

대역폭 할당을 통한 Foundation Fieldbus 통신망 설계

Network Design of Foundation Fieldbus using a Bandwidth Allocation Scheme

홍 승 호, 최 인 호, 문 봉 채
(Seung-Ho Hong, In-Ho Choi, and Bong-Chae Moon)

Abstract : The data link layer of Foundation Fieldbus is operated on the basis of scheduling and token-passing disciplines. This paper presents a network design method of Foundation Fieldbus using a bandwidth allocation scheme. Fieldbus traffic consists of periodic data and sporadically generated time-critical and time-available data. The bandwidth allocation scheme schedules the transmission of periodic data. Time-critical and time-available data are transmitted via a token-passing service. The validity of this bandwidth allocation scheme is determined using an experimental model of a network system. The results obtained from the experimental model show that the proposed scheme restricts the delay of both periodic and time-critical data to a pre-specified bound. The proposed bandwidth allocation scheme also fully utilizes the bandwidth resource of the network system.

Keywords : Foundation Fieldbus, bandwidth allocation, implementation, experimental model, network design

I. 서론

필드버스는 자동화 및 분산제어 시스템의 필드에 설치된 각종 센서, 제어기, PLC, 구동기 등의 필드 장비에서 생성되는 데이터를 전송하는데 사용되는 양방향 디지털 직렬 통신망이다[1]. 필드버스에 접속되는 자동화 및 제어 장비에서 생성되는 데이터는 크게 산발적으로 발생하는 실시간 데이터와 주기적 데이터 및 비실시간 데이터의 세 종류로 구분할 수 있다[2]. 산발적 실시간 데이터는 자동화 시스템에서 발생하는 각종 사건이나 경고 신호를 전달하기 위하여 사용되며, 일반적으로 데이터 길이가 짧고 가장 높은 우선순위로 전송된다. 주기적 데이터는 주로 피드백 제어시스템에서 센서 및 제어기 신호를 전달하기 위해서 사용되며, 반드시 데이터 생성 주기의 제한 시간 이내에 데이터 전송이 완료되어야 한다. 비실시간 데이터에는 프로그램 및 데이터 파일 등이 포함되며, 전송 지연 시간에는 크게 제약을 받지 않는다. 필드버스에서는 이런 다양한 특성을 갖는 데이터들이 하나의 미디어를 공유하게 된다. 따라서 필드버스의 대역폭을 적절하게 관리하지 않으면, 데이터의 전송 지연 시간이 한계값을 초과하게 되어 필드버스에 접속된 응용 시스템의 성능 요구사항을 만족시키지 못하는 경우가 발생할 수 있다.

Hong[3]은 네트워크 내의 노드들이 미리 정해진 순서에 따라 차례로 데이터를 전송하는 순환 서비스 방식으로 동작되는 필드버스에서 산발적 실시간 및 주기적 데이터의 전송 지연시간 요구사항을 만족하는 동시에 필드버스의 대역폭을 충분히 활용할 수 있는 대역폭 할당 기법의 기본 개념을 제시하였다. 대역폭 할당 기법의 기본 개념은 필드버스의 대역폭을 실시간으로 전송되어야 하는 산발적 및 주기적 데

이터에 우선적으로 할당하고, 남은 대역폭을 충분히 활용하여 비실시간 데이터를 전송함으로써 필드버스 대역폭의 활용도를 극대화 시키는 것이다. [3]에서 제시한 대역폭 할당 기법을 토큰-패싱의 순환 서비스 방식으로 동작되는 Profibus에 적용하기 위하여서는 (1) 주기적 및 비실시간 대역폭을 구분하는 기능과 (2) 노드들 간에 시간을 동기화하는 기능이 추가되어야 한다. [4]에서는 이러한 기능들의 구현 방법을 개발하여 이를 토큰-패싱 방식으로 동작되는 Profibus에 적용하는 방법을 제시하였다. 대역폭 할당 기법의 기본 원리는 반드시 순환서비스 방식으로 동작되는 필드버스에만 적용할 수 있는 것이 아니다. [5]에서는 대역폭 할당 기법을 순환서비스 방식이 아닌 CSMA/NBA 방식으로 동작되는 CAN 프로토콜에 적용하는 방법을 제시하였다. Foundation Fieldbus는 토큰-패싱 방식과 스케줄링 방식을 모두 지원한다. 본 논문에서는 [3]에서 제시한 대역폭 할당 기법의 기본 개념을 토큰-패싱뿐만 아니라 스케줄링 방식도 지원하는 Foundation Fieldbus에 적용하는 방법을 제시한다.

Foundation Fieldbus[6]는 현존하는 여러 종류의 필드버스를 가운데 가장 최근에 제품화가 완료된 프로토콜로서 기존의 필드버스 프로토콜들과는 달리 토큰-패싱과 스케줄링의 두 가지 데이터 전송 방식을 모두 지원한다. 본 논문에서는 Foundation Fieldbus에서 대역폭 할당 기법을 통하여 통신망을 설계하는 방법을 제시하고, 이를 실험적으로 검증한다. 본 논문의 II장에서는 Foundation Fieldbus의 데이터링크 계층 프로토콜 구조에 대하여 간략히 기술한다. III장에서는 Foundation Fieldbus의 대역폭 할당 기법에 대해서 기술하고, IV장에서는 본 연구를 통하여 구현된 Foundation Fieldbus 데이터링크 계층 프로토콜의 하드웨어와 소프트웨어에 대하여 기술한다. V장에서는 대역폭 할당기법을 실험모델에 적용한 실험 결과에 대해서 고찰한다. 마지막으로 VI장에서는 본 논문의 결론 및 추후 연구 과제에 대하여 기술하도록 한다.

접수일자 : 2001. 2. 12., 수정완료 : 2001. 6. 13.

홍승호 : 한양대학교 전자컴퓨터공학부(shhong@hanyang.ac.kr)

최인호, 문봉채 : 삼성전자㈜ 지능시스템 연구소(ihchoi@samsung.co.kr/bcmoon99@samsung.co.kr)

* 이 논문은 1999년도 한국학술진흥재단의 연구비에 의하여 지원되었음(KRF-99-041-E00301).

II. Foundation Fieldbus 데이터링크 계층 프로토콜

Foundation Fieldbus의 데이터링크 계층은 IEC 국제 표준 가운데 'type 1' 필드버스 데이터링크 계층[7]의 부분 집합으로 LAS(Link Active Scheduler)라는 버스 스케줄링을 담당하는 노드가 각 노드에서의 메시지 전송 권한을 부여한다. 기본적으로는 중앙 제어 방식으로 동작되며 토큰-패싱 방식과 스케줄링 방식에 의한 데이터 전송을 모두 지원한다. 로컬 링크에는 최대 32개의 노드의 접속이 가능하다. 데이터링크 계층의 모든 메시지는 DLPDU(Data Link Protocol Data Unit)의 형태로 전송되는데, Foundation Fieldbus에서 사용되는 DLPDU는 용도에 따라 EC(Establishment Connection), DC(Disconnect Connection), CD(Compel Data), DT(Data), SR(Status Response), CT(Compel Time), TD(Time Distribution), RQ(Round-Tip-Delay Query), RR(Round-Tip-Delay Reply), PT(Pass Token), RT(Return Token), RI(Request Interval), CL(Claim LAS), TL(Transfer LAS), PN(Probe Node), PR(Probe Response), IDLE의 17가지가 존재한다.

