

# Profibus token passing protocol의 통신 지연 모델 개발

김현희\*, 이경창(부산대 대학원 지능기계공학과), 이 석(부산대 기계공학부)

## Development of communication delay model for Profibus token passing protocol

H. H. Kim, K. C. Lee(Mech. Eng. Dept., PNU), S. Lee(Mech. Eng. School. PNU)

### ABSTRACT

Most networks for automation are tuned to an expected traffic condition at their design stage. During their actual operations, however, the networks experience considerable changes in traffic from time to time. These traffic changes caused by common events like machine failure and production schedule change may adversely affect the network performance and, in turn, the performance of the connected devices. This paper presents communication delay model for Profibus token passing protocol, and introduces TTR selection methods to maintain a uniform level of network performance at all stations under changing network traffic.

**Key Words** : Profibus(프로피버스), Fieldbus(필드버스), Token passing protocol(토큰 패싱 프로토콜), communication delay(통신 지연), desired communication delay(요구 통신 지연)

### 1. 서론

최근 고성능의 마이크로 프로세서가 발달함에 따라 대부분의 제어 시스템은 디지털 제어 시스템으로 구현되고 있다. 컴퓨터를 이용한 디지털 제어 시스템은 뛰어난 유연성 (flexibility)과 적응성(adaptability)을 가지고 있다. 그러나 시스템이 복잡해지면 시스템에서 관리해야 할 데이터 처리량이 증가하게 됨으로써, 단일 시스템으로는 실시간 처리, 엄격한 시간 제약 조건의 만족 및 신뢰성의 확보도 어렵게 되었다. 또한, 기존의 일대일로 연결되는 점대점(point-to-point) 연결방식으로는 너무 많은 배선들을 요구하게 됨에 따라, 시스템의 유지 및 보수가 커다란 문제로 대두되었다.

이러한 문제로 인하여, 점대점 연결방식의 시스템들은 네트워크를 이용한 분산형 연결방식으로 대체되고 있는 추세이다. 즉, 제어기와 필드기기들을 일대일로 연결되는 점대점 방식에서 단일한 전송 매체를 사용하는 산업용 네트워크(industrial network), 즉 필드버스를 이용하여 연결하는 방식으로 전환됨으로써 시스템의 성능 향상과 단순화, 신뢰성 증대 등의 장점을 얻을 수 있게 되었다.[1]

필드버스의 설계에 있어서 각 장치들에서 생성되는 트래픽이나 네트워크 관리자가 만족될 수 있기를 원하는 요구 통신 지연의 특성을 반영하고, 전체적인 시스템의 안정성을 위하여 적절한 네트워크 변수들이 설정되어야 한다.

본 논문에서는 필드버스 프로토콜 중 Profibus-FMS의 전송지연에 관한 수학적 모델을 개발하고, 전송 지연을 각 스테이션별로 주어진 요구 통신 지연에 맞추기 위한  $T_{TK}$ (Target Rotation Time) 선정 방법에 대하여 연구하였다.

본 논문은 4장으로 구성되어 있다. 2장에서는 Profibus-FMS의 전송방법인 토큰 전달 방식(token passing protocol)과 메시지 우선 순위에 관하여 설명하였고, 3장에서는 전송지연의 종류와 Profibus-FMS의 수학적 모델 및  $T_{TK}$  선정 방법에 대하여 설명하였다. 마지막으로, 4장에서는 결과와 향후 과제에 대해서 설명하였다.

### 2. Profibus-FMS and token passing protocol

#### 2.1 Profibus

Profibus는 응용계층의 종류에 따라 FMS, DP, 및 PA로 구분한다. Profibus-FMS는 다양한 실시간 통신 서비스 사용이 가능하며, 다양한 생산시스템에 사용되고 있다. Profibus-DP는 빠른 데이터 전송이 가능하며, 분산된 필드기기들간의 실시간 통신이 요구되는 분산 제어 시스템이나 원격 제어 시스템에 사용되고 있다. 마지막으로, Profibus-PA는 전자기적 간섭에 대한 내성(intrinsic safety)을 가지며, 원자력 발전소와 같은 특수한 환경에 주로 사용되고 있다[2].

