬롯을 맞이하면, 시스템 초기화시 결정된 슬롯시간 동안 메시지를 전송한다. 메시지 스트림에 대한 대역폭 할당은 프레임 시간의 결정 및 결정된 프레임 시간에서 각 스트림에 대한 슬롯시간의 할당으로 귀착된다. 프레임 시간을 F, 스트림에 할당된 슬롯시간을 $H_i(i=1,\cdots,n)$이라 정의하면, 다음과 같은 관계가 성립하며 그림 4와 같다.

$$F=\Sigma H_i+\gamma \qquad (1)$$

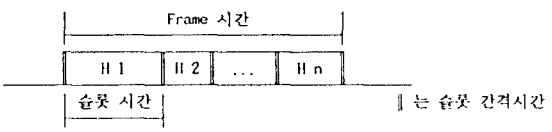

그림 4. 프레임 주기
Fig. 4. A period of frame

γ는 한 프레임 시간에서의 총 오버헤드로서 슬롯 간격시간들의 합이다. 슬롯간의 간격은 시계 동기화에 따른 편차와 패킷 생성 오버헤드 및 최대 전파 지연시간에 의해 결정된다. 즉 서로 다른 노드의 시계가 완벽하기 일치될 수는 없으므로 동시 전송에 의해 충돌이 발생할 수 있다. 이 충돌을 방지하기 위해 시계 동기화 알고리즘이 제시하는 최대 편차의 시간 동안 슬롯 사이에 간격을 두어야 한다. 또한 메시지 스트림이 여러 패킷으로 나뉘어 전송되므로 각 패킷의 생성 오버헤드도 이에 포함된다. 결국 γ는 한 프레임에서의 총 오버헤드로서 슬롯 간격시간과 노드 수의 곱으로 나타내어 진다. 본 장에서는 메시지의 트래픽 특성 및 γ와 같은 오버헤드에 의해 TDMA에서 프레임 시간 및 슬롯시간의 제약조건을 구하여 메시지 스트림의 종료시한을 보장할 수 있는 값을 결정한다. 이러한 결정 및 할당 방식을 f 라 하면 다음과 같은 관계가 성립한다.

$$f(C_1,C_2,\ldots,C_n,P_1,P_2,\ldots,P_n)\rightarrow F,H_1,H_2,\ldots,H_n \qquad (2)$$

4.2.1 프레임 시간의 결정

프레임 시간 F와 슬롯시간 H_i를 결정함에 있어서, 프레임 시간과 슬롯시간은 서로 밀접한 관계를 갖게 된다. 즉 프레임 시간이 슬롯시간 결정에 있어서 영향을 미치게 되며, 반대로 슬롯시간이 프레임 시간 결정에 영향을 주게 된다. 슬롯시간 H_i는 메시지 스트림의 주기 $P_i(1\le i\le n)$와 메시지 스트림의 전송시간 $Ci(1\le i\le n)$을 고려해 결정되어야 한다.

전송가용시간

프레임 시간과 메시지 스트림의 슬롯시간 $\{H_i\}$가 결정되었다면, 메시지 스트림 S_i가 주기내에 메시지를 전송할 수 있는 시간 X_i는 (3)과 같다.

$$X_i = \left(\left\lfloor \frac{P_i}{F} \right\rfloor - 1 \right) \cdot H_i \qquad (3)$$

S_i는 한 주기내에 최소한 $\left\lfloor \frac{P}{F} \right\rfloor$ 번의 슬롯을 맞이하게 된다. 그런데, 자신의 슬롯을 맞이할 때 이미 전송할 패킷이 준비되어 있어야만 슬롯시간을 이용할 수 있다. 그림 5에서와 같이 슬롯시간 중에 전송할 패킷이 도착하는 경우는 한 슬롯을 사용할 수 없게 된다. 그러므로 한 메시지 스트림이 자신의 주기내에서 전송가능한 슬롯을 맞이하는 횟수는 $\left\lfloor \frac{P}{F} \right\rfloor -1$ 이며, 따라서

한 주기내에 전송가능한 시간은 $\left(\left[\frac{P_i}{F}\right]-1\right)\cdot H_i$가 된다.