Foundation Fieldbus의 데이터링크 계층에서는 기능적인 관점에서 따라 기본 기기(Basic Device), 링크 마스터(Link Master), 브리지(Bridge)의 세 가지의 기기를 정의한다. 링크 마스터는 LAS가 될 수 있는 기기를 말한다. 즉, 현재의 LAS에 문제가 발생했을 경우 이를 대체할 수 있는 기기이다. 기본 기기는 LAS가 될 수 없는 일반적인 노드가 되는 기기를 말하며, 브리지는 필드버스의 로컬 링크를 연결할 때 사용하는 기기를 말한다. Foundation Fieldbus의 데이터링크 계층에서는 데이터의 우선순위를 URGENT, NORMAL, TIME-AVAILABLE의 세 가지로 구분하며, 이들의 데이터 필드 길이는 각각 64, 128, 256 옥텟으로 제한된다.

LAS는 주기적 데이터의 전송을 위하여 스케줄링에 의한 데이터 전송을 지원한다. LAS는 주기적으로 전송해야 하는 데이터들에 관한 스케줄 목록을 가지고 있으며, 스케줄 목록에는 주기적 데이터를 생성하는 노드들에 대한 스케줄 서비스 시작 시간과 스케줄 서비스 주기가 기록된다. LAS는 스케줄 서비스 시간이 되면 해당 노드에 CD DLPDU를 전송하여 전송 권한을 부여한다. CD DLPDU를 수신한 노드는 DT DLPDU를 통하여 하나의 데이터를 전송하며, 이때 DT DLPDU는 필드버스 상의 모든 노드에 Publisher/Subscriber 방식으로 전송된다. 데이터를 전송한 후에 해당 노드는 RT DLPDU를 이용해서 LAS로 토큰을 반납한다.

LAS는 또한 비주기적 데이터의 전송을 위하여 토큰-패싱에 의한 데이터 전송을 지원한다. 토큰-패싱에 의한 데이터 전송 서비스는 스케줄링 서비스와 다음의 스케줄링 서비스의 사이에서 수행된다. LAS는 V(TCL, token-circulation-list)에 등록된 노드 주소의 순서에 따라서 PT DLPDU를 통하여 토큰을 전송함으로써 해당 노드에 데이터 전송 권한을 부여한다. PT DLPDU에 의한 데이터 전송은 우선순위 기능을 지원하며, 우선순위가 높은 데이터 전송큐에 보다 많은 데이터 전송 기회를 부여한다. PT DLPDU를 수신한 노드는 PT DLPDU의 우선순위보다 높거나 같은 우선순위를 갖는 데이터만을 전송할 수 있다. 토큰을 수신한 노드는 PT DLPDU에 포함된 매개변수인 PTDD(PT Delegation-Duration)에서

지정하는 시간동안 DT DLPDU를 통하여 데이터를 전송한다. PT DLPDU의 우선순위는 TTRT(Target Token Rotation Time)라는 네트워크 파라미터와 토큰이 모든 노드를 방문하는데 소요되는 시간인 ATRT(Actual Token Rotation Time)를 비교해서 결정한다. LAS는 토큰이 V(TCL)에 등록된 모든 노드를 한번 순환할 때마다 ATRT를 측정하며, TTRT > ATRT이면 토큰의 우선순위를 감소해서 낮은 우선순위의 데이터도 전송할 수 있도록 하고 TTRT ≤ ATRT이면, 토큰의 우선순위를 증가시켜서 높은 우선순위의 데이터만 전송 가능하도록 한다. PT DLPDU를 수신한 노드는 PTDD의 시간이 종료되었거나, 더 이상 전송할 데이터가 없는 경우는 RT DLPDU를 이용해서 토큰을 LAS로 반납한다.

LAS는 다음 스케줄 서비스를 위하여 CD DLPDU를 전송하기 전에 여분의 시간이 존재하는 경우 로컬 링크의 관리 기능을 수행한다. 이러한 네트워크 관리 기능에는 링크의 시간 동기화 기능과 활성 노드의 주소록(Live List) 유지 등을 위한 DLPDU 교환 등이 포함된다. 본 논문에서는 Foundation Fieldbus 데이터링크 계층의 데이터 전송 기능에 대하여 간략히 기술하였으며, 기타 다른 DLPDU들의 동작에 대한 자세한 사항들은 규격서[6]에 명시되어 있으므로 본 논문에서는 생략한다.

III. Foundation Fieldbus의 대역폭 할당 기법

본 장에서는 Foundation Fieldbus의 데이터 링크 계층에서 토큰-패싱과 스케줄링 방식을 모두 사용하여 데이터를 전송하는 경우에 제한된 대역폭을 공유하는 산발적 실시간, 주기적 및 비실시간 데이터가 주어진 성능 요구사항을 만족하도록 하는 대역폭 할당 기법에 의한 통신망 설계 기법에 대하여 기술한다. Foundation Fieldbus에서 주기적으로 발생하는 데이터는 스케줄링 방식으로 전송되는 반면에, 산발적으로 발생하는 실시간 및 비실시간 데이터는 토큰-패싱 방식으로 전송된다. 토큰-패싱 방식으로 동작되는 경우 URGENT와 NORMAL 우선순위 데이터는 실시간 데이터로 분류되며, TIME-AVAILABLE 우선순위 데이터는 비실시간 데이터로 분류된다. 본 논문에서는 토큰-패싱 방식에서 URGENT, NORMAL 및 TIME-AVAILABLE 데이터의 우선순위를 각각 0, 1 및 2로 표기하며, 주기적으로 발생하는 스케줄 데이터는 s 로 표기한다. 다음에는 본 논문에서 사용되는 주요 기호들이 기술되어 있다.

Notation :

N_s : 스케줄 데이터 전송큐의 개수

N_r : 토큰-패싱 데이터 전송을 위하여 V(TCL)에 등록된 노드의 개수

$[N_j, j = 0, 1, 2]$: 우선순위 j 데이터 전송큐의 개수

L_s : 스케줄 데이터 패킷의 전송시간

$[L_j, j = 0, 1, 2]$: 우선순위 j 데이터 패킷의 전송 시간

σ : LAS가 한 노드에서 토큰-패싱 또는 스케줄링 서비스를 처리하는데 소요되는 오버헤드

R : 토큰이 데이터의 전송 없이 모든 노드를 한번 순환하는데 걸리는 오버헤드 시간

ϕ_c : 산발적 실시간 데이터의 최대 허용 지연시간

- [$\phi_i, i = 1$ 에서 N_s] : 전송큐 i 에서 스케줄 데이터의 최대 허용 지연시간
- [$T_i, i = 1$ 에서 N_s] : 전송큐 i 에서 스케줄 데이터의 생성 주기
- [$\lambda_j^i, i = 1$ 에서 N_j] : 전송큐 i 에서 우선순위 j 데이터의 평균 도착 빈도

1. 스케줄 데이터의 대역폭 할당

Foundation Fieldbus의 데이터링크 계층에서는 주기적으로 전송되어야 하는 데이터의 전송을 지원하기 위하여 스케줄링에 의한 데이터 전송 방식을 제공한다. 그러나 프로토콜 규격서[6]에는 스케줄 데이터의 전송을 위한 스케줄의 시작 시간(starting time)과 주기(period of cyclic sequence)가 정해진 경우에 LAS가 CD DLPDU를 통하여 이들을 전송하는 방식에 대하여서만 명시되어 있을 뿐이며, 주기적으로 생성되는 데이터의 성능 요구사항을 만족시키기 위하여 LAS에서 각 노드에 대한 스케줄의 시작시간과 주기를 어떻게 설정하여야 하는가에 대하여서는 기술되어 있지 않다. 즉, 스케줄의 시작 시간과 주기의 결정은 필드버스 통신망의 설계자가 알아서 설정하여야 할 통신망 파라미터로 주어져 있다. 본 절에서는 스케줄 데이터의 최대 허용 지연시간이 주어졌던 경우에 이에 대한 성능 요구사항을 만족시키도록 하는 스케줄링 기법을 대역폭 할당 기법을 통하여 제시한다.

