

Profibus 성능 모델에서 최대 허용 전송 지연을 만족할 수 있는 타이머 선정에 관한 연구

김현희*(부산대 대학원 지능기계공학과), 이경창(부산대 기계공학부 연구원),
이석(부산대 기계공학부)

Timer Selection for Satisfying the Maximum Allowable Delay using Performance Model of Profibus Token Passing Protocol

H.H.Kim(Dept. of Int. & Mecha. Eng., PNU), K.C.Lee (School of Mecha. Eng., PNU),
S.Lee (Mecha. Eng., PNU)

ABSTRACT

Recently, the fieldbus becomes an indispensable component for many automated systems. In the fieldbus system, real-time data containing sensor values and control commands has a tendency to rapidly lose its value as time elapses after its creation. In order to deliver these data in time, the fieldbus network should be designed to have short delay compared to the maximum allowable delay. Because the communication delay is affected by performance parameters such as target rotation timer of token passing protocol, it is necessary to select proper parameter settings to satisfy the real-time requirement for communication delay. This paper presents the timer selection method for Profibus token passing networks using genetic algorithm (GA) to meet the delay requirements.

Key Words : Profibus token passing protocol(프로피버스 토큰 패싱 프로토콜), fieldbus, performance model(성능모델), real-time requirements(실시간 요구조건), protocol parameter(프로토콜 매개변수), genetic algorithm(유전 알고리즘), real value coding

1. 서론

본 논문에서는 Profibus-FMS 토큰 패싱 프로토콜의 성능을 확인하기 위한 하나의 방법으로써, 성능 변수에 따라 전송 지연을 수학적으로 계산할 수 있는 성능 모델을 제시하였다. 그리고, GA 를 이용하여 타이머 선정에 관한 알고리즘을 제시하였다. 성능 모델과 실험 모델의 결과와 비교하여 성능 모델의 유효성을 검토하였고, GA 를 이용한 타이머 선정을 이용하여 메시지 전송 지연을 줄이고자 했다. 특히, 본 논문에서 제안된 성능 모델은 실제 시스템에서 사용되는 트래픽 조건을 입력으로 하여 전송 지연을 빠른 시간 내에 계산 할 수 있기 때문에, 실제 시스템의 시운전 없이도 적절한 성능 변수를 선정할 수 있는 유효한 도구가 될 것이다.

본 논문은 6 장으로 구성되어 있다. 2 장에서는 Profibus-FMS 의 성능 모델을 제시한다. 3 장에서는 GA 에 의한 타이머 선정에 관한 알고리즘, 4 장에서는 성능모델과 실험 모델을 비교하고, GA 로 선정

된 타이머의 유효성을 검증한다. 마지막으로, 5 장에서는 요약과 향후 과제에 대해서 서술한다.

2. Profibus 토큰 패싱 프로토콜에서의 성능 모델

본 장에서 제안된 Profibus-FMS 토큰 패싱 프로토콜의 성능 모델은 다음과 같은 가정을 바탕으로 하고 있다. 첫째, 네트워크 인터페이스 카드(network interface card, NIC)의 송신 큐를 충분히 크게 함으로써, 각 스테이션에서 생성되는 메시지는 취소(overflow)되지 않고 모두 송신 큐(queue)에 저장될 수 있다고 가정한다. 둘째, 전송 중 에러는 발생하지 않는다고 가정한다. 셋째, GAP 갱신 시간(gap update time)을 충분히 크게 설정함으로써, GAP 갱신을 위한 통신과 스테이션의 출입은 발생하지 않는다고 가정한다.

$$Nh_{0i}^j = Nh_{0i}^j(0, T_{i1}^j) - \sum_{l=1}^{i-1} Nh_{il}^j \quad (1)$$

$$N_{Q_i}^j = N_{G_i}^j(0, T_{E_i}^j) - \sum_{l=1}^{i-1} N_{T_l}^j$$

이러한 가정을 바탕으로 하여, j 번째 스테이션의 i 번째 순회에서 토큰을 수신한 시점에 송신 큐에 저장되어 있는 높은 우선 순위 메시지의 수($N_{H_{O_i}}^j$)와 낮은 우선 순위 메시지의 수($N_{L_{O_i}}^j$)는 식(1)에 나타낸 바와 같이 i 순회에서 $T_{E_i}^j$ 시간까지 생성된 메시지의 총 수에서 i-1 순회까지 전송된 메시지의 총수를 뺀 값과 같다.