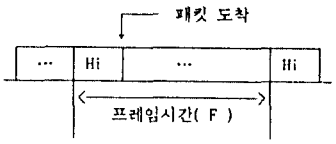

그림 5. 전송가용시간
Fig. 5. Available transmission time

프레임 시간의 범위

스케줄 가능한 대역폭 할당이 존재하려면 프레임 시간은 다음의 범위에 있어야 한다.

$$\frac{\gamma}{1-U} \le F \le \frac{P_{min}}{2}, \quad 단\ U = \sum_{i=1}^{n} \frac{C_i}{P_i} \qquad (4)$$

한 프레임 시간 F는 (1)에서와 같이 슬롯시간과 오버헤드로 구성된다. 한 프레임 시간에서 실제 전송할 수 있는 시간은 (F-γ)가 된다. 그리고 한 프레임 시간내에서 각 스트림에 의해 필요한 전송시간은 F · U 이다. 여기서 U는 사용율로서 각 스트림이 필요로 하는 전송 비율의 합이며, 사용율 U가 1을 넘으면 종료시한 내 전송을 보장할 수 없다. 그리고 주어진 메시지 스트림의 집합에 대해 종료시한내 전송을 보장하려면 실제 전송할 수 있는 시간이 필요한 전송시간보다 많아야 한다.

$$F - \gamma \ge F \cdot U$$

$$F \ge \frac{\gamma}{1-U}$$

또한 프레임 시간은 모든 메시지 스트림의 주기보다는 작아야 한다. 프레임 시간이 어떤 메시지 스트림의 주기보다 길다면, 그 스트림은 한 주기내에 자신의 슬롯 시간을 받을 수 없는 경우가 생기므로 종료시한을 만족시킬 수 없다. 또한 (3)에서 계산된 전송가용시간은 0 보다 커야 하므로

$$\left[\frac{P_{min}}{F}\right] \ge 2 \text{ 가 되고,}$$

$$F \le \frac{P_{min}}{F}$$

와 같이 되어 (4)가 성립한다.

각 노드의 전송 시간을 결정함에 있어서 한 프레임 주기가 길어진다면, 메시지의 전송시간이 늘어나게 되어 메시지의 종료시한을 만족시키지 못하며, 프레임 주기가 짧다면, 전송시 마다 오버헤드가 부가되게 된다. 일반적으로 이러한 프레임의 주기를 결정함에 있어서, 각 메시지 주기의 최대공약수로 프레임 시간을 설정하는 것이 효율적이다. 그러나 메시지 주기들이 서로 소이거나, 최대공약수가 상당히 작은 경우에 있어서는 프레임의 주기를 설정할 수 없다. 프레임 시간이 메시지 스트림 주기의 약수로 설정될 때, 그 메시지 스트림의 전송가용 시간이 많아진다.

4.2.2 슬롯시간의 할당

한 프레임 시간을 결정하면 이 프레임 시간 내에서 각 메시지 스트림에 대해 슬롯시간을 할당한다.

프로토콜 제약조건

각 노드들에게 할당된 슬롯시간의 합은 프레임 시간 중 가용부분(latency를 제외한 시간)보다는 작아야 한다. 이 제약조건은 FDDI의 할당 방식의 제약조건과 같다(6).

$$즉, \quad \sum_{i=1}^{n} H_i \le F - \gamma \qquad (5)$$

를 만족시켜야 한다.