본 논문에서 제시하는 대역폭 할당 기법에서는 Foundation Fieldbus의 대역폭을 최소 주기적 데이터 생성 주기인 T_1 의 시분할 구간으로 나누고, 이를 다시 주기적 구간들과 비주기적 구간들로 나눈다. 스케줄 구간에서는 주기적으로 생성되는 데이터만이 스케줄링 방식에 의하여 전송되며, 비주기적 구간에서는 산발적으로 발생하는 실시간 및 비실시간 데이터가 토큰-패싱 방식에 의하여 전송된다. Foundation Fieldbus의 데이터 링크 계층에서는 토큰-패싱 방식으로 동작되는 도중에 스케줄 데이터를 전송해야 하는 시간이 되면 주기적 데이터가 우선적으로 전송되며, 따라서 비주기적 데이터의 전송은 일시 중지된다. 본 논문에서 제시하는 대역폭 할당은 기존에 제시된 순환 서비스 방식으로 동작되는 필드버스의 대역폭 할당 기법[3]을 Foundation Fieldbus의 프로토콜 특성에 맞게 적절히 변형한다.

그림 1에는 Foundation Fieldbus에서 대역폭 할당 기법을 통하여 필드버스의 대역폭을 할당하는 예가 나타나 있다. 대역폭 할당 기법의 기본개념은 N_s 개의 스케줄 데이터를 생성하는 노드가 $\gamma(\gamma \leq N_s)$ 개의 윈도우를 서로 동적으로 공유하도록 하여 T_1 구간 동안 생성되는 스케줄 데이터의 개수가 γ 개를 초과하지 않는 동시에 데이터의 최대 허용 지연시간인 ϕ_i 를 초과하지 않도록 각 노드에서 데이터의 생성 주기인 $T_i(i=1$ 에서 $N_s)$ 와 첫 번째 데이터의 생성시간 $t_i(t=1$ 에서 $N_s)$ 를 결정하는 것이다. LAS에서 CD DLPDU를 분배하는 스케줄 시작 시간 s_i 와 주기 S_i 는 각각 각 노드의 t_i 와 T_i 에 매핑된다.

[3]의 순환서비스 방식에서 제시된 대역폭 할당 기법에서는 T_1 의 구간 내에서 γ 개의 주기적 데이터를 연속적으

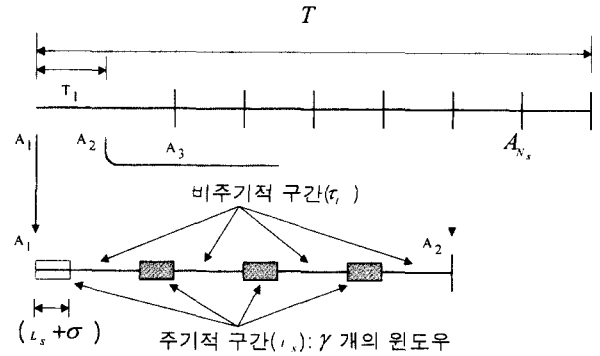


그림 1. Foundation Fieldbus에서 대역폭 할당 기법의 예.
Fig. 1. An example of bandwidth allocation scheme in the Foundation Fieldbus.

로 전송하는 방식을 채택하고 있다. 그러나 Foundation Fieldbus에서 이러한 방식을 그대로 적용하면 비주기적 구간에서 전송되는 산발적 실시간 데이터의 전송이 지연된다. 따라서 본 논문에서 제시하는 Foundation Fieldbus의 대역폭 할당 기법에서는 그림 1에서 보는 바와 같이 γ 개의 윈도우를 T_1 의 주기내에 골고루 분산시키고 분산된 비주기적 구간에서 토큰-패싱을 이용하여 산발적 실시간 데이터와 비실시간 데이터를 전송하도록 하여 산발적으로 발생하는 실시간 데이터의 전송 지연시간에 대한 성능을 향상시킨다. 그림 1에서 보는 바와 같이 주기적 구간 τ_s 는 스케줄링된 하나의 주기적 데이터를 전송할 수 있는 윈도우로 구성되며, 윈도우의 크기는 스케줄 데이터의 전송시간(L_s)과 스케줄 데이터의 전송을 위한 LAS에서의 토큰 처리 시간(σ)의 합이다($\tau_s = L_s + \sigma$). 윈도우가 끝나게 되면 비주기적 구간 τ_j 가 시작되며, 비주기적 구간은 새로운 윈도우가 시작될 때까지 계속된다.

N_s 개의 노드에서 생성되는 주기적 데이터의 최대 허용 지연시간은 다음의 벡터 Φ 로 주어진다고 가정한다.

$$\Phi = [\phi_1, \phi_2, \dots, \phi_{N_s}] \tag{1}$$

또한, N_s 개의 노드에서 생성되는 주기적 데이터의 생성 주기를 원소로 하는 벡터 T 는 다음과 같이 정의된다.

$$T = [T_1, T_2, \dots, T_{N_s}] \tag{2}$$

여기서 T 와 Φ 는 오름차순으로 정렬한다(즉, $T_i \leq T_{i+1}$, $\phi_i \leq \phi_{i+1}$). 그림 1에서 T_1 은 N_s 개의 노드 중에서 최소 생성 주기이고 T_{N_s} 은 최대 생성 주기이다. 대역폭 할당 기법 [3]으로부터 T_i 는 다음과 같이 결정된다.

$$T_1 = \phi_1 \tag{3}$$

$$T_i = k_i T_1, \quad k_i = \left\lfloor \frac{\phi_i}{T_1} \right\rfloor, \quad \forall i = 2 \text{에서 } N_s \tag{4}$$

여기서 $y = \langle x \rangle$ 는 y 가 x 를 초과하지 않으나 x 와 가장 가까운 2의 멱(즉, 2^n , $n \in \{0, 1, 2, L\}$)으로 정의된다. (4)로부터 T_i 는 최대 허용 지연 시간 ϕ_i 를 초과하지 않는 범위 내에서 모든 주기적 데이터의 생성 주기가 서로 배수의 관계를 갖는다. 즉,

$$\text{Rem} \left[\frac{T_i}{T_j} \right] = 0, \quad \forall i, j \quad (j \leq i) \quad (5)$$

따라서 네트워크 대역폭은 T_N 주기로 반복된다. T_1 의 주기내에서 윈도우의 개수 γ 는 다음과 같이 구할 수 있다.

$$\gamma = \lceil \alpha_K \rceil, \quad \alpha_K = \sum_{i=1}^N \frac{T_1}{T_i} \quad (6)$$

여기서 α_K 는 T_1 동안 생성되는 주기적 데이터 개수의 평균값이다.

T_1 의 구간 동안 생성되는 주기적 데이터의 개수가 γ 를 초과하지 않도록 하기 위하여 대역폭 할당 기법은 각 노드의 첫번째 데이터 생성 시간 t_i 를 적절히 선정한다. T_1 의 주기 동안에 전송되는 γ 개의 데이터는 LAS에서 스케줄링에 의한 토큰을 전달하는 CD DLPDU가 도착하기 이전에 미리 전송큐에서 대기하고 있어야 한다. 따라서 δ 만큼의 여유 시간이 필요하다. 그림 1의 A_l (l 에서 N_s)을 T_N 구간 내에서 l -번째 T_1 슬롯이 시작되는 시점이라고 하자. 주기적 데이터를 생성하는 각 노드는 자신의 첫 번째 데이터가 생성되어야 할 시간인 t_i 를 A_l 에 대응시킨다. t_i 는 윈도우 스케줄링 기법[3]으로부터 다음과 같이 결정된다.

$$t_i = \inf [A_l \geq A_{l-1} : u^i(A_l) \leq \gamma] - \delta, \quad (7)$$

$i=1$ 에서 N_s , $l=1$ 에서 T_N/T_1

여기서 t_i ($t_i \leq t_{i+1}$)는 l -번째 노드부터 시작하며, $u^i(A_l)$ 은 i -번째 노드에서 생성된 주기적 데이터를 포함하여 A_l 의 순간에 생성된 주기적 데이터의 개수이다. 식 (7)의 조건은 T_1 구간 내에서 생성되는 주기적 데이터의 개수가 γ 를 초과하지 않도록 제한한다.

