

# Foundation Fieldbus에서 긴급데이터의 지연시간 성능해석 및 실험적 검증

## Delay Analysis of Urgent Data in the Foundation Fieldbus and Experimental Verification

홍승호\*, 손병관  
(Seung Ho Hong and Byung Kwan Son)

**Abstract** : The data link layer of Foundation Fieldbus provides both token-passing and scheduling services for periodic, time-critical and time-available data. This study developed an analytical model that evaluates the delay performance of urgent data when the data link layer of Foundation Fieldbus provides token-passing service. The validity of analytical model is verified using an experimental model that consists of network interface boards of Foundation Fieldbus. Comparison of analytical and experimental models shows that the analytical model can be utilized in the approximate analysis of the delay characteristics of time-critical data in the Foundation Fieldbus. The analytical model can also be used in the basic design stage of Foundation Fieldbus network system.

**Keywords** : Foundation Fieldbus, token-passing, urgent, performance analysis, experimental model, verification

### I. 서론

산업의 전 분야에서 컴퓨터 기술이 널리 사용되면서 자동화 기능을 수행하는 컴퓨터들을 통신망으로 연결하는 컴퓨터 통신망의 사용이 확산되고 있다. 필드버스는 자동화 시스템에서 필드에 설치된 각종 제어 및 자동화 관련 필드 장비들에서 생성되는 데이터들의 실시간 통신을 지원하는 통신망이다[1]. 자동화 시스템에 필드버스 통신망을 도입하는 경우에 데이터 전송 지연 시간과 단위 시간당 데이터 처리량 등의 네트워크 성능을 미리 분석할 필요가 있으며, 특히 실시간 처리를 요구하는 자동화 시스템에서는 필드버스의 실시간 데이터 전송에 관한 특성을 미리 분석 및 조사할 필요가 있다.

Foundation Fieldbus[2]는 가장 최근에 개발이 완료된 필드버스로서, 주로 공정 제어 분야에서 널리 사용되고 있다. Foundation Fieldbus의 데이터링크 계층에서는 주기적 및 비주기적 데이터의 전송을 위하여 스케줄링과 토큰-패싱에 의한 데이터 전송 방식을 지원한다. 본 논문에서는 Foundation Fieldbus의 데이터링크 계층에서 토큰-패싱 방식으로 데이터를 전송하는 경우에 실시간 전송을 요구하는 긴급 데이터의 전송 지연시간 성능을 분석할 수 있는 수학적 모델을 제시한다. 수학적 모델의 타당성은 Foundation Fieldbus의 데이터 링크 계층이 구현된 통신망 인터페이스 보드들로 구성된 실험 모델을 통하여 검증되었다. 본 연구에서는 또한 수학적 모델과 실험 모델을 통하여 Foundation Fieldbus의 토큰-패싱 방식에서 긴급데이터의 전송지연시간에 대한 특성을 분석하였다.

본 논문의 전체 구성은 6장으로 되어 있다. II장에서는 Foundation Fieldbus의 전체 구조 및 데이터 연결 계층의 프로토콜을 살펴본다. III장에서는 Foundation Fieldbus의 데이터링크 계층 성능 해석을 위한 수학적 모델을 제시하고, IV장에서는 실험모델의 구성에 대하여 기술한다. V장에서는 수학적 모델과 실험 모델을 통하여 도출된 성능 해석 결과를 비교 분석한다. 마지막으로 본 연구의 결론 및 추후 연구 과제에 대해서 VI장에 기술하도록 한다.

### II. Foundation Fieldbus

#### 1. Foundation Fieldbus 통신망 구성

Foundation fieldbus는 물리계층, 데이터링크계층과 응용계층의 기능을 수행하는 통신스택 및 사용자계층으로 구성된다. 물리계층은 전송속도에 따라 31.25 Kbps, 1.0 Mbps 그리고 2.5 Mbps 필드버스로 나뉜다. 31.25 Kbps 필드버스는 레벨이나 온도제어같은 비교적 정적인 시스템에 사용된다. 또한, 버스로부터 전원을 공급받는 기기를 가지고 있는 필드버스에게 내부 보호기능을 지원한다. 1.0 Mbps와 2.5 Mbps 필드버스는 고성능 프로세스 제어, 원격 입·출력 그리고 빠른 속도를 필요로 하는 공장 자동화 시스템에 사용된다.

Foundation Fieldbus의 통신 스택은 DLL(Data Link Layer), FAS(Fieldbus Access Sublayer) 그리고 FMS(Fieldbus Message Specification)로 구성된다. DLL은 필드버스상에서 메시지의 전송을 제어하는 계층으로 필드 버스 메시지의 전송을 LAS(Link Active Scheduler)라고 불리는 버스 스케줄러에 의해서 관리된다. 메시지 전송 방법은 토큰-패싱과 스케줄링의 두 가지 방식을 제공한다. FAS는 DLL과 상위 계층인 FMS를 연결한다. FAS는 토큰-패싱 방식에서 일-대-일 통신을 지원하는 client/server, 토큰-패싱 방식에서 일-대-다 통신을 지원하는 report distribution, 스케줄링 방식에서 일-대-다 통신을 지원하는 publisher/subscriber의 세 가지 통신 서비스

\* 책임저자(Corresponding Author)

논문접수 : 2003. 1. 30., 채택확정 : 2003. 4. 17.

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

손병관 : 한양대학교 전자전기제어계측공학과(nadason@hanmir.com)

※ 본 논문은 2001년 한양대학교 교내연구비 지원으로 연구되었습니다.

를 지원한다. FMS 서비스들은 표준화된 메시지 형식을 이용해서 필드버스 상으로 메시지를 보낼 때 이용된다. FMS가 제공하는 통신 서비스에는 가장 필드 장비의 관리, 연결 관리, 객체 사전 관리, 변수 접근, 사건 관리, 영역 관리, 프로그램 기동 관리 등이 있다.

사용자계층은 기능 블록, 시스템 관리 및 통신망 관리의 세 가지 기능을 지원한다. 기능 블록은 자원 블록, 기능 블록, 트랜스듀서 블록의 세 가지로 나뉜다. 자원 블록은 필드 장비의 기기 이름, 제조업체, 시리얼 번호 등의 기기의 정보를 관리한다. 기능 블록은 제어 시스템에서 자주 사용되는 기능들을 블록화 시켜 놓은 것이다. 기능 블록의 입출력 매개변수는 필드버스를 통해서 연결되며, 각각의 기능 블록은 정확히 스케줄링된 대로 실행된다. 트랜스듀서 블록은 센서 값을 읽거나 출력하는데 있어서 하드웨어로의 명령에 필요한 입·출력 기능을 지원하며, 기능블록과 독립적으로 동작된다. 시스템 관리는 기능 블록의 동기화 기능을 제공한다. 즉 기능블록의 실행 시간이 되면 사용자 계층에 알려 정확한 시간에 실행될 수 있도록 한다. 또한 동일 매체내의 모든 노드들간의 시간 동기화 기능을 수행하고, 새로운 노드에 주소를 할당하는 등의 시스템 차원의 관리 기능을 수행한다. 통신망 관리는 통신 스택에 필요한 통신망 파라미터를 관리하는 기능을 제공한다. 여기에는 각각의 프로토콜 계층에 관한 정보를 비롯하여 LAS 스케줄에 관한 정보들이 포함된다. 또한 통신망 관리 기능에는 통신망 성능관리와 고장관리 기능이 포함된다.