사용율 제약조건

각 노드의 메시지 스트림의 사용율, 프레임 시간 설정에 따른 오버헤드, 슬롯 간격 시간의 오버헤드의 합은 1.0을 넘어서는 안된다. 즉,

$$O_f + U + \frac{\gamma}{F} \le 1.0 \qquad (6)$$

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

O_f는 프레임 시간 설정에 따른 오버헤드로서 프레임 시간이 메시지 스트림의 주기의 약수가 아니어서 발생되는 오버헤드이다. 이 오버헤드는 주기를 프레임 시간으로 나누어 나머지로 주어지는 시간은 전송에 이용할 수 없기 때문에 발생된다. 이 오버헤드는

각 스트림의 주기당 낭비시간과 프레임시간과의 비율을 산정하여 구해진다. U는 각 메시지 스트림의 주기 대 전송시간 비로서 순수한 메시지 전송을 위해 필요한 양이다. $\frac{\gamma}{F}$는 프레임 시간 마다 소요되는 오버헤드로서 잠재시간과 프레임시간의 비이다. 이들의 합이 1.0을 넘게 되면, 종료시한을 만족시킬 수 있는 슬롯시간 할당이 존재하지 않는다.

종료시한 제약조건

동기 메시지에 대해 종료시한 이전에 전송을 할 수 있도록 슬롯시간이 할당되어야 한다. 슬롯시간 할당의 결과 모든 메시지 스트림에 대해 주기 Pi내에 노드가 전송할 수 있는 시간 $X_i(1\leq i\leq n)$가 메시지 전송시간 Ci 보다 커야 한다.

즉,

$$X_i \geq C_i \text{ (for all i)} \qquad (7)$$

를 만족시켜야 한다.

슬롯시간의 결정

또한 (3)을 (7)에 대입하면 H_i를 구할 수 있다. 즉, 종료시한을 만족시키기 위한 H_i는 (3)과 (7)에 의해 다음과 같이 계산된다.

$$H_i \geq \frac{C_i}{\left(\left\lfloor \frac{P_i}{F} \right\rfloor - 1\right)} \qquad (8)$$

그림 6은 프레임 시간 결정 및 슬롯시간을 할당하는 알고리즘이다. 각 메시지 스트림의 주기는 단위시간의 정수배라고 가정한다. 단위시간은 시스템의 메시지 스트림의 특성에 따라 결정된다. (4)에서 보여진 프레임 시간의 범위에 대해 각각 슬롯시간을 (8)과 같이 결정하여 각종 제약조건을 만족시키는지 검사한다. 주기들이 서로 소라면 최대 공약수는 1이 되며 범위 내 모든 프레임 시간에 대해 스케쥴 가능성을 검사한다. 그러나 오프라인시에 계산되므로 최대 공약수가 작다고 하더라도 충분한 시간내에 프레임 시간 및 슬롯시간을 계산할 수 있다. 각 노드의 통신 관리자는 한 프레임 내의 시간표(time table) 만을 유지하면 되므로 통신을 위해 관리하는 데이타의 양은 노드의 수에 비례한다.

Table 1은 위의 알고리즘을 사용하여 종료시한을 만족시키는 프레임 시간 및 슬롯시간을 결정한 예이다. 시간의 단위는 100μ을 기준으로 하였다.

표 1. 메시지 스트림 집합 (슬롯간격 시간=4)
Table 1. Example set of message stream(interslot time=4)

	τ1	τ2	τ3	τ4	τ5
주 기	555.00	1577.00	1866.00	701.00	705.00
전송시간	100.00	150.00	200.00	150.00	100.00

위의 메시지 스트림 집합에 대해 사용율은 0.738303 이며 (3)에 의해 구해진 프레임 시간의 범위는 Table 2와 같다.

표 2. 프레임 시간의 범위 (=20)
Table 2. The range of frame time(=20)

최소	최대
38.212078	287.500000

알고리즘에 의해 결정된 프레임 시간 및 계산된 슬롯시간은 Table 3과 같다. 프레임 시간 50.0 에서 스케쥴 가능한 슬롯시간 할당이 존재한다.