대역폭 할당 기법에서 T_1 의 구간동안 γ 개의 주기적 구간과 비주기적 구간이 존재하므로 주기적 데이터의 안정화 조건은 다음과 같이 주어진다.

$$\gamma(\tau_s + \tau_i) \leq T_1 \quad (8)$$

만일 이 조건이 만족되지 않는 경우에는 필드버스 통신망 설계자는 스케줄 데이터 전송큐의 개수 N_s 를 줄여야 한다. 네트워크 설계자는 주기적 데이터를 생성하는 노드 i 의 첫 번째 데이터 생성 시간인 t_i 와 데이터 생성 주기인 T_i 를 기반으로 하여 LAS의 스케줄링 리스트를 작성한다. 스케줄

링 리스트에는 각 노드에 대한 스케줄 시작 시간과 주기는 각각 $s_i = t_i + \delta$ 와 $S_i = T_i$ 로 기록되며, LAS는 이에 따라 해당 노드에 CD DLPDU를 전송한다. 대역폭 할당 기법을 적용하는 경우 모든 노드에서 생성되는 주기적 데이터는 $T_1 (= \phi_1)$ 시간 이내에 전송이 완료되며 따라서 모든 노드에서 최대 허용 지연시간 요구사항 $\phi_i (\geq \phi_1)$ 을 만족한다.

2. 토큰-패싱 데이터의 대역폭 할당

Foundation Fieldbus의 데이터링크 계층에서는 산발적으로 발생하는 데이터의 전송을 지원하기 위하여 토큰-패싱에 의한 데이터 전송 방식을 제공한다. 토큰-패싱에 의하여 전송되는 산발적 데이터의 우선순위는 URGENT, NORMAL 및 TIME-AVAILABLE로 구분되며, 데이터 전송의 우선순위 기능을 부여하기 위하여 사용자가 설정할 수 있는 통신망 파라미터로 TTRT와 PTDD를 제공하고 있다. Foundation Fieldbus에서는 필드버스 통신망 설계자가 TTRT와 PTDD 파라미터의 값을 적절히 설정하여 산발적으로 발생하는 실시간 데이터의 전송 지연시간에 대한 요구사항을 만족하도록 하고 있으나, 규격서[6]에는 이러한 값들을 어떻게 설정할 것인가에 대하여는 명시되어 있지 않다. Foundation Fieldbus에서 제공하는 TTRT 기반의 우선순위 기능은 우선순위가 높은 데이터 전송큐에 더 많은 데이터 전송 기회를 부여하는 확률적 기반으로 동작된다. 타이머 기반으로 동작되는 토큰-패싱 프로토콜에서 TTRT 파라미터의 설정값과 데이터 전송 지연시간에 대한 상관 관계를 구하는 문제는 매우 어려운 문제로 인식되고 있으며[8], 비록 이러한 상관 관계를 구한다 하더라도 데이터 지연 시간에 대한 요구사항은 평균값을 기준으로 만족될 수밖에 없다.

필드버스에서 생성되는 산발적 실시간 데이터 가운데는 주어진 최대 허용 지연시간을 반드시 만족하여야 하는 경우가 있다. 본 논문에서는 특정 노드에서 발생하는 URGENT 우선순위의 산발적 실시간 데이터의 지연시간이 주어진 최대 허용 지연시간 ϕ_c 를 반드시 만족시켜야 하는 경우에 PTDD 파라미터 값을 설정하는 방법을 제시한다. 본 논문에서 URGENT 데이터의 생성 주기는 ϕ_c 를 초과하지 않는 것으로 가정한다. 산발적 실시간 데이터의 최대 허용 지연시간의 요구사항을 만족시키기 위하여서는 LAS에서 네트워크 내의 모든 노드에게 데이터 전송 기회를 부여하는데 소요되는 토큰 순환시간의 최대값 TR_{max} 가 최대 허용 지연시간 ϕ_c 보다 크지 않아야 한다. 즉, ϕ_c 는 다음의 조건을 만족하여야 한다.

$$\phi_c \geq TR_{max} \quad (9)$$

대역폭 할당 기법을 적용하는 경우에 TR_{max} 동안에 최대 전송되는 스케줄 데이터의 개수는 $\lceil TR_{max} \gamma / T_1 \rceil$ 이며, 토큰-패싱 방식에 의하여 각 노드에서 전송되는 데이터의 전송시간은 PTDD로 제한된다. LAS 노드는 다음의 스케줄 시간 동안 한번의 토큰-패싱 서비스를 수행할 시간이 충분하지 않은 경우에 네트워크 관리 서비스를 수행한다. 이러한 네트워크 관리에 소요되는 시간은 $t_m = (\tau_i - \lceil \tau_i / PTDD \rceil)$

PTDD) $\lceil \frac{TR_{max}\gamma}{T_1} \rceil$ 이며, 여기서 $\tau_i = T_1/\gamma - \tau_s$ 는 비주기적 구간의 길이이다. 따라서 토큰 순환시간의 최대값 TR_{max} 는 다음과 같다.

$$TR_{max} = \lceil \frac{TR_{max}\gamma}{T_1} \rceil \tau_s + N_i \times PTDD + t_m + R \quad (10)$$

식 (10)에서 TR_{max} 를 ϕ_c 로 치환하여 식 (9)에 적용함으로써, PTDD는 다음의 관계식을 만족하는 가장 큰 값으로 결정되어야 한다.

$$\phi_c \geq \lceil \frac{\phi_c\gamma}{T_1} \rceil \left(\frac{T_1}{\gamma} \lceil \frac{\tau_i}{PTDD} \rceil PTDD \right) + N_T \times PTDD + R \quad (11)$$

각 노드가 토큰-패싱에 의하여 데이터를 전송할 수 있는 시간은 PTDD로 제한되므로, 그림 1에서 보는 바와 같이 PTDD의 길이는 대역폭 할당 기법에 의하여 주어지는 비주기적 구간의 길이 τ_i 를 초과하여서는 안된다. 또한, PTDD의 시간 동안에 최소한 하나의 비주기적 데이터의 전송이 가능하여야 하므로 PTDD는 $L_i = \max \{L_0, L_1, L_2\}$ 보다는 커야 한다. 따라서 (11)에서 결정된 PTDD가 이러한 조건들을 만족하지 못하는 경우에 네트워크 설계자는 비주기적 데이터의 전송규 개수 N_i 또는 패킷 전송시간 L_i 를 줄여야 한다.

본 논문에서 제시하는 토큰-패싱 서비스에서 네트워크 설계 기법은 URGENT 데이터의 최대 허용 전송 지연시간이 ϕ_c 로 주어진 경우에 적용한다. URGENT 데이터 가운데 ϕ_c 의 요구사항을 반드시 만족할 필요가 없는 데이터와 NORMAL 및 TIME-AVAILABLE 데이터의 전송은 TTRT에 의한 우선순위 기능을 이용한다. 이러한 경우에 데이터 지연시간의 요구사항은 평균값을 기준으로 만족될 수밖에 없다. Foundation Fieldbus에서 대역폭 할당 기법에 의한 통신망 설계 알고리즘을 정리하면 다음과 같다.