## 2. Foundation Fieldbus DLL

Foundation Fieldbus DLL의 주된 기능은 필드버스 매체로의 메시지 전송 제어이며, LAS(Link Active Scheduler)라는 버스 스케줄러를 통해서 필드버스로의 메시지 접근을 관리한다. 기본적으로는 중앙 제어 방식의 프로토콜을 사용하고 있으나 스케줄링(Scheduling)과 토큰-패싱(Token Passing) 방식에 의한 데이터 전송을 모두 지원한다. Foundation Fieldbus의 DLL에서 모든 프레임은 DLPDU(Data Link Protocol Data Unit)의 형태로 전송된다.

스케줄링 방식은 주로 주기적으로 처리되어야 하는 데이터의 전송에 사용된다. 스케줄 데이터를 전송하기 위하여 관리자는 미리 LAS 노드에 스케줄 데이터를 전송할 노드와 전송시간(전송 시작시간과 전송 주기)을 입력한다. 스케줄 데이터를 전송해야 하는 시간이 되면 LAS는 해당 노드에 CD(Compel Data) DLPDU를 전송한다. CD를 수신한 노드는 미리 스케줄된 데이터를 전송한다. 데이터를 전송한 후에는 RT(Return Token) DLPDU를 이용해서 LAS로 토큰을 반납한다.

토큰-패싱 방식은 주로 비주기적으로 생성되는 데이터의 전송에 사용된다. LAS는 TCL(Token Circulation List)에 입력된 노드 주소의 순서에 따라서 PT(Pass Token) DLPDU를 전송한다. 비주기적 데이터는 긴급(urgent), 보통(normal), 시간 허용(time-available)의 세 가지 우선순위로 구분되며, PT를 수신한 노드는 PT의 우선순위보다 높거나 같은 우선순위를 갖는 데이터를 전송할 수 있다. 이 때 전송되는 전송 데이터의 길이는 PT에 포함된 매개변수인 PT-duration로 제한된

다. PT의 우선순위는 통신망 파라미터인 TTRT(Target Token Rotation Time)와 실제로 토큰이 모든 노드를 방문하는데 걸리는 시간인 ATRT(Actual Token Rotation Time)을 비교해서 결정한다. LAS는 토큰이 TCL에 입력된 모든 노드를 순환할 때마다 매번 ATRT를 측정하며,  $ATRT < TTRT$ 이면 토큰의 우선순위를 감소해서 낮은 우선순위의 데이터가 전송될 수 있도록 하고,  $ATRT \geq TTRT$ 이면 토큰의 우선순위를 증가시켜서 높은 우선순위의 데이터만 전송 가능하도록 한다.

## III. Foundation Fieldbus DLL의

### 성능해석을 위한 수학적 모델

본 논문에서는 Foundation Fieldbus에서 토큰-패싱 방식을 사용하여 비주기적으로 발생하는 긴급데이터의 전송 지연시간의 성능을 분석하기 위한 수학적 모델을 제시한다. 본 논문에서 전송지연시간은 데이터가 전송큐에 도착한 시간부터 전송이 완료된 시간까지로 정의된다. 본 논문에서 전송지연시간은 ATRT의 분포를 기반으로 하여 도출된다. 토큰-패싱의 순환서비스에서 ATRT를 정확히 수학적으로 모델링하는 것은 매우 복잡한 문제로 인식되고 있다[3]. 이는 토큰-패싱과 같은 순환(cyclic) 서비스 방식으로 동작되는 네트워크 시스템에서 ATRT는 이전의 ATRT들에 의하여 직접적으로 영향을 받기 때문이다. 따라서 대부분의 토큰-패싱 프로토콜의 해석은 ATRT의 확률분포가 *i.i.d.*(independent and identically distributed)하다는 가정을 채택하며 [4, 5], 본 논문에서도 ATRT의 확률분포는 *i.i.d.*하다고 가정한다.

Foundation Fieldbus의 토큰-패싱 방식에서는 데이터의 우선순위를 긴급, 보통, 시간허용의 3단계로 구분하나, 본 논문에서는 통신망 시스템을 단순화시키기 위하여 긴급데이터와 시간허용데이터의 두 개의 우선순위가 존재하는 경우로 국한하며, 이들의 우선순위는 각각 0(긴급)과 1(시간허용)로 지정한다. 각 노드는 우선순위 0 또는 1의 전송큐를 가지며, 긴급 데이터와 시간허용 데이터의 도착은 포아송 분포를 갖는 것으로 가정한다. 데이터의 길이는 우선순위별로 일정한 값을 가지며, 동일 우선순위에 대한 각 노드에서의 트래픽 분포는 동일하다. 각 전송큐에서 데이터 전송은 한번에 하나의 데이터만이 전송되는 단일전송방식으로 동작되며, 따라서 토큰의 PT-duration은 우선순위별로 하나의 데이터만 전송할 수 있도록 설정된다. 다음은 본 논문에서 사용되는 기호들이다.