Table 3. 슬롯시간의 할당 (프레임 시간 50.0)
Table 3. Allocation of each slot time(frame time 50.0)

	τ1	τ2	τ3	τ4	τ5
슬롯시간	10.00	5.00	5.56	11.54	7.69
전송시간	100.00	150.00	200.00	150.00	100.00

V. 실험 평가 및 토의

본 논문에서 제안하는 가변적인 할당방식을 평가하기 위해 140개의 태스크 집합을 생성하고 각 태스크 집합에 대해 스케쥴 가능성을 분석하였다(14). 또한 고정적인 슬롯 할당 방식과 통계 TDMA(Stastical TDMA)의 스케쥴 가능성을 분석하여 비교하였다. 생성된 태스크 집합의 특성은 다음과 같다. 30% 대부터 90% 대까지의 사용율을 갖는 태스크 집합이 각 20개씩이며, 태스크의 갯수는 2~10개, 주기는 100~1000, 전송시간은 1~200 까지 임의로 분포한다. 이 태스크 집합에 대해 각각의 기법으로 대역폭을

할당하고 SMPL(15)을 이용한 시뮬레이션에 의해 종료시한 만족여부를 측정하였다. 고정슬롯 방식과 통계 TDMA 방식의 슬롯시간은 MARS에서와 같이 2ms로 설정하였다. 그림 7과 8은 사용율에 따른 종료시한 만족도를 나타낸다. 슬롯간격 시간은 시계동기화 편차에 의해 영향을 받는데, 시계동기화 편차는 동기화에 참여하는 노드의 수, 하드웨어 시계동기화 장치의 유무, 운영체제의 오버헤드, 시계동기화의 주기 등의 요소에 의해 결정된다. 그러므로 슬롯간격 시간을 변화시켜가며 종료시한 만족여부를 측정하였다. 가변적인 할당방식에 의해 슬롯시간을 결정하면 다른 방식보다 많은 태스크 집합에 대해 종료시한내 전송을 보장할 수 있다. MARS에서와 같은 10μs의 슬롯간격 시간을 설정한다면 그림 7에서 보이는 바와 같

```
procedure frame_slot()
input  : {Ci} : the transmission times of message set {Si}
         {Pi} : the periods of message set {Si}

Fmin ← γ / (1 - U)

Fmax ← Pmin / 2

compute GCM of {Gi}

     frame ← ⌈Fmin / GCM⌉ · GCM    /* first multiplier of GCM which is  */
                                   /* larger than Fmin                  */
     while ( frame ≤ Fmax) {
       if (npalloc( Ci, Pi) == success) exit
       else frame ← frame + GCM
     }
     return(no schedule)
end mainloop

procedure npalloc()
input : {Ci}, {Pi}

     if (Of + U + γ/F ) ≤ 1.0      return(fail)

     for all i

     Hi ← Ci / (⌊Pi/F⌋ - 1)

     if ( ∑ Hi ≤ F - γ )  return(success)
     else  return(fail)
```

그림 6. 프레임 시간 결정 및 슬롯시간 할당 알고리즘
Fig. 6.The algorithm of deciding frame time and allocating slot time

이 사용율 90%대의 태스크 집합에 대해서도 스케쥴할 수 있다. 그림 8은 슬롯간격 시간이 200㎲에서 측정한 결과로서, 슬롯간격시간이 늘어난다면 종료시한을 만족시킬 수 있는 태스크 집합의 수가 줄어들지만 다른 방식에 비해 우수함을 나타낸다. 그림 9와 그림 10은 사용율 0.5%대와 0.7%대의 태스크 집합들에 대해 각 슬롯간격 시간 별로 종료시한 만족도를 측정한 결과이다. 역시 가변적인 슬롯할당 방식이 우수한 결과를 보여준다. 물론 고정슬롯 방식도 슬롯의 크기에 따라 스케쥴가능성이 향상될 수 있으나 본 논문에서는 MARS의 슬롯시간을 기준으로 하였다.