Given :

$$N_i, [N_j, j=0\sim 2], N_s, [L_j, j=0\sim 2], L_s, \sigma, R, \delta, \phi_c, [\phi_i, i=1\sim N_s], [\lambda_j^i, j=0\sim 2, i=1\sim N_j]$$

Step 1: 스케줄 데이터의 스케줄 주기 S_i 를 구한다.

$$\phi_1 = \min [\phi_i, i=1\sim N_s]$$

$$T_1 = \phi_1$$

$$k_i = \lceil \frac{\phi_i}{T_1} \rceil, \forall i=1\sim N_s$$

$$\alpha_k = \sum_{i=1}^{N_s} \frac{1}{K_i}$$

$$\gamma = \lceil \alpha_k \rceil$$

$$T_i = k_i T_1, \forall i=2\sim N_s$$

만일 $\gamma(\tau_s + \tau_1) > T_1$, 주기적 데이터 트래픽의 과부하

$\Rightarrow N_s$ 감소: go to Step 1

$$S_i = T_i$$

Step 2: 스케줄 데이터의 첫 번째 스케줄 시작 s_i 를 구한다.

$$t_1 = A_1 = 0$$

For ($i=2, l=1; i \leq N_s, l \leq k_N; i=i+1, l=l+1$)

$$t_i = \inf [A_i \geq A_{l-1}; u^i A_l \leq \gamma] - \delta$$

$$s_i = t_i + \delta$$

Step 3: 다음의 관계식을 만족하는 최대값으로 토큰-패싱 서비스의 PTDD를 구한다. (ϕ_c 가 주어진 경우)

$$\phi_c \geq \lceil \frac{\phi_c\gamma}{T_1} \rceil \left(\frac{T_1}{\gamma} \lceil \frac{\tau_i}{PTDD} \rceil PTDD \right) + N_T \times PTDD + R$$

만일 $PTDD > \tau_i$ 또는 $PTDD < L_i$, 비주기적 데이터 트래픽의 과부하 $\Rightarrow N_i$ 또는 L_i 감소 : go to Step 3

End

IV. Foundation Fieldbus 데이터링크 계층 구현

본 연구에서는 III 장에서 제시한 대역폭 할당에 의한 필드버스 설계기법의 타당성을 검증하기 위하여 실험 모델을 구축하였다. 실험 모델은 Foundation Fieldbus의 물리계층과 데이터링크 계층이 구현된 보드들로 구성된다. Foundation Fieldbus의 물리계층은 전용 ASIC 칩인 FB3050[9]을 사용하여 구현하였다. 그리고 데이터링크 계층 프로토콜은 프레임 분석과 송·수신, 에러처리, 흐름제어, 매체접속권한 관리 기능을 소프트웨어로 구현하였다. 데이터링크 계층의 에러제어 방식은 DLPDU 레벨의 에러검출을 위해 다중비트 FCS(Frame Check Sequence) 계산을 사용하며, 토큰 전달 및 데이터 송/수신 중 에러 검출 여부와 재전송 요구를 위해 송/수신측 모두 타이머와 일련번호(sequence number)를 이용한다. 데이터링크 계층에서는 신뢰성 있고 원활한 데이터 전달의 흐름을 위해 내부 변수를 이용하여 전송되는 DLSDU의 수를 제한하거나, 시간 동기화를 통하여 흐름을 제어한다. 또한 수신측에서의 처리시간 및 inter-communication 지연시간 등을 알 수 있도록 하는 피드백 메커니즘을 사용한다. 구현된 보드는 Dual-Port RAM을 가지고 있으며 PC 슬롯에 장착할 수 있게 설계하였다.

1. 데이터링크 계층 인터페이스 보드 구현

본 연구에서 PC 인터페이스 보드는 INTEL사의 386EX CPU와 필드버스 제어기인 FB3050을 기반으로 구현하였다. FB3050은 전송 속도에 따라 별도의 MAU(Medium Attachment Unit) 회로를 통하여 필드버스 라인에 접속된다. 본 연구에서 FB3050은 1Mbps의 전류모드로 동작하며 이에 적합한 별도의 MAU 회로를 구현하였다. 프로그램 저장을 위한 플래시 롬(29C020)과 프로그램 실행시 필요한 4개의 1Mbit S-램(K6R1008C1C)을 사용하고, 프레임의 송·수신시 이를 저장할 수 있는 256Kbit S-램(IS61C256AH)을 사용하였다. PC 인터페이스 보드에는 데이터링크 계층까지만 구현되며, 상위계층은 PC 기반에서 동작을 하므로 이들과의 데이터 교환이 필수적이다. 이러한 인터페이스를 위해 2KB의 Dual-Port RAM(IDT71321SA)을 사용하였다. INTEL386EX는 내부적으로 2개의 하드웨어 인터럽트 컨트롤러 82C59를 내장하고 있어 최대 15개의 인터럽트를 처리할 수 있다. 외부소자에서 CPU로 요청하는 하드웨어적인 인터럽트는 FB3050에서 요청하는 프레임 송·수신 관련 인터럽트와 1ms 기준

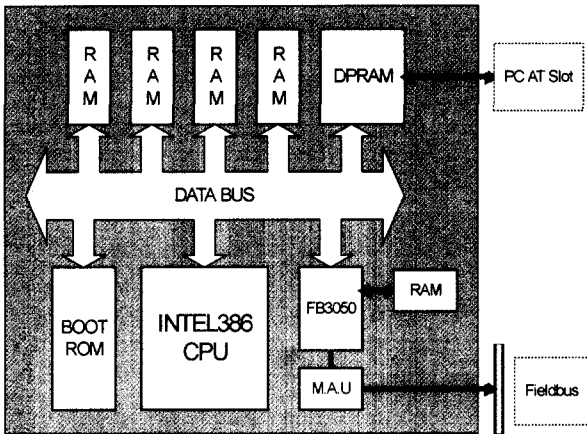


그림 2. PC 인터페이스 보드의 구조도.
Fig. 2. The structure of PC interface board.

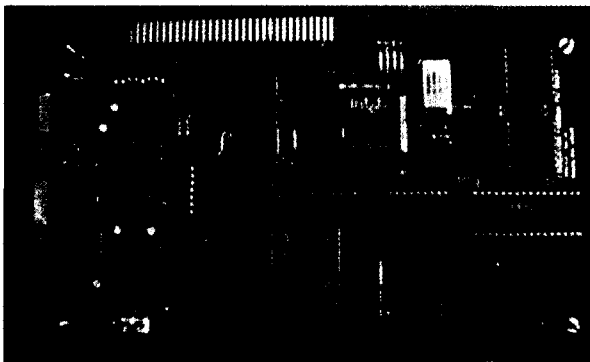


그림 3. PC 인터페이스 보드의 사진.
Fig. 3. The picture of PC interface board.

Timer, Octet Timer, Time Out Timer, TRT(Token Rotation Timer)에서 요청하는 인터럽트 및 DPRAM에서 요청하는 인터럽트로 크게 구분할 수 있다. 그림 2와 3에는 각각 PC 인터페이스 보드의 구조도와 사진이 나타나 있다.

2. 데이터링크 계층 소프트웨어 구현

데이터링크 계층(DLL: Data Link Layer)에는 매체 접근 제어(MAC : Medium Access Control)와 논리적 연결 제어(LLC: Logical Link Control) 기능이 있다. 매체 접근 제어는 매체를 이용하고 있는 노드 중에서 오직 하나의 노드만이 데이터를 전송할 권한을 가지도록 제어하는 역할을 하며, 논리적 연결 제어는 통신을 하려는 노드들간에 논리적 연결을 설정하고 해제하는 기능을 비롯하여 노드들간의 통신과정에서 발생할 수 있는 각종 오류를 처리함으로써 노드간에 신뢰할만한 데이터 전송이 이루어지도록 하는 역할을 수행한다.