$N_i$  : 필드버스 내에서 우선순위  $i$  데이터를 생성하는 노드의 개수

$L_i$  : 우선순위  $i$  데이터의 패킷 전송 시간

$\lambda_i$  : 우선순위  $i$  데이터 전송큐의 평균 데이터 도착빈도

$R$  : 데이터의 전송 없이 토큰이 링크 내의 모든 노드 한번 순환하는데 걸리는 시간

$\rho_{ij}$  : 토큰의 우선순위가  $i$ 인 경우에 우선순위  $j$  전송큐에 데이터가 존재할 확률

$T(t)$  : ATRT의 분포

$\bar{T}$  :  $T(t)$ 의 평균값

$r(t)$  : 전송큐 끝단에 도착한 긴급데이터가 토큰을 수신하

는데 소요되는 시간의 분포

$P_i$  : 토큰의 우선순위가  $i$ 가 될 확률

$T^i(t)$  : 토큰의 우선순위가  $i$ 인 경우에  $ATRT$ 의 분포

$\overline{T^i}$  :  $T^i(t)$ 의 평균값

$\mu_i$  : 토큰의 우선순위가  $i$ 인 경우에  $ATRT$ 가  $TTRT$ 보다 작을 확률

$p_n$  : 긴급데이터 전송큐에  $n$ 개의 데이터가 존재할 확률

$d_n(t)$  : 긴급데이터 전송큐 내에  $i$ -번째 데이터의 전송 지연시간 분포

$d$  : 긴급데이터의 평균 전송지연 시간

### 1. $ATRT$ (Actual Token Rotation Time) 해석

$ATRT$ 의 확률분포를 구하기 위하여서는 먼저 토큰의 우선순위가 0이 되어서 긴급데이터만 전송하는 경우에  $ATRT$ 의 분포인  $T^0(t)$ 와 토큰의 우선순위가 1이 되어서 긴급데이터와 시간허용데이터 모두를 전송할 수 있는 경우의  $ATRT$ 의 분포인  $T^1(t)$ 을 도출한다. 다음에 토큰의 우선순위가 0이 될 확률  $P_0$ 과 우선순위가 1이 될 확률  $P_1$ 을 구한다.  $ATRT$ 의 확률분포  $T(t)$ 는 다음과 같이 구해진다.

$$T(t) = \sum_{i=0}^1 P_i \times T^i(t) \quad (1)$$

먼저  $T^i(t)$ 는  $ATRT$ 의 분포가  $i.i.d.$  하다는 가정 하에 다음과 같이 구할 수 있다[4, 5].

$$T^0(t) = \sum_{i=0}^{N_0} \binom{N_0}{i} \rho_{00}^i (1 - \rho_{00})^{N_0 - i} \delta(t - (R + iL_0)) \quad (2)$$

$$T^1(t) = \sum_{i=0}^{N_0} \sum_{j=0}^{N_1} \binom{N_0}{i} \binom{N_1}{j} \rho_{10}^i (1 - \rho_{10})^{N_0 - i} (1 - \rho_{11})^{N_1 - j} \delta(t - (R + iL_0 + jL_1)) \quad (3)$$

(2), (3)에서  $\delta(t)$ 는 다음과 같이 정의되는 Impulse 함수이다.

$$\delta(t - a) = \begin{cases} 1 & \text{for } t = a \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

데이터 도착 빈도가 프아송 분포를 가지는 경우에 (2)에서 우선순위가 0인 토큰이 도착했을 때 우선순위가 0인 긴급데이터 전송큐에 데이터가 존재할 확률  $\rho_{00}$ 은 다음과 같다[6].

$$\rho_{00} = \lambda_0 \overline{T^0} \quad (5)$$

(5)의  $\rho_{00}$ 를 (2)에 대입하여  $T^0(t)$ 의 평균값인  $\overline{T^0}$ 를 구하면 다음과 같다.

$$\overline{T^0} = \int_0^\infty t \cdot T^0(t) dt = R + L_0 N_0 \lambda_0 \overline{T^0}$$

따라서  $\overline{T^0}$ 는 다음과 같이 간략히 표현된다.

$$\overline{T^0} = \frac{R}{1 - N_0 \lambda_0 L_0} \quad (6)$$

(5), (6)으로부터  $\rho_{00}$ 는 다음과 같다.

$$\rho_{00} = \frac{R \lambda_0}{1 - N_0 \lambda_0 L_0} \quad (7)$$

(3)에서 우선순위가 1인 토큰이 도착했을 때 우선순위 0 데이터 전송큐에 긴급데이터가 존재할 확률  $\rho_{10}$  역시 프아송 분포의 가정을 이용하여 다음과 같이 구할 수 있다.

$$\rho_{10} = \lambda_0 \overline{T^1} \quad (8)$$

여기서  $T^1(t)$ 의 평균값인  $\overline{T^1}$ 는  $\overline{T^0}$ 와 동일한 방법으로 다음과 같이 구하여진다.

$$\overline{T^1} = \frac{R}{1 - N_0 \lambda_0 L_0 - N_1 \lambda_1 L_1} \quad (9)$$

(8), (9)로부터  $\rho_{10}$ 는 다음과 같이 구할 수 있다.

$$\rho_{10} = \frac{R \lambda_0}{1 - N_0 \lambda_0 L_0 - N_1 \lambda_1 L_1} \quad (10)$$

(3)에서 우선순위가 1인 토큰이 도착했을 때 우선순위 1 데이터 전송큐에 시간허용 데이터가 존재할 확률  $\rho_{11}$  역시 (9)의  $\overline{T^1}$ 를 이용하여 다음과 같이 구할 수 있다.

$$\rho_{11} = \lambda_1 \overline{T^1} = \frac{R \lambda_1}{1 - N_0 \lambda_0 L_0 - N_1 \lambda_1 L_1} \quad (11)$$

(7), (10), (11)을 (2), (3)에 대입하여  $T^0(t)$ 과  $T^1(t)$ 의 분포를 구할 수 있다.

다음은 토큰의 우선순위가 0이 될 확률  $P_0$ 와 토큰의 우선순위가 1이 될 확률  $P_1$ 을 도출한다. 토큰의 우선순위는 LAS가  $ATRT$ 와 데이터 링크 계층 파라미터인  $TTRT$ 를 비교해서 결정한다. 토큰의 우선순위가 0인 경우,  $ATRT$ 가  $TTRT$ 보다 크거나 같으면( $ATRT \geq TTRT$ ) 토큰의 우선순위는 0으로 유지되고,  $ATRT$ 가  $TTRT$ 보다 작으면( $ATRT < TTRT$ ) 토큰의 우선순위는 1이 된다. 토큰의 우선순위가 1인 경우,  $ATRT$ 가  $TTRT$ 보다 크거나 같으면( $ATRT \geq TTRT$ ) 토큰의 우선순위는 0으로 감소하고,  $ATRT$ 가  $TTRT$ 보다 작으면 ( $ATRT < TTRT$ ) 토큰의 우선순위는 1로 유지된다.  $ATRT$ 가  $i.i.d$  하다는 가정하에서 이를 식으로 표현하면 다음과 같다.

$$P_0 = (1 - \mu_0)P_0 + (1 - \mu_1)P_1 \quad (12)$$

$$P_1 = \mu_0 P_0 + \mu_1 P_1 \quad (13)$$

여기서  $\mu_i$ 는 토큰의 우선순위가  $i$ 인 경우에  $ATRT$ 가  $TTRT$ 보다 작을 확률이며, 이는  $T^i(t)$ 로부터 다음과 같이 구할 수 있다.

$$\mu_0 = \int_0^{TTRT} T^0(t) dt, \quad \mu_1 = \int_0^{TTRT} T^1(t) dt \quad (14)$$

(12), (13) 및  $P_0 + P_1 = 1$ 의 관계를 이용하여  $P_0$ 은 다음과 같이 도출된다.