본 논문에서는 시스템 동작 중에 메시지 스트림의 변화가 없다고 가정하여 T_c를 0으로 설정하여 성능평가를 하였다. 그러나 메시지 스트림의 특성이 변화하고 새로운 스트림의 추가 및 삭제를 고려한다면 그림 2에서 보이는 T_c 슬롯을 사용하여 조정자 노드와 각 노드가 수시로 정보를 교환하여야 한다. 이 과정에서도 스트림 추가 및 삭제의 빈도수와 같은 시스템의 특성을 고려하여 T_c 슬롯의 크기를 결정할 수 있다. 스트림의 변화가 많은 환경에서 T_c는 시스템의 오버헤드로서 성능저하 요인이 된다. 이 경우 알고리즘의 적용의 실시간성 문제는 조정자와 각 노드간 데이타를 주고 받고 이에 의해 조정자 노드가 슬롯시간을 재결정하여 다시 각 노드에 알리는 과정의 수행시간에 따라 결정되는데 알고리즘이 계산할 수 있는 제한된 시간 이내에 수행되므로 예측가능성을 보장할 수 있다.

동기화된 시계를 기반으로 하여 TDMA를 구현함에 있어서 시스템내 한 노드의 시계고장과 같은 통신 시스템의 결함 발생에 의해 메시지 충돌이 발생할 수 있으며, 이 경우에는 모든 노드들에 대해 실시간 통신을 보장할 수 없다. MARS에서는 자기검사(self-checking) 장치에 의해 고장 발견시 그 노드의 전송을 불가능하게 함으로써 충돌을 방지한다. 자기검사 장치를 사용할 수 없다면 충돌 발생후 재구성 과정에 의해 고장 노드를 배제하여 프레임 시간과 슬롯시간을 재결정하는 방식이 필요하다. 통신 시스템의 결함에 기인한 재구성 과정에서는 메시지 스트림의 전송이 불가능해지므로 여분의 통신 시스템을 이용하여 메시지 스트림의 전송을 보장하는 결함허용 통신 시스템을 구성하는 것이 바람직하다. 또한 본 논문에서는 각 노드들의 시계가 충분한 해상도를 제공할 수 있는 것으로 가정하고 슬롯시간을 할당하였는데, 시계 해상도가 낮다면 슬롯간격시간이 더 커질 수 있으므로 프레임 시간내의 오버헤드도 증가한다. 그러나 대부분의 시계가 충분한 해상도를 제공하므로 해상도에 의해 증가하는 오버헤드는 시스템 구성에서 큰 영향을 미치지 않는다.

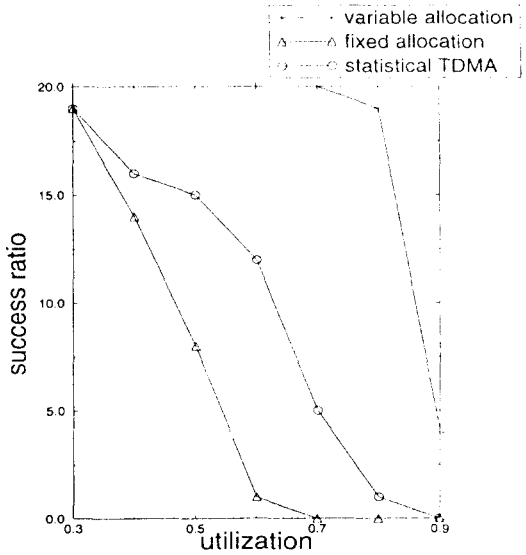

그림 7. 사용율과 만족도(슬롯간격=10㎲)
Fig. 7. Schedulability versus Utilization(Interslot time=10㎲)