Profibus 프로토콜은 마스터/슬레이브 방식과 토

큰 패싱(token passing protocol)을 같이 사용하는 혼성접근방식(hybrid medium access)을 가지고 있다. 그중 FMS에서는 토큰 패싱 전송 방식을 주로 사용한다. 토큰 패싱 방식은 토큰이 스테이션 사이클을 순환하면서 토큰을 가진 스테이션만이 전송의 권한을 갖는 방식이다. 특히, Profibus 프로토콜은 high, low의 두 가지 우선순위를 가진다. 여기에서, high 메시지는 긴급 데이터를 처리할 때 사용하고, low 메시지는 일반적인 상태의 데이터 전송 시 사용한다. 따라서, 한 스테이션이 토큰을 소유한 경우, high 메시지를 우선적으로 전송하며, 전송 용량에 여유가 있는 경우에 low 메시지를 전송한다.

FMS는 전송 용량의 결정을 위하여,  $T_{TR}$ ,  $T_{RR}$  (Real Rotation Time) 및  $T_{TH}$  (Token Holding Time)과 같은 3개의 타이머를 가지고 있다. 여기에서,  $T_{TR}$ 은 토큰이 1회전하는데 예상되는 시간이며,  $T_{RR}$ 은 실제 토큰이 버스를 1회전하는데 걸린 시간이며,  $T_{TH}$ 는  $T_{TR} - T_{RR}$  로서 각 스테이션이 토큰을 수신한 후에 전송할 수 있는 시간을 나타낸다.

그림 1은 FMS의 전송 순서를 나타내고 있다. 그림에서 마스터는 토큰을 받은 후, THT를 계산하고 나서, 하나의 높은 우선순위 메시지 사이클(high priority message cycle)을 수행한다. 하나의 메시지 사이클이 끝나고 난 후, 추가적인 메시지 전송은 THT가 0보다 큰 경우에만 수행될 수 있다. 만약 THT가 종료되었거나, 전송할 메시지가 없는 경우, 마스터는 논리적 링의 순서대로 다음 스테이션에게 토큰을 전송한다.

### 3. Communication delay model of Profibus-FMS

#### 3.1 Notation

- $i$  iteration(순회), token rotation number,  $i = 1 \sim m$
- $j$  station number,  $j = 1 \sim n$
- $k$  message number,  $k = 1 \sim p$
- $T_1$  토큰 전송에 소요되는 시간