$$P_1 = \frac{\mu_0}{1 + \mu_0 - \mu_1}, \quad P_0 = \frac{1 - \mu_1}{1 + \mu_0 - \mu_1} \quad (15)$$

(2), (3), (15)를 (1)에 대입해서  $T(t)$ 의 확률분포를 구하면 다음과 같다.

$$T(t) = \frac{1 - \mu_1}{1 + \mu_0 - \mu_1} \sum_{i=0}^{N_0} \binom{N_0}{i} \rho_{00}^i (1 - \rho_{00})^{N_0 - i} \delta(t - iL_0 - R) + \frac{\mu_0}{1 + \mu_0 - \mu_1} \sum_{i=0}^{N_0} \sum_{j=0}^{N_1} \binom{N_0}{i} \rho_{10}^i (1 - \rho_{10})^{N_0 - i} \binom{N_1}{j} \rho_{11}^j (1 - \rho_{11})^{N_1 - j} \delta(t - iL_0 - jL_1 - R) \quad (16)$$

여기서,  $\rho_{00}, \rho_{01}, \rho_{11}$ 은 (7), (10), (11)로 주어진다.

2. ATRT의 잉여 시간

전송큐 끝단에 도착한 데이터가 토큰을 수신할 때까지 소요되는 시간의 분포  $r(t)$ 는 ATRT의 확률 분포의 잉여시간(residual time)으로부터 다음과 같이 구할 수 있다[6].

$$r(t) = \frac{1 - TR(t)}{\bar{T}} \quad (17)$$

(17)에서  $\bar{T}$ 은 ATRT의 분포인  $T(t)$ 의 평균값으로 (18)과 같으며,  $TR(t)$ 은  $T(t)$ 의 cumulative 함수로 (19)로 구한다.

$$\bar{T} = P_0 \frac{R}{1 - N_0 \lambda_0 L_0} + P_1 \frac{R}{1 - N_0 \lambda_0 L_0 - N_1 \lambda_1 L_1} \quad (18)$$

$$TR(t) = \int_0^t T(t) dt \quad (19)$$

(19)를 (17)에 대입해서  $r(t)$ 를 구하면 다음과 같다.

$$r(t) = \frac{1}{\bar{T}} \left[ 1 - P_0 \sum_{i=0}^{N_0} \binom{N_0}{i} \rho_{00}^i (1 - \rho_{00})^{N_0 - i} U(t - R - iL_0) - P_1 \sum_{i=0}^{N_0} \sum_{j=0}^{N_1} \binom{N_0}{i} \rho_{10}^i (1 - \rho_{10})^{N_0 - i} \binom{N_1}{j} \rho_{11}^j (1 - \rho_{11})^{N_1 - j} U(t - R - iL_0 - jL_1) \right] \quad (20)$$

여기서  $U(t)$ 는 Unit Step 함수로 다음과 같이 정의된다.

$$U(t) = \begin{cases} 1 & \text{if } t \geq 0 \\ 0 & \text{else} \end{cases} \quad (21)$$

3. 긴급데이터의 지연시간

우선순위가 0인 긴급데이터 지연시간의 분포는  $T(t)$ 과  $r(t)$ 로부터 도출될 수 있다. 전송큐의 끝단에 있는 우선순위 0 데이터가 토큰이 도착할 때까지 기다리는 시간의 분포는  $r(t)$ 이다. 따라서  $d_1(t)$ 는  $r(t)$ 와 동일하다. 우선순위 0 전송큐의 끝단에서 두 번째에 위치한 데이터는 첫 번째 데이터의 전송이 완료된 이후 다음 번에 토큰이 도착한 순간에 전송된다. 따라서  $d_2(t) = T(t) \otimes r(t) = \int_0^t T(\tau) r(t - \tau) d\tau$ 이다. 여기서  $\otimes$ 는 convolution integral 연산자이다. 동일한 방법으로  $d_n(t)$ 는 다음과 같이 구할 수 있다.

$$d_n(t) = T(t) \otimes \dots \otimes T(t) \otimes r(t) \quad (22)$$

└── (n-1) ─┘

Khinchine's argument로부터 긴급데이터 전송큐에  $n$ 개의 데이터가 존재할 확률  $p_n$ 은 긴급데이터가 전송큐를 떠나

는 순간에 큐길이의 상태 확률과 같다.  $p_n$ 은 토큰이 도착한 순간에 긴급데이터 전송큐의 길이가  $i$ 일 확률  $\pi_i$ 로부터 다음과 같이 구할 수 있다[6].

$$p_n = \sum_{i=1}^{n+1} \frac{\pi_i}{1 - \pi_0} a_{n+1-i} \quad (23)$$

여기서  $a_m = \{(\lambda_0 L_0)^m / m!\} e^{-\lambda_0 L_0}$ 은  $L_0$ 의 긴급데이터 패킷 전송시간 동안  $m$ 개의 긴급데이터가 도착할 확률이다.  $\pi_i$ 는 embedded Markov chain 기법으로부터 다음과 같이 구할 수 있다.

$$\pi_i = \pi_0 \cdot \pi_{0i} + \sum_{j=1}^{i+1} \pi_j \cdot \pi_{ji} \quad (24)$$

여기서  $\pi_0 = 1 - \lambda_0 \bar{T}$ 이며,  $\pi_{ji}$ 은 토큰의 순환 시간동안 전송큐의 길이가  $i$ 에서  $j$ 으로 변화될 확률로 다음과 같이 표현된다.

$$\pi_{ji} = \begin{cases} \int_{t=0}^{\infty} e^{-\lambda_0 t} \frac{(\lambda_0 t)^{i-j+1} \cdot T(t)}{(i-j+1)!} dt, & j > 0 \\ \int_{t=0}^{\infty} e^{-\lambda_0 t} \frac{(\lambda_0 t)^i \cdot T(t)}{i!} dt, & j = 0 \end{cases} \quad (25)$$

패킷 전송시간을 포함하는 우선순위가 0인 긴급데이터의 평균 지연시간은 다음과 같다.

$$d = \sum_{n=1}^{\infty} p_n \int_0^{\infty} t d_n(t) dt + L_0 \quad (26)$$

IV. Foundation Fieldbus DLL의 성능해석을 위한 실험 모델

본 연구에서는 Foundation Fieldbus DLL의 데이터 지연시간 성능을 실험적으로 측정하기 위하여 선행 연구[7]를 통하여 개발된 Foundation Fieldbus의 통신망 인터페이스 보드 및 데이터링크 계층 소프트웨어를 이용하였다. 통신망 인터페이스 보드에는 Foundation Fieldbus의 DLL 계층이 구현되었으며, INTEL 계열의 386EX가 CPU로 사용하였다. 필드버스 통신을 위해 필드버스 제어기인 FB3050을 사용하였으며, 본 실험에서는 필드버스 물리계층 규격서에서 정의한 1Mbps의 전류모드로 동작하도록 하였다. 통신망 인터페이스 보드는 또한 프로그램 저장을 위해서 플래시 롬(29C020)과 프로그램 실행시 필요한 1Mbit S-램(K6R1008C1C)을 4개