데이터링크 계층 프로그램은 표 1에서처럼 상위계층인 FAS(Fieldbus Access Sublayer) 인터페이스 함수, 인터럽트 함수, 프레임 처리 함수, 상태 천이 함수, 매체 접속 권한 함수, 메모리 관리 함수, 타이머 함수, 물리 계층 인터페이스 함수 등 8부분으로 나누어진다.

데이터링크 계층(DLL)은 상위계층인 FAS(Fieldbus Access Sublayer)에게 Connection-oriented Services, Connectionless Data Transfer, Time Synchronization, Compel Service 등의 통

표 1. 데이터링크 계층 함수의 분류.

Table 1. Classification of the functions of data link layer.

FAS 인터페이스 함수	Receiving_FAS_ReqNRes(), Sending_FAS_IndNCon(), Put_mem_queue(), Get_mem_queue(), Pc_Data()
인터럽트 함수	Po_int(), TimerInterrupt(), _386xTimer1(), _386xTimer2(), DpramInterrupt()
프레임 처리 함수	GetXXframe(), SendXXframe(), CheckAddr(), CheckFC(), GetFrameLength(), CopyFrame(), getch0(), ReadFCpriority(), WriteFCpriority(), ReadFCtoken(), WriteFCtoken()
상태 천이 함수	DLE_state(), DLCEP_state(), Responder_state(), LM_state(), Token_state(), Receiver_state()
매체 접속 권한 함수	xmission_CD(), xmission_PT(), Receive_CD(), Receive_PT(), monitoring_response(), SendNextPT(), InitPTparameter(), PT_monitoring(), Receive_RT(), Receive_RI()
메모리 관리 함수	Copy2UsQ(), getUsQ(), Copy2UrgentQ(), getUrgentQ(), Copy2NormalQ(), getNormalQ(), Copy2TaQ(), getTaQ(), getLengthSDU()
타이머 함수	DLL_timer_init(), StartTimer(), StopTimer(), Receive_CT(), Receive_TD()
물리계층 인터페이스 함수	GetPhLchar(), ReadPhLReqStatus(), ReadPhLIndStatus(), Sending_PhL_ReqNRes(), Receiving_PhL_IndNCon(), TmpXmtBuffer2FBC()

신 서비스를 제공한다. 인터럽트 함수는 각 인터럽트에 대한 처리 함수들이다. 여기에는 필드버스 제어기인 FB3050에 관련된 인터럽트 처리 함수와 각종 타이머 및 DPRAM 인터럽트 처리 기능들이 포함된다. 프레임 처리 함수는 수신되어 그 종류가 분석된 각 DLPDU를 구조체에 복사하거나 전송을 위해 각 DLPDU들을 구성하는 함수이다. 상태 천이 함수들은 데이터링크계층의 상태 변화에 따라 각각의 해당 상태 함수를 호출하는 함수들이다. 매체 접속 권한 함수는 데이터 전송 권한을 가진 CD나 PT DLPDU를 전송하거나 수신하였을 때 실행되는 함수들이다. 메모리 관리 함수는 DLL에서 사용되는 우선순위에 따른 큐와 버퍼를 관리하는 함수이다. 타이머 함수는 DLL에서 사용되는 각종 타이머의 동작에 관한 함수들이다. 물리계층 인터페이스 함수는 FB3050 필드버스 제어기로부터 수신된 프레임을 1차적으로 처리하는 함수이다. 그림 4는 데이터의 흐름에 따른 함수들의 호출 경로와 데이터링크계층의 소프트웨어의 전체 구조를 나타내고 있다.

V. 실험 및 결과

본 장에서는 대역폭 할당 기법의 타당성을 실험 모델을 통하여 검증한 결과를 기술한다. 실험 모델은 열 개의 노드

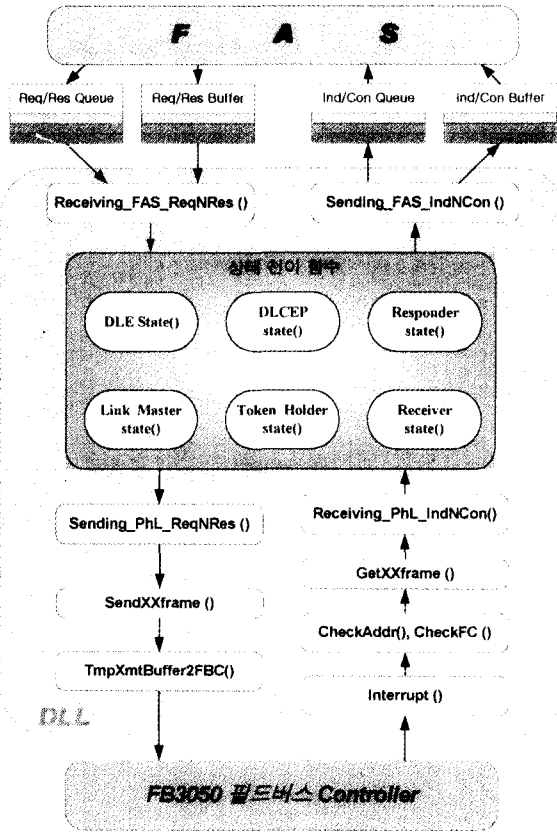


그림 4. 데이터링크계층 소프트웨어 구조 및 인터페이스.
Fig. 4. Software structure and interfaces of data link layer.

와 프레임 모니터링을 위한 모니터 보드로 구성된다. 그림 5에는 본 논문에서 사용된 실험 모델의 구성도가 나타나 있다. 실험 모델에서 모든 노드들은 주기적 데이터와 URGENT, NORMAL, TIME-AVAILABLE 데이터를 생성하며 ($N_s = N_t = N_0 = N_1 = N_2 = 10$), 각 보드에는 데이터 생성 프로그램이 탑재되어 있다. 그림 5의 구성도에서 10번 노드는 본래의 LAS의 기능 이외에 노드들 간에 시간 동기화를 위한 시간-마스터 노드 역할을 병행하여 수행한다.

모니터 노드는 매체를 통하여 전송되는 프레임의 상태를 모니터링하기 위한 노드이다. 각 노드에서 생성되는 메시지의 데이터 필드에는 프레임 생성 시간과 프레임 전송 시간이 기록되며, 모니터 보드는 매체를 통하여 전송되는 모든 프레임을 수신하여 PC에게 전달해주는 기능을 수행한다. PC에서는 이러한 프레임들을 분석하여 데이터 지연시간을

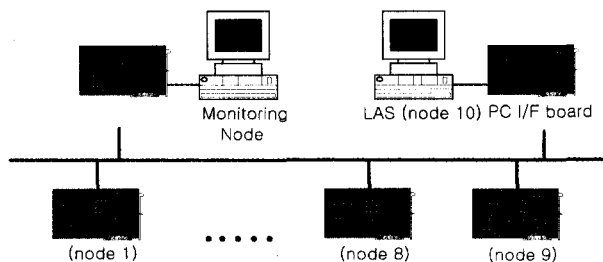


그림 5. 실험 모델의 구성도.
Fig. 5. Configuration of the experimental model.

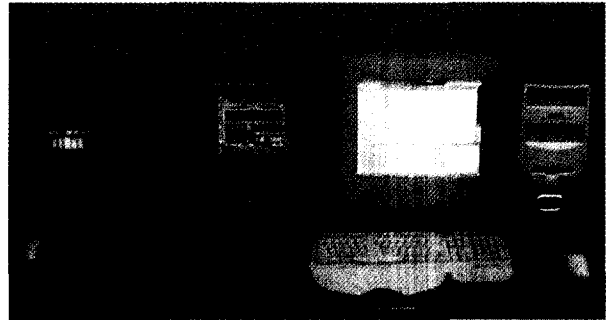


그림 6. 실험모델의 사진.
Fig. 6. Picture of the experimental model.