$$N_{A_i}^j = \begin{cases} 1 & \text{if } G\left(\frac{T_{H_i}^j}{Th_c^j}\right) \leq 0 \\ G\left(\frac{T_{H_i}^j}{Th_c^j}\right) & \text{if } 0 < G\left(\frac{T_{H_i}^j}{Th_c^j}\right) \leq N_{H_{O_i}}^j \\ N_{H_{O_i}}^j + G\left(\frac{T_{H_i}^j - (N_{H_{O_i}}^j \times Th_c^j)}{Th_c^j}\right) & \text{otherwise} \end{cases} \quad (2)$$

식에서, $G(x)$ 는 이산 단극 램프함수(discrete unipolar ramp function)로서 x 가 1보다 작은 경우에는 1이며, x 가 1보다 클 때는 소수 첫째자리에서 올림한 정수 값으로 정의된다. Th_c^j 는 j 번째 스테이션에서 높은 우선 순위 메시지의 송신에 소요되는 시간이고 Th_l^j 는 낮은 우선 순위 메시지의 송신에 소요되는 시간으로써, 메시지 송신 시간은 스테이션의 성능과 메시지 전송에 사용되는 통신 서비스에 따라 결정된다.

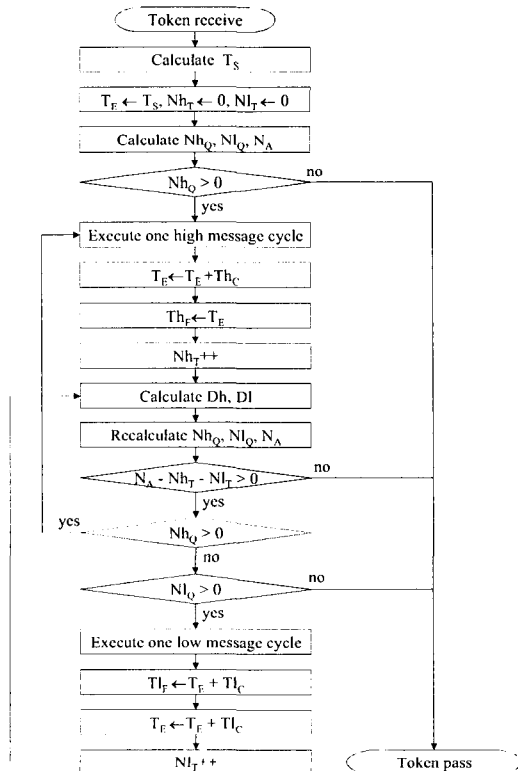


Fig.1 Flowchart of performance model for calculating the communication delay

$$(N_{H_{O_i}}^j, N_{T_i}^j) = \begin{cases} (N_{A_i}^j, 0) & \text{if } N_{A_i}^j \leq N_{H_{O_i}}^j \\ (N_{H_{O_i}}^j, \min(N_{T_i}^j, N_{A_i}^j - N_{H_{O_i}}^j)) & \text{otherwise} \end{cases} \quad (3)$$

결과적으로, 식 (1)과 식 (2)로부터 j 번째 스테이션의 i 번째 순회에서 전송되는 높은 우선 순위의 메시지 수($N_{H_{T_i}}^j$)와 낮은 우선 순위의 메시지 수($N_{L_{T_i}}^j$)는 다음과 같이 계산된다.

3. GA에 의한 타이머 선정

이 장에서는 GA를 이용하여 Profibus 토큰 패싱 프로토콜에서의 네트워크 관리자에 의해서 선정된 최대 허용 전송지연을 만족하는 타이머 선정 알고리즘에 관해서 소개한다.