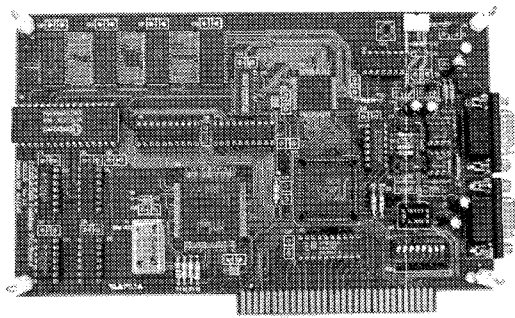


그림 1. 통신망 인터페이스 보드의 사진.  
Fig. 1. Picture of network interface board.

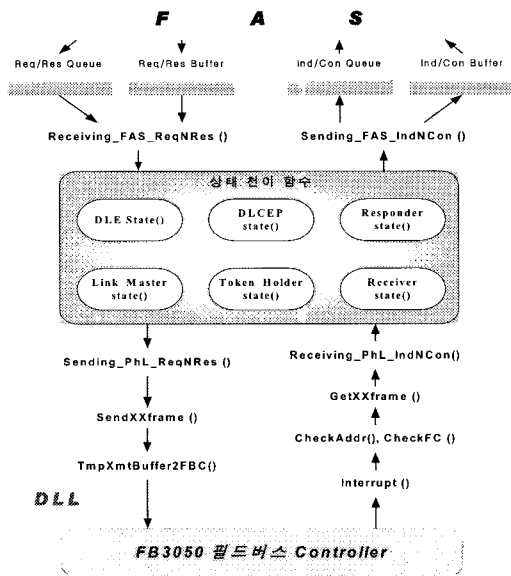


그림 2. DLL 소프트웨어 구조 및 인터페이스.  
Fig. 2. DLL software structure and interface.

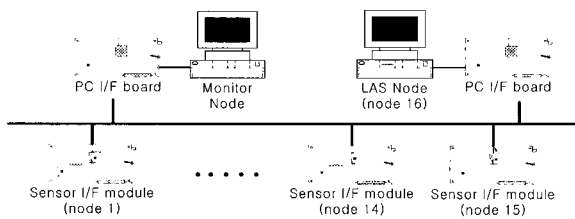


그림 3. 실험 모델의 네트워크 시스템 구성도.  
Fig. 3. Configuration of network system in the experimental model.

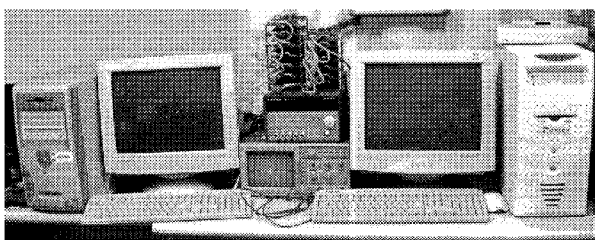


그림 4. 실험 모델 사진.  
Fig. 4. Picture of experimental model.

사용하고, 프레임의 송·수신시 이를 저장할 수 있는 256Kbit S-램(IS61C256AH)을 사용하였다. 통신망 인터페이스 보드의 DLL 상위계층은 PC 기반에서 동작을 하므로 이들과의 데이터 교환이 필수적이다. 이러한 인터페이스를 위해 2KB의 Dual-Port RAM(IDT71321SA)을 사용하였다. 그림 1에는 선행연구를 통하여 개발된 보드의 사진이 나타나 있다. 그림 2에는 DLL의 소프트웨어의 전체 구조와 상위계층과 하위계층간의 인터페이스 및 DLL 계층 내부에서 데이터의 흐름에 따른 함수들의 호출 경로가 나타나 있다.

본 연구에서는 통신망 인터페이스 보드를 이용하여 전송 지연시간의 성능을 측정하기 위한 Foundation Fieldbus 통신

망의 실험 모델을 구성하였다. 실험 모델은 그림 3에서 제시된 바와 같이 16 개의 노드와 프레임을 모니터링하기 위한 모니터 노드로 구성된다. 그림 3의 구성도에서 16번 노드는 본래의 LAS의 기능 이외에 노드들 간에 시간 동기화를 위한 시간-마스터 노드 역할을 병행하여 수행한다. 모니터 노드는 매체를 통하여 전송되는 프레임을 모니터링하기 위한 노드이다. 실험 모델에서 LAS 노드와 모니터 노드에는 통신망 인터페이스 보드가 탑재된다. 이를 제외한 다른 노드들은 센서, 구동기와 같은 필드장비 접속 모듈로 동작되는 것으로 가정한다. 이들은 긴급(우선순위 0) 데이터와 시간허용(우선순위 1) 데이터를 생성하며, 각 보드에는 데이터 생성 프로그램이 탑재되어 있다. 각 노드에서 생성되는 메시지의 데이터 필드에는 프레임 생성 시간과 프레임 전송 시간이 기록되며, 모니터 노드는 매체를 통하여 전송되는 모든 프레임을 수신하여 PC로 전달해주는 기능을 수행한다. PC에서는 이러한 프레임들을 분석하여 데이터 지연시간을 측정한다. 그림 4는 실험 모델의 사진이다.

V. Foundation Fieldbus DLL의 성능해석

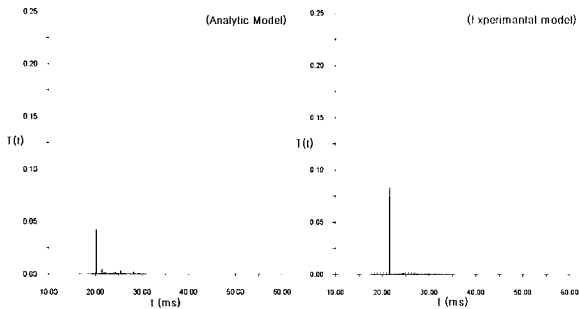
본 장에서는 III장에서 제시한 수학적 모델의 타당성을 IV장에서 기술한 실험모델을 통하여 검증하며, 이러한 결과를 바탕으로 Foundation Fieldbus DLL의 성능을 분석한다. 수학적 모델과 실험 모델에서 데이터 전송 속도는 모두 1 Mbps로 설정되었다. 긴급(우선순위 0) 데이터와 시간허용(우선순위 1) 데이터의 길이는 11 바이트의 헤더를 포함하여 각각 75 바이트와 267 바이트로 설정되었으며, 따라서  $L_0$ 와  $L_1$ 는 각각 0.6 msec와 2.136 msec이다. 각각의 노드에서는 긴급 데이터와 시간허용 데이터를 생성하며 따라서  $N_0 = N_1 = 16$ 이다. 실험 모델에서 노드당 토큰 오버헤드는 1 msec로 측정되었으며, 따라서 데이터의 전송 없이 토큰이 링크 내의 모든 노드를 한번 순환하는데 걸리는 시간  $R = 16$  msec이다. 통신망의 트래픽 부하는  $U = N_0\lambda_0L_0 + N_1\lambda_1L_1$ 로 정의되며, 트래픽 부하량은  $\lambda_0$ 와  $\lambda_1$ 로 조정한다. 본 논문에서 통신망의 성능은  $U = 0.2$ (낮은 트래픽),  $U = 0.35$ (중간 트래픽),  $U = 0.5$ (높은 트래픽)의 세 가지 경우에서 대하여  $TTRT$ , 메시지 발생 주기, 메시지 길이의 변화에 대한 메시지 전송 지연시간의 변화를 측정하였다.