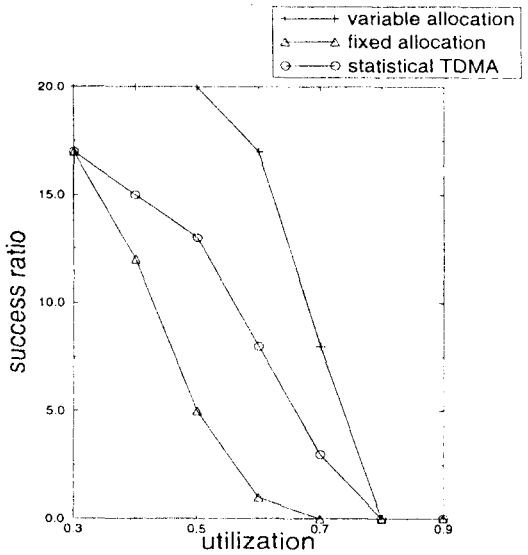

그림 8. 사용율과 만족도(슬롯간격=200㎲)
Fig. 8. Schedulability versus Utilization(Interslot time=200㎲)

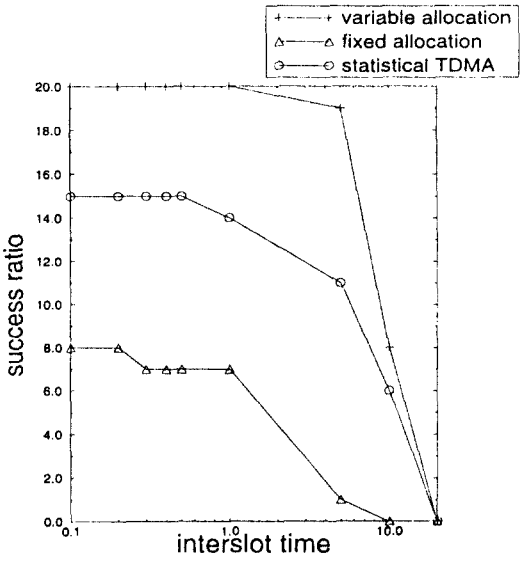

그림 9. 슬롯간격과 만족도(사용율=0.5)
Fig. 9. Schedulability versus Interslot time(Utilization=0.5)

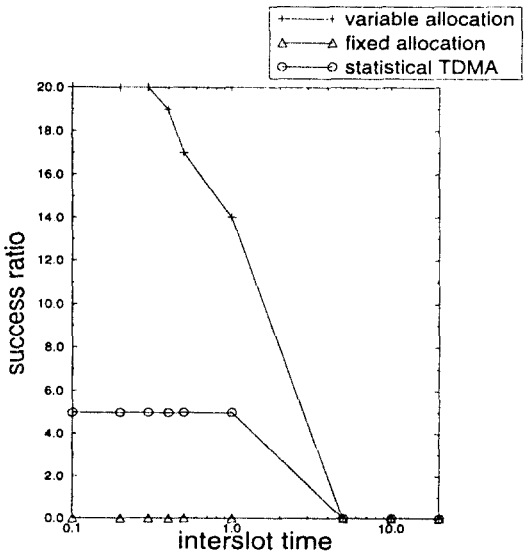

그림 10. 슬롯간격과 만족도(사용율=0.7)
Fig. 10. Schedulability versus Interslot time(Utilization=0.7)

Ⅵ. 결 론

본 논문에서는 분산 실시간 시스템 구성에 있어서 기본적인 구성요소인 통신 시스템으로 실시간 특성이 부여된 이더네트를 사용하였다. 이더네트 위에 TDMA 방식 구현에 의해 충돌과 캐리어를 감지하기 위한 대기시간을 제거하여 예측가능한 통신시스템을 구축하였다. 또한 주어진 메시지 스트림의 집합을 오프라인시에 분석하여 TDMA의 프레임 시간을 결정하고 슬롯시간을 할당하는 알고리즘을 제시하였다. 메시지 집합의 종료시한을 만족시키기 위한 제약조건을 구하였으며, 이 제약조건을 만족시키도록 각 노드의 슬롯시간을 가변적으로 할당하였다. 그리고 성능을 평가하기 위해 임의의 태스크 집합들을 생성하고 이들 집합에 대해 스케쥴가능성을 검사하였으며 다른 대역폭 할당방식과 비교하였다. 가변적인 대역폭 할당방식은 프레임 시간과 슬롯시간을 동시에 고려하여 대역폭을 할당하므로, 메시지들의 주기가 서로 소이거나 최대공약수가 작은 상황에서도 가능한 스케쥴을 찾을 수 있다.