일반적으로, 각 스테이션의 최대 허용 전송 지연은 다르게 설정된다. 왜냐하면, 한 스테이션이 다른 어떤 스테이션보다 보다 엄격한 실시간 요구 조건을 만족해야 하기 때문이다. 그러므로 각 스테이션은 T_{TR} 을 만족시키는 서로 다른 메시지 전송 한계 값을 가진다.[1]

유전 알고리즘(Genetic Algorithm)은 1962년 Holland에 의하여 소개된 적자생존의 법칙에 기반을 둔 최적화 이론으로서, 최적화 문제나 PID 튜닝과 같은 분야에서 효과적인 것으로 알려져 있다.[2][3]

그림 1은 최대 허용 전송 지연을 만족하는 T_{TR} 선정을 위한 단순 유전 알고리즘 구조를 보여준다. 우선, 랜덤하게 생성된 초기 해집단(initial population)을 구성(T_{TR_i})하였다. 각 T_{TR} 을 이용하여 성능 모델에서 계산된 전송 지연(d_i), T_{TR_i} 의 개체값은 선택(selection)되고 그들의 적합도(fitness)에 따라 재생성된다. 이때 재생성된 T_{TR} 들은 교차, 변이를 거쳐 새로운 해집단을 형성한다. 이 사이클은 목적값에 다달할때까지 수행한다.

3.1 적합도 함수(Fitness function)

GA에서 적합도 함수는 스트링(string)의 우수성을 평가하기 위한 척도로서, 적자생존의 법칙에 따라 보다 높은 적합도를 가진 스트링이 더 많은 자손을 생성할 수 있는 확률을 가지게 된다.

성능모델에서, 네트워크 내의 각 스테이션은 자신의 최대 허용 전송 지연을 만족해야 한다. 본 논문에서는 가중 합산 접근법(weighted sum approach)을 수정하여 다음과 같이 정의 하였다. 그림 2는 다중 합산 접근법을 스케치로 나타낸 것이다.

$$f = \sum_{i=1}^m w_i (1 - \sigma) f_i \quad (4)$$

$$\text{where } f = 1 - \frac{\sum_{k=0}^n e_i(k)}{n}$$

$$e_i(k) = \begin{cases} 1 & \text{if } d_i(k) > d_i^d \\ 0 & \text{otherwise} \end{cases}$$

$$\sum_{i=1}^m w_i = 1$$

식 (4)에서, 전송 지연 측정을 위한 메시지 수는 n , 스테이션의 수는 m 이다. 각 스테이션의 적합도 함수(f_i)는 1로 표준화 되었고, $e_i(k)$ 는 i -th 스테이션의 k -th 메시지의 에러 함수이다. 마지막으로 σ 는 f_i 의 표준 편차이고, w_i 는 네트워크 관리자가 설정한 가중치이다.

3.2 유전 연산자

본 논문에서는 선택(selection)방법으로 stochastic remainder technique 을 사용하였으며, 다수의 스트림의 적합도가 유사한 경우 부모와 동일한 비율의 자손들이 생성되는 것을 막기 위하여, 아래 (5)와 같은 선형 스케일링(linear scaling)을 사용하였다.

$$f' = pf + q$$

$$\text{where } c = \frac{f_{\max} - f_{\min}}{f_{\text{avg}} - f_{\min}}$$

$$p = \frac{(c-1)f_{\text{avg}}}{f_{\max} - f_{\text{avg}}}$$

$$q = \frac{f_{\text{avg}}(f_{\max} - cf_{\text{avg}})}{f_{\max} - f_{\text{avg}}} \quad (5)$$

식에서 f 는 적합도 함수이며, f' 은 스케일링된 적합도 함수이다. 또한 f_{\max} 는 적합도 함수의 최대 값이며, f_{\min} 은 최소값, f_{avg} 는 평균값이다. 그리고, 더 빠른 진화를 위하여, 현 세대 최상의 개체가 다음 세대까지 살아 남을 수 있는 엘리트스트(elitist model) 모델을 사용하였다. 산술적 교배(arithmetic crossover)는 동일 위치에 있는 두 부모의 유전자 값을 평균 내어 자식의 동일 위치 유전자 값으로 삼는 방법이다. 여기서 산술적 교배를 수정하여 (3)과 같은 식으로 표현하였다.

$$x_i' = \alpha(x_{i,1} - \frac{x_{i,2}}{2}) + (1-\alpha)(x_{i,2} + \frac{x_{i,1}}{2}) \quad (6)$$

x_i' 는 i -th 생성된 자식 개체이고, $x_{i,1}$ 와 $x_{i,2}$ 는 i -th 생성된 두개의 부모 개체이다. 그리고, α 는 균일 랜덤 변수 $[0, 1]$ 이다. 그리고 이 교배 연산자