1. TTRT 변화에 대한 긴급데이터 지연시간 변화

본 절에서는  $TTRT$  파라미터값의 변화에 대하여 수학적 모델과 실험 모델에서 얻어진 데이터 전송 지연시간을 분석한다. 그림 5와 6은 트래픽 부하  $U = 0.35$ 에서  $TTRT$ 의 변화에 대한  $ATRT$ 의 확률분포를 나타낸 그래프이다. 여기서  $x$ 축  $t$ 는  $ATRT$ 의 불규칙변수이고,  $y$ 축  $T(t)$ 는  $t$ 에 대한 확률이다. 그림 5와 6의  $ATRT$  분포에서 보듯이  $TTRT$ 의 값이 작게 설정되어있을 경우 우선순위가 높은 긴급 데이터의 전송이 주로 이루어지므로 토큰이 모든 노드를 방문하는 실제 시간인  $ATRT$ 의 값도 작아지게 된다. 따라서  $ATRT$ 의 확률분포도 낮은 시간대에 밀집되어 나타나게 된다. 반면  $TTRT$ 의 값을 크게 설정할수록  $ATRT$ 가  $TTRT$ 보다 작아질 확률이 커지게 되어 토큰의 우선순위가 낮아지는 확률이

증가되므로 시간허용 데이터가 전송될 확률이 커지게 된다. 따라서  $ATRT$  값도 커지게 되고 따라서  $ATRT$ 의 확률분포도  $TTRT$ 의 값이 작게 설정된 경우보다 큰 시간대에 분포되는 형태를 보인다.

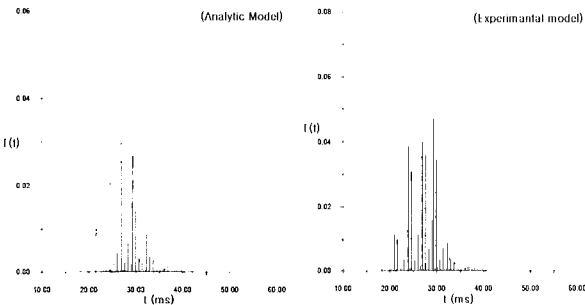
그림 7은  $TTRT$  파라미터 값의 변화에 따른 긴급 데이터의 전송지연시간의 변화를 나타낸 그래프이다. 여기서 가로축은  $TTRT$ 이며 세로축  $d$ 는 긴급 데이터 전송지연시간의 평



(a) 수학적 모델 결과 (b) 실험 모델 결과

그림 5.  $TTRT=16.6$  msec에서  $ATRT$ 의 확률분포.

Fig. 5. Probability density function of  $ATRT$  when  $TTRT=16.6$  msec.



(a) 수학적 모델 결과 (b) 실험 모델 결과

그림 6.  $TTRT = 34.816$  msec에서  $ATRT$ 의 확률분포.

Fig. 6. Probability density function of  $ATRT$  when  $TTRT=34.816$  msec.

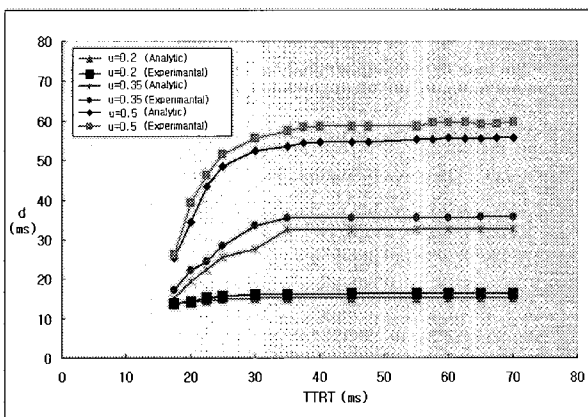


그림 7.  $TTRT$  변화에 대한 긴급 데이터 전송지연시간.

Fig. 7. Delay of urgent data with respect to the change of  $TTRT$ .

균값으로 단위는 msec이다. 그림에서 보면  $TTRT$ 가 감소할수록 긴급 데이터의 전송지연시간은 감소한다. 이는  $TTRT$ 를 작게 설정할수록  $ATRT$ 가  $TTRT$ 보다 커질 확률이 증가되고, 따라서 토큰의 우선순위가 상향조정되어 긴급 데이터 전송큐에서 데이터가 전송될 확률이 증가하기 때문이다. 반면에  $TTRT$ 가 커지게 되면  $ATRT$ 가  $TTRT$ 보다 작아질 확률이 증가하게 되어 우선순위가 하향 조정되기 때문에 데이터 전송시간을 많이 소비하는 시간허용 데이터가 전송될 확률이 커지며, 따라서 긴급 데이터 전송지연시간은 증가한다.

그림 5부터 그림 7에서 보는바와 같이 수학적 모델과 실험모델에서 얻어진 결과의 차이는 크지 않은 것으로 나타났다. 트래픽 부하가 작은 경우에는 수학적 모델과 실험 모델의 결과가 거의 일치하는 것을 볼 수 있다. 트래픽이 증가함에 따라 수학적 모델과 실험 모델에 차이가 나타나는 것에 대한 가장 큰 이유는 수학적 모델에서  $ATRT$  분포가 *i.i.d.*하다고 가정하였기 때문이다. 이러한 결과들로부터 III장에서 제시한 수학적 모델은 Foundation Fieldbus에서 긴급데이터의 근사적 성능 해석에 매우 유용하게 사용할 수 있음이 검증되었으며, 따라서 Foundation Fieldbus 통신망의 기본 설계 단계에서도 매우 유용하게 사용될 수 있다. 상세 설계 및 설계 검증 단계에서는 그러나 시뮬레이션 모델 또는 실험 모델과 같이 정확도가 높은 모델이 요구될 수 있다.