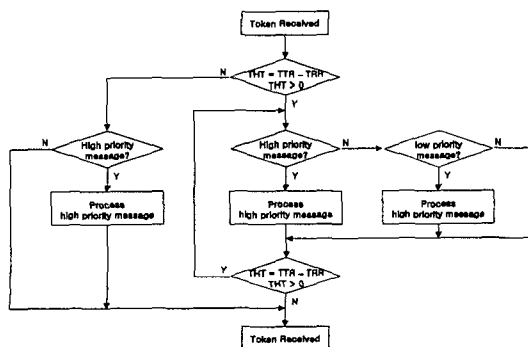


Fig. 1 Communication procedure of Profibus-FMS

$T_C^j$  메시지 사이클에 소요되는 시간

$f_k^j$   $k$ 번째 메시지의 생성 확률 분포

$T_{Gk}^j$   $k$ 번째 메시지가 생성되는 시간

여기에서,  $T_{Gk}^j = T_{Gk-1}^j + f_k^j(x)$

$T_{S_i}^j$  토큰 수신 시간

$T_{E_i}^j$  토큰 송신 시간

$T_{F_k}^j$   $k$ 번째 프레임 전송 시간

$N_{Q_i}^j$  큐에 저장되어 있는 메시지 수

$N_{A_i}^j$   $T_{TH}$  동안 전송 가능한 메시지 수

$N_{T_i}^j$  전송된 메시지 수

$T_{iH_i}^j$   $i$ 번째 스테이션에  $i$ 번째 순회에서의  $T_{TH}$

$T_{iR}^j$   $i$ 번째 스테이션의  $T_{TR}$

$D_k^j$   $k$ 번째 메시지의 전송지연

$C(x)$  소수 첫째자리에서 버림 함수

$U(x)$  소수 첫째자리에서 올림 함수

$T_M^j$   $j$ 스테이션의 메시지 생성 주기

$N_{G_i}^j(a, b)$   $i$ 번째 스테이션에서  $a$ 에서  $b$ 시간 사이에 생성된 메시지의 수, 주기적인 메시지가 생성될 경우는  $N_{G_i}^j(a, b) = C(b-a)/T_M^j$  가 된다.

#### 3.2 Communication delay model

본 논문에서는 다음과 같은 가정을 바탕으로 하여 통신 지연 모델을 개발하였다. 첫째, 전송 큐의 사이즈는 충분히 크기 때문에, 모든 메시지를 큐에 저장할 수 있다고 가정한다. 둘째, 전송 중 에러는 없다고 가정한다. 셋째, 모든 스테이션에서 초기의 토큰 수신 시간과 토큰 전송 시간은 0 이고, 각 스테이션에서 0시간에 메시지가 생성되어 있다고 가정한다.

먼저,  $j$  번째 스테이션에서  $i$  번째 순회에서의 큐에 저장되어 있는 메시지의 수는 다음과 같다.

$$N_{Q_i}^j = (N_{A_{i-1}}^j - N_{T_{i-1}}^j) + N_{G_i}^j(T_{E_{i-1}}^j, T_{E_i}^j) \quad (1)$$

$$= N_{G_i}^j(0, T_{E_i}^j) - \sum_{k=1}^{i-1} N_{T_k}^j$$

식에서, 첫 번째 식은  $i-1$  번째 순회에서 전송하고 남은 메시지 수와  $i-1$  번째 순회에서 토큰을 전송한 후부터,  $i$  번째 순회에서 토큰을 송신하기 직전까지 생성된 메시지의 수의 합이다. 또한, 두 번째 식은 시간 0에서  $i$  번째 순회에서 토큰을 전송하기 직전까지 생성된 메시지의 총 수와 시간 0에서  $i-1$  번째 순회까지 전송된 메시지의 총 합이다.

$i$  번째 스테이션에서  $i$  번째 순회에서의  $T_{HT}$  동안 전송 가능한 메시지의 수는 다음과 같다.

$$N_{Ai}^j = U g\left(\frac{T_{HTi}^j}{T_C^j}\right)$$

$$\text{where, } T_{HTi}^j = T_{TR}^j - (T_{Si}^j - T_{Si-1}^j) \quad (2)$$

$$g(x) = \begin{cases} x, & x > 0 \\ 1, & x \leq 0 \end{cases}$$

식에서,  $g(x)$ 는  $x$ 가 0보다 적은 경우, 즉  $T_{HT}$ 가 0이하인 경우에는 1개의 메시지만 전송하기 때문에 단극램프함수(unipolar ramp function)로 정의한다. 해당 주기에 전송할 수 있는 메시지의 수는 활용 가능한  $T_{HT}$ 를 메시지 사이클 주기로 나눈 값의 소수점 첫째자리에서 올림한 값 만큼이다.

$j$ 번째 스테이션에서  $i$ 번째 순회에서의 전송된 메시지 수는 다음과 같다.

$$N_{Ti}^j = \min(N_{Qi}^j, N_{Ai}^j) \quad (3)$$

식에서, 전송 할 수 있는 메시지의 수는 큐에 저장되어 있는 메시지의 수와  $T_{HT}$ 동안 전송 가능한 메시지의 수의 최소값이다.