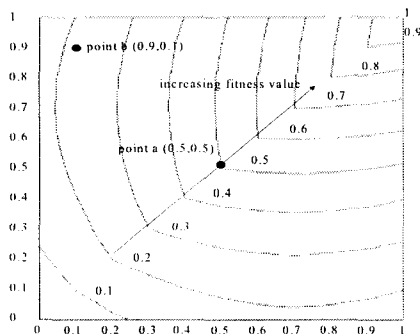


Fig. 2 Lines of equal fitness value with the modified weighted sum approach

는 일반적으로, 0.6 과 0.9 사이의 교배 확률 (P_c) 을 가지는 랜덤 방식을 적용하였다.

돌연변이 연산자는 균일 변이를 이용하였다.

$$x_i' = \begin{cases} x_i + \Delta(x_{\max}) & \text{if } v \leq 0.5 \\ x_i - \Delta(x_{\min}) & \text{otherwise} \end{cases} \quad (7)$$

x_i' 는 자식 개체, x_i 는 부모 개체, x_{\max} 는 최고의 적합도 값을 가지는 개체, 그리고 x_{\min} 는 최소의 적합도 값을 가지는 개체이다. 또한 v 는 균일 랜덤 변수 $[0, 1]$, $\Delta(y)$ 는 $[0, y]$ 범위에서 되돌려지는 값이다. 교배 연산자처럼, 변이는 P_m , 확률 값을 가지나, 매우 적은 값인 0.001 와 0.01 사이의 값이다.

4. 시뮬레이션 결과와 실험모델의 결과

본 절에서는 본 연구의 알고리즘의 유효성을 검증하기 위하여, 성능 모델과 타이머 선정 알고리즘을 Visual C++을 이용하여 프로그램 하였다. 이 프로그램은 전송 속도가 500Kbps 를 가지는 4 대의 스테이션으로 구성된 Profibus-FMS 네트워크를 고려하여 작성되었다.

전송 속도가 500Kbps 인 경우, 1 비트 타임은 2 μsec 로 계산된다. T_C^j 는 1,750 비트 타임, T_T 는 500 비트 타임, Th_D^j 와 Tl_D^j 4,000 비트 타임이다. 이는 Pentium III (500MHz)급의 PC 에서 40 바이트의 메시지를 Information Report 서비스로 송신한 경우에 측정된 값이다. 스테이션 1 과 3 의 높은 우선순위 메시지는 균일 분산으로 7500 비트 타임과 12,500 비트 타임 사이의 값으로 생성된다. 낮은 우선순위 메시지는 일정한 25,000 비트 타임 마다 생성된다. 스테이션 2 와 4 의 높은 우선순위 메시지는 균일 분산으로 17,500 비트 타임과 22,500 비트 타임 사이의 값으로 생성되고, 낮은 우선순위 메시지는 스테이션 1,3 과 동일하게 하다. 스테이션의 최대 허용 지연은 1 과 3 스테이션에서는 10,000 비트 타임, 2 와 4 스테이션에서는 20,000 비트 타임으로 나타났다.

그림 3 은 스테이션 1 의 높은 우선순위 메시지 500 개에 대한 실험 모델의 결과를 비교하여 나타낸 것이다. 그림 3(a)는 제시된 표준 값으로 T_{TR} 을 설정한것으로 최대 전송 지연은 20,000 비트 타임, 평균 전송 지연은 9,024 비트타임, 그리고 표준 편차는 3,040 비트 타임으로 나타났다. 그림 3(b)는 조정된 값으로서 최대 전송 지연은 15,000 비트 타임, 평균 전송 지연은 8,202 비트 타임, 표준 편차는 1,820 비트 타임으로 나타났다.

표 1 은 성능 모델의 실험 결과를, 표 2 는 그림 3(a)와 3(b)를 비교한것이다.