참고문헌

1. Carrier Sense Multiple Access with Collision Detection(CSMA/CD) : Access Method and Physical Layer Specifications, ANSI/IEEE Standard 802.3-1985, IEEE, 1985.
2. Hermann Kopetz and W. Merker, "The architecture of MARS," Proc. of 15-th Fault-tolerant Computing Symposium, pp.274-279, June 1985.
3. Hermann Kopetz and et al, "Distributed fault-tolerant real-time systems : the MARS approach," IEEE Micro, pp.25-40, Febrary 1989.
4. A. Damm, J. Reisinger, W. Schwabl, and Hermann Kopetz,"The real-time operating systems of MARS," Operating Systems Review, pp.141-157, 1988.
5. Joseph K. Y. Ng and Jane W. S.Liu, "Performance of local network protocols for hard real-time application," Proc. IEEE International Conference on Distributed Computer Systems, pp. 318-326, 1991.
6. Gopal Agrawal, Biao Chen, and Wei Zhao, "Guaranteeing synchronous message dead-

lines with the timed token protocol," Proc. IEEE International Conference on Distributed Computer Systems, pp. 468-475, June 1992.
7. Biao Chen, Gopal Agrawal, and Wei Zhao, "Optimal synchronous capacity allocation for hard real-time communications with the timed token protocol," Proc. Real-Time Systems Symposium, pp.198-207, December 1992.
8. N. Malcolm and W. Zhao, "Guaranteeing synchronous messages with arbitary deadline constraints in an FDDI network," Proc. the Conf. on Local Area Networks, IEEE CS Press, pp. 186-195, 1993.
9. Hermann Kopetz and Wilhelm Ochsenreiter, "Clock synchronization in distributed real-time systems," IEEE Trans. Computers, Vol. C-36, No. 8, pp.933-940, August 1987.
10. Riccardo Guesella and Stefano Zatti, "The accuracy of the clock synchronization achieved by TEMPO in Berkeley UNIX 4.3BSD," IEEE Trans. Software Engineering, Vol. 15, No. 7, pp.847-853, July 1989.
11. P. Verssimo, L. Rodrigues, and A. Casimiro, "Using atomic broadcast to implement a posteriori agreement for clock synchronization," 12-th Symposium on Reliable Distributed Systems, pp.115-124, 1993.
12. S. Mukherjee, D. Saha, M. C. Saksena, and S. K. Tripathi, "A bandwidth allocation scheme for time constrained message transmission on a slotted ring LAN," Proceedings of Real-Time Systems Symposium, pp.44-53, December 1993.
13. Sonu Mirchandani and Raman Khanna, FDDI : Technology and Applications, John Wiley & Suns, Inc., 1993.
14. Krithi Ramamritham, "Allocation and scheduling of complex periodic tasks," Proc. IEEE International Conference on Distributed Computer Systems, pp.108-115, 1990.
15. M. H. MacDougall, Simulating Computer Systems : Techniques and Tools, MIT Press, 1987.

李政勳 (Jung hoon Lee)

1988 : 서울대학교 컴퓨터공학과 공학사
1990 : 서울대학교 컴퓨터공학과 석사
1990~1992 : 대우통신 종합연구소 통신기술 연구실 근무
1992~현재 : 서울대학교 컴퓨터공학과 박사과정

申玹植 (Heon shik Shin)

1973 : 서울대학교 응용물리학과 공학사
1980 : 미국텍사스대학교 의공학 석사
1985 : 미국텍사스대학교 컴퓨터공학 박사
1986~1990 : 서울대학교 컴퓨터공학과 조교수
1990~현재 : 서울대학교 부교수