$j$ 번째 스테이션에서  $i$ 번째 순회에서의 토큰 수신 시간은 이전 스테이션의 토큰 전송시간을 더한 값으로서 다음과 같이 계산된다.

$$T_{Si}^j = T_{Ei-1}^{j-1} + T_T \quad j \neq 1$$

$$T_{Si}^j = T_{Ei-1}^j + T_T \quad j = 1 \quad (4)$$

다음으로,  $j$ 번째 스테이션에서  $i$ 번째 순회에서의 토큰 송신시간은 토큰 수신 시간에 메시지 전송에 소요된 시간을 더한 값으로 다음과 같이 계산된다.

$$T_{Ei}^j = T_{Si}^j + N_{Ti}^j T_C^j \quad (5)$$

$j$ 번째 스테이션에서  $i$ 번째 순회에서의 전송지연을 구하기 위해서는 토큰 수신 시간, 토큰 송신 시간, 전송된 메시지의 수 등을 계산하여야 한다. 이는 다음과 같은 절차에 의하여 구할 수 있다.

step 1. 식 (4)로부터 토큰 수신 시간( $T_{Ei}^j$ )을 구한다.

step 2. 토큰이 수신된 경우, 식 (1)에서 큐에 저장되어 있는 메시지의 수( $N_{Qi}^j$ )를 구한다.

step 3.  $T_{HTi}^j$ 를 계산하고, 식 (2)에서 전송 가능한 메시지의 수( $N_{Ai}^j$ )를 구한다.

step 4. 식 (3)을 이용하여 전송할 수 있는 메시지의 수( $N_{Ti}^j$ )를 구한다.

step 5.  $N_{Ti}^j$  개수만큼 메시지 사이클을 수행하고 난 후에  $T_{HTi}^j$ 가 종료되었는지를 검사한다. 만약,  $T_{HTi}^j$ 가 종료되지 않은 상태에서 큐에 메시지를 남아있는 경우 step 4를 반복한다.

step 6.  $T_{HTi}^j$ 가 종료된 경우, 식 (5)에서 토큰 송신 시간( $T_{Ei}^j$ )을 구하고, 다음 스테이션으로 토큰을 전송한다.

이상과 같은 방법으로 구한 값들을 이용하여,  $j$ 번째 스테이션에서  $k$ 번째 메시지의 전송지연을 다음과 같이 계산할 수 있다.

$$D_k^j = T_{Fk}^j - T_{Gk}^j \quad (6)$$

$$T_{Fk}^j = T_{Si}^j + k - N_{Gi}^j(0, T_{Ei-1}^j) T_C^j$$

$$k = N_{Gi}^j(0, T_{Ei-1}^j), N_{Gi}^j(0, T_{Ei-1}^j) + 1, \dots, N_{Gi}^j(0, T_{Ei-1}^j) - 1$$

이상에서,  $n$ 개의 스테이션,  $m$ 번째 순회(iteration) 수행 후의 평균 전송 지연은 식 (7)과 같으며, 최대 전송 지연은 식 (8)과 같다.

$$\sum_{i=1}^n \frac{D_i^j}{p} \quad (7)$$

$$\max_i D_i^j \quad (8)$$

만약, 토큰을 수신하면, 토큰을 수신하기 직전까지 생성되어 있는 메시지만 현재 순회(iteration)에서 전송하고, 토큰을 수신한 후 생성된 메시지는 다음 순회에서 전송된다고 가정(네번째 가정)한다면, 식 (1)은 이 상호 독립이 되기 때문에 계산을 단순화시켜 아래와 같이 표현될 수 있다.

$$N_{Qi}^j = (N_{Ai-1}^j - N_{Ti-1}^j) + N_{Gi}^j(T_{Si-1}^j, T_{Si}^j) \quad (9)$$

$$= N_{Gi}^j(0, T_{Si}^j) - \sum_{i=1}^{i-1} N_{Ti}^j$$

또한, 식 (4)의 토큰 수신 시간은 다음과 같이 단순화될 수 있다.