측정한다. 그림 6에는 실험 모델의 사진이 제시되어 있다.

다음의 표 2에는 본 실험에 사용된 네트워크 트래픽의 조건이 나타나 있다. 구현된 데이터링크 계층 프로토콜의 전송속도는 1Mbps로 설정되었으며, 스케줄 데이터의 길이는 68바이트(헤더:4바이트, 순수 데이터:64바이트)이다. 산발적으로 발생하는 데이터의 경우 URGENT 데이터의 길이는 71바이트(헤더:7 바이트, 순수 데이터:64바이트), NORMAL 데이터는 135바이트(헤더:7바이트, 순수 데이터:128바이트), TIME-AVAILABLE 데이터는 263바이트(헤더:7바이트, 순수 데이터:256바이트)이다. 실제 전송되는 데이터의 전송시간은 실험으로 측정된 결과 스케줄 데이터는 0.58msec, 우선 순위 URGENT의 데이터는 0.6msec, NORMAL 데이터는 1.1 msec, TIME-AVAILABLE 데이터는 2.1msec의 전송시간을 가진다. 이는 물리계층에서 FCS(Frame Check Sequence)와 시작 및 끝 구획문자(delimiter)를 포함한 것이다. 스케줄 데이터는 하나의 데이터 패킷을 수용할 수 있는 버퍼를 통하여 전송되며, URGENT, NORMAL, TIME-AVAILABLE 데이터 전송큐의 크기는 각각 4Kbytes, 8Kbytes, 16 Kbytes로 설정되었다. 이는 각 전송큐에서 약 60개의 데이터 패킷을 수용할 수 있는 크기이다. 본 실험에서는 TIME-AVAILABLE 데이터 전송큐에서 포화상태로 인한 데이터 손실이 발생하기 직전까지 데이터 도착 빈도를 증가시켰다. 따라서 본 실험의 트래픽 부하는 네트워크의 대역폭 자원을 거의 모두 활용하는 경우이다.

표 2. 실험 모델에서 네트워크 트래픽 조건.
Table 2. Network traffic condition in the experimental model.

항 목	값
전송 속도	1Mbps
스케줄 데이터 전송 시간(L_s)	0.58msec
URGENT 데이터 전송 시간(L_0)	0.6msec
NORMAL 데이터 전송 시간(L_1)	1.1msec
TIME-AVAILABLE 데이터 전송 시간(L_2)	2.1msec
URGENT 데이터 도착 빈도(λ_0)	0.0025msec^{-1}
NORMAL 데이터 도착 빈도(λ_1)	0.01msec^{-1}
TIME-AVAILABLE 데이터 도착 빈도(λ_2)	0.0067msec^{-1}
토큰 오버헤드(σ)	2msec
스케줄링 여유 시간(δ)	1msec

실험 모델에서 산발적 실시간 데이터의 최대 허용 데이터 지연시간은 $\phi_c = 200\text{msec}$ 으로 설정되었으며, 10개의 노드에서 생성되는 스케줄 데이터의 최대 허용 지연시간은 $\phi = [64, 80, 130, 170, 260, 300, 520, 600, 1100, 1500] \text{msec}$ 로 주어지는 것으로 하였다. 이러한 트래픽 조건에 대하여 III장에서 제시한 대역폭 할당 기법 알고리즘의 Step 1을 적용하여 스케줄 데이터의 스케줄 주기를 구하면 다음과 같다.

$$S_i = [64, 64, 128, 128, 256, 256, 512, 512, 1024, 1024] \text{msec}.$$

또한, Step 2로부터 각 노드에서 스케줄 데이터의 첫 번째 스케줄 시작 시간 s_i 를 구하면 다음과 같다(LAS 노드의 로컬 링크 기준시간).

$$s_i = [0, 0, 0, 0, 64, 64, 192, 192, 448, 448] \text{msec}$$

토큰-패싱 서비스의 PTDD는 Step 3으로부터 다음과 같이 결정된다.

$$PTDD = 12.9\text{msec}$$

위에서 구한 결과에 따라 첫 번째 T_{Ns} 주기 동안에 10개의 노드에서 생성되는 스케줄 데이터의 데이터 생성 순간이 그림 7에 나타나 있다. 그림에서 보는 바와 같이 각각의 T_1 슬롯 내에서 생성되는 스케줄 데이터의 개수는 $\gamma = 4$ 개를 초과하지 않으며, 이러한 데이터 생성 패턴은 T_{Ns} 의 주기로 반복된다. III장에서 설명한 바와 같이 각 노드에서 전송하는 데이터는 CD DLPDU가 도착하기 전에 미리 전송큐에 대기하고 있어야 하므로 아래 그림에서 δ 시간 이전에 데이터가 생성된다.

표 3에는 실험 모델을 통하여 측정한 스케줄, URGENT, NORMAL 및 TIME_AVAILABLE 데이터의 평균값과 최대값이 나타나 있다. 본 실험에서 TTRT는 ATRT의 최대값인 $TR_{\max} = 200\text{msec}$ 보다 큰 값인 250msec 로 설정되었으며, 따라서 토큰-패싱 서비스에서 URGENT, NORMAL 및 TIME_AVAILABLE 데이터는 토큰의 우선순위에 상관없이 전송된다. 이는 URGENT 데이터의 전송이 하위 우선순위 데이터의 전송에 가장 크게 영향을 받는 최악의 경우를 고려한 것이다. 그림 8과 그림 9에는 스케줄 데이터와 URGENT 데이터의 분포도가 나타나 있다. 표 3과 그림 8에서 보는 바와 같이 주기적 데이터의 전송 지연 시간은 모든 노드에서 64msec 을 초과하지 않는 것으로 나타났다. 또한 URGENT

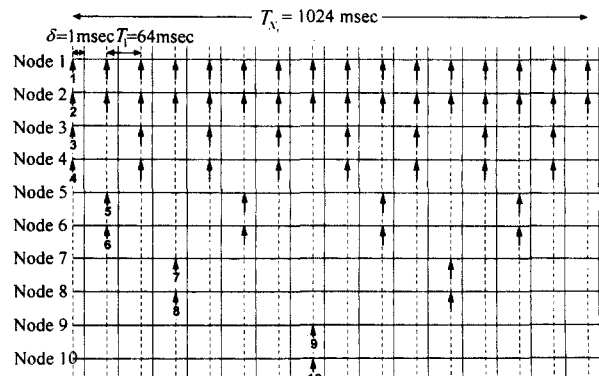


그림 7. 스케줄 데이터의 생성 패턴.
Fig. 7. Generation pattern of schedule data.

표 3. 실험 모델에서 데이터 지연 시간.
Table 3. Network-induced delay in the experimental model.

스케줄	데이터 지연시간(msec)	
	평균값	최대값
스케줄	30.6	63
URGENT	75.90	198
NORMAL	117.97	761
TIME-AVAILABLE	300.65	994

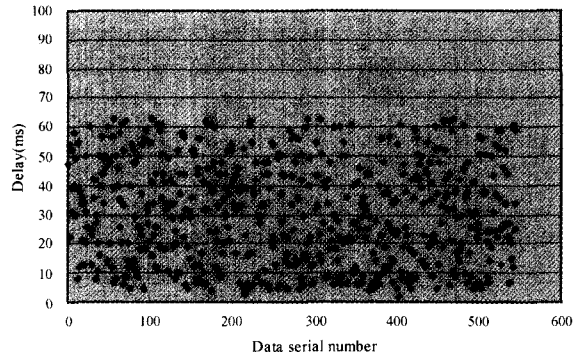


그림 8. 스케줄 데이터 지연시간의 분포.
Fig. 8. Delay distribution of schedule data.