5. 결론

본 논문은 실시간 요구 조건을 만족하는 Profibus 토크 패싱 프로토콜 성능 모델을 GA 에 기반한 타이머 선정 알고리즘을 나타내었다. 이 논문에서, Profibus-FMS 프로토콜 성능 모델을 개발 하였으며, 전송 지연을 계산 할 수 있는 알고리즘을 제시하였다. 또한, 이 모델은 적합도 함수를 평가하기 위한 모델로 사용하였고, GA 에 기반한 타이머 선정 알고리즘은 메시지 지연을 줄이기 위하여 개발되었다. 결과적으로, 타이머 선정 알고리즘은 성능 모델 시뮬레이션과 Profibus-FMS 테스트 베드의 실험들 다에서 그 유효성이 증명 되었다.

따라서, 본 논문에서 제안된 성능 모델은 네트워크 관리자가 경험적 방법에 의하여 성능 변수를 선정하는데 있어서 유효한 도구가 될 것으로 평가된다. 게다가, 타이머 선정 알고리즘은 최대 허용 전송 지연과 트래픽이 주어졌을 때 매우 유효하게 T_{TR} 의 값을 선정 할 수 있는 도구로 증명되었다. 이러한 결과로 네트워크 관리자는 선정한 성능 변수를 직접 실제 네트워크 시스템에 적용하기에 앞서, 성능 모델을 통하여 충분한 성능 평가를 거친 후 적용할 수 있게 됨으로써, 보다 안정되고 최적화된 네트워크 타이머를 선정 할 수 있을 것이다.

향후과제로는 온라인(on-line)상에서 시간에 따라 변하는 트래픽 값에 따라 조절 가능한 타이머 선정에 관한 연구가 이루어져야 할 것이다.

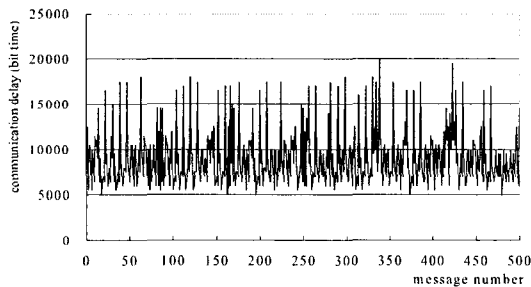


Fig. 3(a) communication delay of high messages from station 1 with recommended T_{TR}

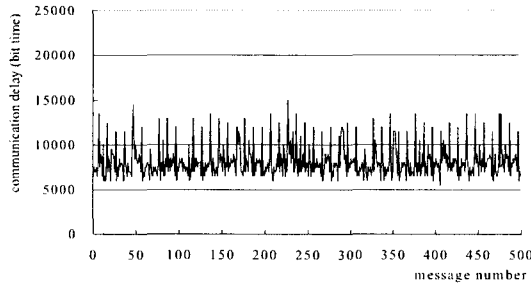


Fig. 3(b) communication delay of high messages from station 1 with adjusted T_{TR}

참고문헌

1. M.Y. Chow and Y. Tipsuwan, "Network-based control systems: a tutorial," The 27th Annual Conference of the IEEE Industrial Electronics Society, Denver, Colorado, 2001, pp. 1593-1602
2. J.F. Whidborne and R.S.H. Istepanian, "Genetic algorithm approach to designing finite-precision controller structures," IEE Proceedings - Control Theory and Applications, vol. 148, no. 5, pp. 377-382, 2001
3. D.A. Scott, C.L. Karr, and D.E. Schinstock, "Genetic algorithm frequency-domain optimization of an anti-resonant electromechanical controller," Engineering Applications of Artificial Intelligence, vol. 12, no. 2, pp.

Table 1. Comparison of communication delays calculated by performance model from station 1

	with recommended timer	with adjusted timer	Improvement ratio
maximum communication delay	20,700	17,375	16.1%
average communication delay	10,662	8,593	19.4%
standard deviation communication delay	3,545	2,626	25.9%
the number of messages delivered within the maximum allowable delay	222	341	53.6%

Table 2. Comparison of communication delays measured on experimental testbed from station 1 (bit-time)

	with recommended timer	with adjusted timer	Improvement ratio
maximum communication delay	20,000	15,000	25.0%
average communication delay	9,024	8,202	9.1%
standard deviation communication delay	3,040	1,820	40.1%
the number of messages delivered within the maximum allowable delay	379	432	14.0%