$$T_{Si}^j = T_{Si-1}^{j-1} + T_T \quad j \neq 1$$

$$T_{Si}^j = T_{Si-1}^j + T_T \quad j = 1 \quad (10)$$

마지막으로, 식(6)의  $j$ 번째 스테이션에서  $k$ 번째 메시지의 전송지연은 다음과 같이 단순화될 수 있다.

$$D_k^j = T_{Fk}^j - T_{Gk}^j \quad (11)$$

$$T_{Fk}^j = T_{Si}^j + k - N_{Gi}^j(0, T_{Si-1}^j) T_C^j,$$

$$k = N_{Gi}^j(0, T_{Si-1}^j), N_{Gi}^j(0, T_{Si-1}^j) + 1, \dots, N_{Gi}^j(0, T_{Si-1}^j) - 1$$

### 3.3 Calculation of communication delay

본 절에서는 3.2절에서 제시한 통신 지연 모델을 이용하여 각 스테이션에서 메시지의 발생 주기가 다를 때,  $T_{TR}$ 에 따른 통신 지연을 계산하였다. 본 절에서는 계산의 단순화를 위하여 4대의 스테이션만을 사용하였으며, 각 스테이션에서는 주기적 메시지만이 전송된다고 가정하였다. 또한, 각 스테이션들은 통신이 시작되면 즉시 메시지가 발생된다고 가정하였다.

그림 2는 각 스테이션에서의 토큰 수신 시간, 토

$T_S$	$T_E$
$N_Q + N_G$	$N_T$

그림 2. Calculation form of Communication delay

큰 송신 시간, 전송할 수 있는 메시지의 수를 계산하기 위한 모델을 나타내고 있다.

또한, 스테이션 1과 3에서는 2000 bit time 주기로 메시지가 발생되며, 스테이션 2와 4에서는 500 bit time 주기로 메시지가 발생된다. 여기에서, 1 bit time 은 Profibus-FMS의 전송 속도가 500Kbps인 경우, 2  $\mu$  sec로 계산된다.

이상과 같은 조건에서,  $T_{TR}$ 을 3000 bit time으로 설정하여 통신 지연을 계산하면 표 2와 같다. 표에서 스테이션 2와 4의 경우, 메시지 발생량이 많기 때문에, 시간이 지날수록 큐에 메시지가 쌓이게 된다. 이러한 현상은 메시지의 통신 지연을 길게 만든다.

특히, 각 스테이션마다 네트워크 관리자에 의하여 요구 통신 지연 (desired communication delay), 즉, 각 스테이션마다 허용될 수 있는 통신 지연이 다른 경우에서, 특정 스테이션에서 통신 지연이 요구 통신 지연보다 증가한다면, 다른 스테이션에서의 통신 지연을 조금 증가시키더라도 그 스테이션에서의 통신 지연을 감소시키는 전략이 필요하다.

즉, 표 2에서 메시지 사이클에 소요되는 시간 ( $T_{\text{cycle}}$ 을 200 bit time, 토큰 전송에 소요되는 시간 ( $T_{\text{token}}$ 을 100 bit time이라고 가정하여 각 스테이션에서의 통신 지연은 다음과 같다.

station 1 :

200,400,1800,400,400,600

station 2 :

500,1200,900,600,1900,1600,1300,1000,2100,1800,  
1500,1200,2700,2400,3700,3300,3000,2700,2400,

여기에서, 10,000 bit time이 지난 후에, 스테이션 1의 최대값은 1800 bit time이나, 스테이션 2의 최대값은 3700 bit time으로 크게 증가되었으며, 계속적으로 큐에 저장됨으로서 통신 지연이 급속하게 증가됨을 알 수 있다.