2. 데이터발생빈도( $\lambda_i$ ) 변화에 대한 긴급데이터 지연시간 변화

본 절에서는 네트워크의 트래픽 부하에 영향을 주는 파라미터중 하나인 데이터의 발생 빈도( $\lambda_i$ ) 변화에 따른 데이터 전송지연시간의 변화를 측정하였다. 데이터 발생빈도가 전송지연시간에 미치는 영향을 알아보기 위하여 네트워크 트래픽 부하  $U$ 에 영향을 미치는 나머지 파라미터들( $N_i, L_i$ )을 고정시킨 상태에서 각각의 트래픽에서 긴급 데이터와 시간허용 데이터의 발생빈도  $\lambda_0, \lambda_1$ 을 2:1, 1:1, 1:2의 비율로 변화시키면서 그에 따른 데이터 전송지연시간의 변화를 측정하였다.

그림 8은 데이터 발생빈도 변화에 따른 긴급 데이터 전송지연시간의 변화를 나타낸 그래프이다. 여기서 가로축은 데이터 발생 빈도  $\lambda_0, \lambda_1$ 의 변화이며 세로축  $d$ 는 전송 지연 시간이다. 그림 8에서 보는 바와 같이 동일한 트래픽 부하에서 데이터 발생빈도 변화에 따른 긴급 데이터 전송지연시간의 변화는  $TTRT$ 의 변화에 비해 크지 않다. 그러나 메시지 전송시간을 많이 소비하는 시간허용 데이터가 많이 발생하는 경우에 긴급 데이터 전송 지연시간이 증가하는 것을 볼 수 있다.

그림 9는 그림 8의 그래프를 네트워크 트래픽 부하의 변화에 따라 나타낸 것이다. 여기서 가로축은 네트워크 트래픽 부하의 변화이며 세로축  $d$ 는 전송 지연 시간이다. 그림 9에서 보는 바와 같이 데이터 발생 빈도에 의한 트래픽 부하의 증가는 긴급 데이터의 전송지연시간을 증가시킨다.

3. 데이터전송시간( $L_i$ ) 변화에 대한 긴급데이터 지연시간 변화

본 절에서는 트래픽 부하에 영향을 주는 파라미터중 하

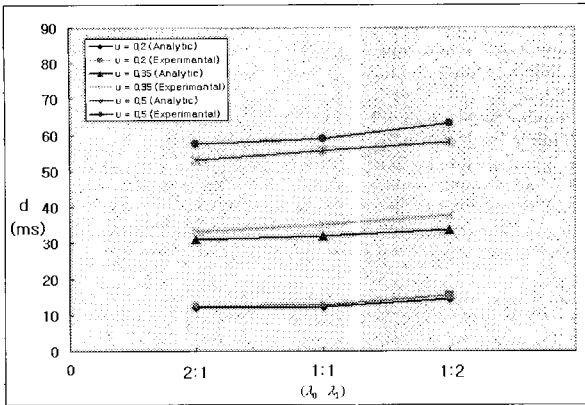


그림 8. 데이터 발생 빈도 변화에 대한 긴급 데이터 전송지연시간.  
 Fig. 8. Delay of urgent data with respect to the change of data arrival rate.

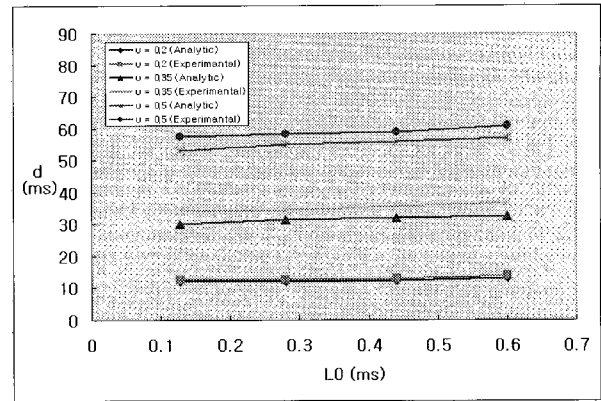


그림 11. 긴급 데이터 전송시간 변화에 대한 긴급 데이터 전송지연시간.  
 Fig. 11. Delay of urgent data with respect to the change of urgent data transmission time.

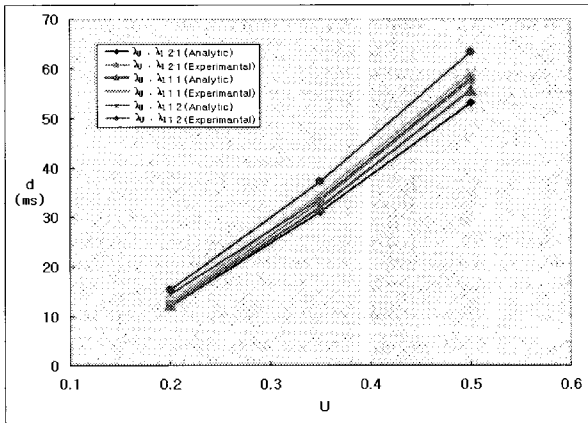


그림 9. 트래픽 부하의 변화에 대한 긴급 데이터 전송지연시간.  
 Fig. 9. Delay of urgent data with respect to the change of traffic load.

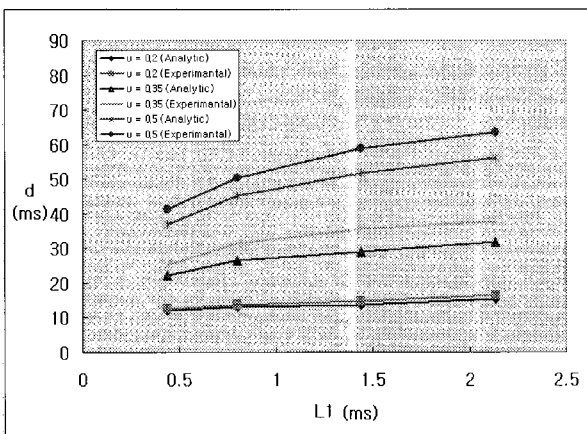


그림 10. 시간허용 데이터 전송시간 변화에 대한 긴급 데이터 전송지연시간.  
 Fig. 10. Delay of urgent data with respect to the change of time-available data transmission time.

나인 데이터 전송시간( $L_1$ )을 변화시키면서 그에 따른 긴급 데이터의 전송지연시간의 변화를 측정하였다. 데이터 전송시간의 변화는 각 데이터의 길이를 변경시킴으로써 그에 따라 데이터 전송에 걸리는 시간에 변화를 주었다. 이때 각 데이터의 발생은 동일한 비율로 발생하고  $TTRT$ 는 48.68 msec로 고정하였다.