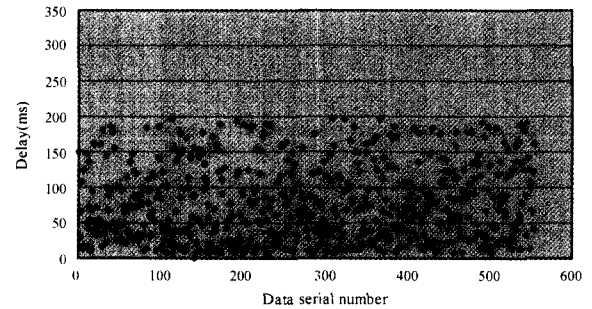


그림 9. URGENT 데이터 지연시간의 분포.
Fig. 9. Delay distribution of URGENT data.

데이터의 전송 지연시간도 그림 9에서 보는 바와 같이 $\phi_c = 200\text{msec}$ 을 초과하지 않는다.

그림 10에는 TTRT의 변화에 대한 URGENT, NORMAL, TIME_AVAILABLE 데이터의 평균 지연시간이 나타나 있다. 그림 10에서 보는 바와 같이 TTRT를 감소할수록 URGENT

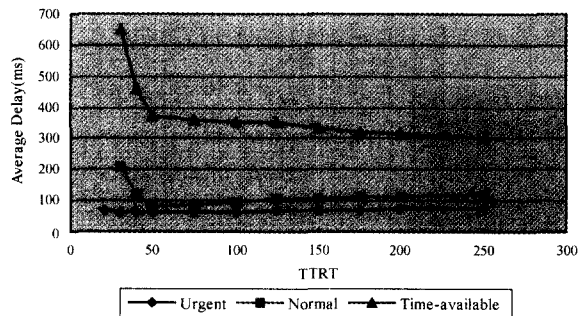


그림 10. TTRT 변화에 대한 평균 데이터 지연 시간.
Fig. 10. Average delay with the change of TTRT.

데이터 지연시간의 평균값은 감소한다. 이는 TTRT를 작게 설정할수록 ATRT가 TTRT보다 커질 확률이 증가되고, 따라서 토큰의 우선순위가 상향 조정되어 URGENT 데이터 전송 큐에서 데이터가 전송될 확률이 증가하기 때문이다. 그림 10에서 보면 TTRT가 감소할수록 NORMAL 데이터의 평균 지연시간은 감소하다가 증가하는 반면에 최하위 우선순위인 TIME-AVAILABLE 데이터는 전송큐에서 데이터를 전송할 기회를 부여받을 확률이 감소하여 평균지연시간은 증가하는 추세를 보인다.

VI. 결론 및 추후 연구 과제

본 논문에서는 Foundation Fieldbus에서 대역폭 할당 기법을 통한 네트워크 설계 기법을 제시하였고, 이에 대한 타당성을 실험을 통하여 검증하였다. 실험 모델의 결과에 의하면 대역폭 할당 기법은 이용 가능한 필드버스의 대역폭을 충분히 활용하는 높은 트래픽 부하에서도 스케줄 및 URGENT 데이터의 데이터 지연 시간이 미리 지정된 최대 허용 한계 값을 초과하지 않는 것으로 나타났다. 본 논문에서 제시하는 대역폭 할당 기법은 Foundation Fieldbus 프로토콜 자체에서 제공하는 스케줄 리스트 및 네트워크 파라미터에 적절한 값을 설정해주면 되므로 기존의 Foundation Fieldbus 시스템에 바로 적용될 수 있다.

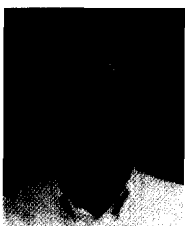
본 논문에서는 또한 TTRT 파라미터의 값을 조정함으로써 토큰-패싱 방식으로 전송되는 URGENT, NORMAL 및 TIME-AVAILABLE 데이터의 평균 데이터 지연시간이 조절될 수 있음을 실험 모델을 통하여 보였다. 본 연구의 후속 연구로 토큰-패싱 방식에서 데이터 지연시간의 요구사항이 평균값으로 주어지는 경우에 이를 만족시키도록 하는 최적의 TTRT 파라미터의 값을 결정하는 문제를 연구할 예정이다.

본 연구에서는 Foundation Fieldbus의 마스터 노드용 PC 인터페이스 보드를 개발하였다. 현재 본 연구진은 센서와

액츄에이터 등의 필드 장비를 Foundation Fieldbus에 바로 연결하도록 하는 트랜스듀서 인터페이스 보드를 개발하고 있다. 트랜스듀서 인터페이스 보드로는 CPU는 INTEL사의 16 비트 단일칩 마이크로 컨트롤러인 80196KC를 사용하며, 필드버스 제어기로는 FB3050을 사용한다. 추후 연구로는 트랜스듀서 인터페이스 보드를 활용한 지능형 센서와 액츄에이터를 개발하고, PC 인터페이스 보드를 활용한 개방형 제어기와와의 통합 시험을 수행할 예정이다.

참고문헌

- [1] J. R. Jordan, *Serial Networked Field Instrumentation*, Wiley, 1995.
- [2] Klaus Bender, *PROFIBUS The Fieldbus for Industrial Automation*, Prentice Hall, 1993.
- [3] S. H. Hong, "Bandwidth allocation scheme in the cyclic-service fieldbus networks," *IEEE/ASME Trans. on Mechatronics*, in press.
- [4] 홍승호, 김유철, 김지용, "PROFIBUS에서 대역폭 할당 기법 구현 및 실험적 평가," 제어·자동화·시스템공학 논문지, 제6권, 제11호, pp. 943-954, 2000.11.
- [5] S. H. Hong and W. H. Kim, "Bandwidth allocation scheme in the CAN protocol," *IEE Proceedings-Control Theory and Applications*, vol. 147, pp. 37-44, Jan., 2000.
- [6] *FoundationTM Specification: Fieldbus Message Specification*, Fieldbus Foundation, 1996.
- [7] *IEC 61158-4: Fieldbus Standard for Use in Industrial Control Systems: Data Link Protocol Specification*, IEC, 2000.
- [8] S. H. Hong, "Approximate analysis of timer-controlled priority scheme in the single-service token-passing systems," *IEEE/ACM Trans. on Networking*, vol. 2, pp. 206-215, April 1994.
- [9] *Smar FB3050TM Fieldbus Communication Controller Data Sheet and Application Note*, Smar Research Corporation, 1996.



홍 승 호

1956년 5월 31일생. 1982년 연세대학교 기계공학과(공학사). 1985년 Texas Tech. University 기계공학과(공학석사). 1989년 Pennsylvania State University 기계공학과 졸업(공학박사). 1989년~1992년 한국 전자통신연구소 선

임연구원. 1992년~현재 한양대학교 전자컴퓨터공학부 부교수. 관심분야는 필드버스, 빌딩자동화 통신망, 홈네트워크.



최 인 호

1973년 4월 6일생. 1999년 울산대학교 제어계측공학과(공학사). 2001년 한양대학교 제어계측공학과(공학석사). 2001년~현재 삼성전자(주) 지능시스템연구소 연구원. 관심분야는 필드버스, 자동화 통신망, 분산제어.



문 봉 체

1961년 9월 28일생. 1984년 서울대학교 제어계측공학과(공학사). 1986년 한국과학기술원 전기 및 전자공학과(공학석사). 1991년 한국과학기술원 전기 및 전자공학과(공학박사). 1989년~1994년 삼성SDS 근무. 1995

년~현재 삼성전자(주) 지능시스템연구소 수석연구원. 관심분야는 자동화시스템 및 네트워크.