이를 표 3과 같이  $T_{TR}$ 을 스테이션 1과 3에서는 3000 bit time으로 유지하고, 스테이션 2와 4에서는 2000 bit time으로 조절하였다. 이러한 경우 전체적으

로 큐에 저장되는 현상이 현저하게 줄어들었다. 또한, 각 스테이션에서의 통신 지연은 다음과 같다.

station 1 :

200,1600,2000,2800,1000,1800

station 2 :

500,1200,900,600,1900,1600,1300,1000,2300,2000,  
1700,1400,1100,2800,2500,2200,1900,1600,2600

이상과 같이 스테이션 1 및 2의 통신 지연이 3000 bit time 이상으로 증가하지는 않으며, 큐에서 거의 저장되지 않는 것으로 나타났다. 또한,  $T_{TR}$ 을 최적값으로 조절하는 경우, 최대 통신 지연이 최소가 되도록 설정할 수 있다. 특히, 네트워크 관리자의 요구 통신 지연이 3000 bit time이라고 가정하였다면, 표 2에서는 만족을 시키지 못하였지만, 표 3에서는 만족을 시키는 결과를 낳게 된다.

#### 4. 결론

본 논문에서는 산업용 네트워크인 Profibus-FMS에서의 통신 지연을 수학적으로 계산할 수 있는 통신 지연 모델을 제안하였으며,  $T_{TR}$ 에 따라 통신 지연을 네트워크 관리자가 요구하는 요구 통신 지연 이하로 맞출 수 있는 가능성을 보였다.

향후에는 개발된 통신 지연 모델이 실험 모델에서의 통신 지연과 유사한 패턴을 가지는가를 평가하여야 한다. 또한, 통신 지연을 최소화시킬 수 있는  $T_{TR}$  선정 방법에 대하여 연구를 진행하여야 한다.

#### 참고문헌

1. 홍승호, 김기암, 김지용, 고성준, "분산제어 및 자동화시스템과 필드버스", 제어·자동화·시스템 공학회지, 2권 4호, pp19-29, 1996.7.
2. PROFIBUS Specification - Normative Parts of PROFIBUS-FMS, -DP, -PA according to the European Standard, EN 500 170, vol.2, March, 1998

Table 2. Communication delay with same TTR

s \ i	station 1 $T_{TR} = 3000$ $T_M = 2000$		station 2 $T_{TR} = 3000$ $T_M = 500$		station 3 $T_{TR} = 3000$ $T_M = 2000$		station 4 $T_{TR} = 3000$ $T_M = 500$	
	1	0	200	300	500	600	800	900
	0+1	1	0+1	1	0+1	1	0+2	2
2	1400	1400	1500	2100	2200	2400	2500	3300
	0+0	0	0+3	3	0+1	1	0+4	4
3	3400	3600	3700	4500	4600	4800	4900	5500
	0+1	1	0+4	4	0+1	1	0+4	3
4	5600	5800	5900	6700	6800	7000	7100	7900
	0+1	1	0+4	4	0+1	1	1+5	4
5	8000	8400	8500	8900	9000	9200	9300	10100
	0+2	2	0+6	2	0+1	1	2+4	4
6	10200	10400	10500	11500	11600	11800	11900	12300
	0+1	1	4+4	5	0+1	1	2+5	2

Table 3. Communication delay with different TTR

s \ i	station 1 $T_{TR} = 3000$ $T_M = 2000$		station 2 $T_{TR} = 4000$ $T_M = 500$		station 3 $T_{TR} = 3000$ $T_M = 2000$		station 4 $T_{TR} = 4000$ $T_M = 500$	
	1	0	200	300	500	600	800	900
	0+1	1	0+1	1	0+1	1	0+2	2
2	1400	1400	1500	2100	2200	2400	2500	3300
	0+0	0	0+3	3	0+1	1	0+4	4
3	3400	3600	3700	4500	4600	4800	4900	5700
	0+1	1	0+4	4	0+1	1	0+4	4
4	5800	6000	6100	7100	7200	7400	7500	8500
	0+1	1	0+5	5	0+1	1	0+5	5
5	8600	9000	9100	10100	10200	10400	10500	11500
	0+2	2	0+6	5	0+2	1	0+6	5
6	11600	11800	11900	13100	13200	13400	13500	14500
	0+1	1	1+5	6	1+1	1	1+6	5