그림 10은 시간허용 데이터의 전송시간( $L_1$ )변화에 따른 긴급 데이터 전송지연시간의 변화를 나타낸 것이다. 여기서 가로축은 시간허용 데이터의 전송시간 변화이며 세로축  $d$ 는 긴급 데이터의 전송지연시간이다. 그림에서 보듯이 트래픽 부하가 고정된 상태에서 시간허용 데이터의 길이가 변화함에 따라 긴급 데이터의 지연시간은 크게 변한다. 데이터 전송시간을 많이 소비하는 시간허용 데이터의 길이를 짧게 하는 것이 긴급 데이터의 전송지연시간을 감소시키므로, 동일한 트래픽 부하에서 긴급 데이터의 전송지연시간을 줄이려면 시간허용 데이터의 길이를 가능한 한 짧게 하는 것이 바람직하다.

다음에는 시간허용 데이터의 길이를 고정하고 긴급 데이터의 길이를 변경하여 긴급 데이터의 전송시간에 변화를 주었다. 그림 11은 긴급 데이터의 길이 변화에 따른 데이터 전송지연시간의 그래프이다. 여기서 가로축은 긴급 데이터의 전송시간이며 세로축  $d$ 는 긴급 데이터의 전송지연시간이다. 그림에서 보는 바와 같이 긴급 데이터 길이의 변화가 긴급 데이터의 전송지연시간에 미치는 영향은 크지 않다. 긴급 데이터는 원래 길이가 짧고 가장 높은 우선순위로 전송되므로 긴급 데이터 길이의 변화가  $ATRT$ 의 변화에 크게 영향을 미치지 않는다. 따라서 긴급 데이터의 길이 변화는 다른 네트워크 파라미터들에 비해 전송지연시간에 미치는 영향은 크지 않다는 것을 알 수 있다.

**VI. 결론**

Foundation Fieldbus는 토큰-패싱과 스케줄링 방식을 통하여 주기적 데이터, 비주기적 실시간 데이터 및 비실시간 데이터의 전송을 모두 지원하는 필드버스 통신망이다. 본 논

문에서는 Foundation Fieldbus의 데이터링크 계층이 토큰-패싱 방식을 통하여 비주기적 데이터를 전송하는 경우에 긴급데이터의 전송지연시간 성능을 해석할 수 있는 수학적 모델을 제시하였다. 수학적 모델의 타당성은 Foundation Fieldbus의 데이터링크 계층이 구현된 보드들로 구성된 실험 모델을 통하여 검증되었다. 실험 모델의 결과와 비교에 의하면 수학적 모델은 긴급데이터의 지연시간 성능을 근사적으로 분석하는데 있어서 매우 유용하게 사용될 수 있는 것으로 나타났다.

수학적 모델 및 실험 모델의 성능해석 결과에 의하면 Foundation Fieldbus에서 긴급데이터의 지연시간은  $TTRT$  파라미터의 변화에 의하여 영향을 받으며, 통신망의 트래픽 부하가 증가할수록  $TTRT$  파라미터의 변화에 대한 영향이 증가한다. 긴급데이터의 지연시간 성능요구사항이 주어진 경우에 이를 만족시키기 위한  $TTRT$  파라미터 값을 어떻게 설정하여야 할 것인가는 Foundation Fieldbus의 통신망 설계에 있어서 중요한 문제이다. 본 논문에서 제시한 수학적 모델은 이러한 문제를 해결하기 위한 Foundation Fieldbus 통신망의 기본 설계 단계에서 매우 유용하게 활용될 수 있다. 긴급데이터의 전송지연시간은 트래픽 부하가 증가함에 따라 당연히 증가된다. 트래픽 부하를 고정시킨 상태에서 긴급 및 시간허용 데이터의 발생 주기( $\lambda_i$ )와 데이터 길이( $L_i$ )를 변화시키는 경우에 이러한 파라미터의 변화가 긴급데이터의 지연시간에 미치는 영향은  $TTRT$ 의 변화에 비하여 크지 않다. 그러나 시간허용 데이터 길이의 변화가 긴급데이터의 지연시간에 미치는 영향은 비교적 크므로, Foundation Fieldbus의 설계 시 시간허용 데이터의 길이는 가능한 한 짧게 하는 것이 바람직하다.

본 연구에서는 Foundation Fieldbus의 데이터링크 계층이 토큰-패싱 방식으로 동작되는 경우에 대하여서만 고려하였다. 후속 연구로는 Foundation Fieldbus의 데이터링크 계층이 토큰-패싱과 스케줄링을 모두 지원하는 경우와 토큰-패싱 방식에서도 긴급, 보통, 시간허용의 세 가지 우선순위를 모두 지원하는 경우에 통신망 성능해석을 위한 수학적 모델 및 통신망 설계 기법의 개발에 대하여 연구 범위를 확장할 필요가 있다.

#### 참고문헌

- [1] J. R. Jordan, *Serial networked field instrumentation*, John Wiley & Sons, 1995.
- [2] *FOUNDATION<sup>TM</sup> Specifications: data link protocol specifications*, Fieldbus Foundation, Austin, TX, 1996.
- [3] H. Levy and M. Sidi, Polling systems: applications, modeling and optimization, *IEEE Trans. on Communication*, vol. 38, no. 10, pp. 1750-1760, 1990.
- [4] P. J. Kuehn, Multiqueue Systems with nonexhaustive cyclic service, *Bell Syst. Tech. J.*, vol. 58, no. 31, pp. 671-698, 1979.
- [5] S. H. Hong, Approximate analysis of timer-controlled priority scheme in the single-service token-passing systems, *IEEE/ACM Trans. on Networking*, vol. 2, no. 2, pp. 206-215, 1994.
- [6] P. King, *Computer and communication systems performance modeling*, Prentice Hall, 1990.
- [7] 홍승호, 최인호, 문봉채, “대역폭 할당을 통한 Foundation Fieldbus 통신망 설계” 제어 · 자동화 · 시스템공학논문지, 제 7 권, 제 10 호, pp. 882-890, 2001.



홍 승 호

1956년 5월 31일생. 1982년 연세대학교 기계공학과 (공학사). 1985년 Texa Tech University 기계공학과 (공학석사). 1989년 Pennsylvania State University 기계공학과(공학박사). 1989년~1992년 한국전자통신연구소 선임연구원. 1992년~현재

한양대학교 전자컴퓨터공학부 교수. 관심분야는 필드버스, 빌딩자동화통신망, 홈네트워크.



손 병 관

1975년 2월 11일생. 2001년 한양대학교 제어계측공학과 (공학사). 2001년~현재 한양대학교 전자전기제어계측공학과 석사과정. 관심분야는 필드버스, 자동화통신망, 분산제어.