Proceedings of the 12th
KACC, October 1997

제한된 버퍼를 가진 IEEE 802.4 토큰 버스 네트워크의 성능 해석

Analytic Performance Evaluation of the IEEE 802.4 Token Bus Network with Finite Buffer Capacity

°문상용, 박홍성*, 권욱현

서울대학교 전기공학부 (Tel: +82-2-873-2279; Fax: +82-2-878-8933; E-mail: sangyong@isltg.snu.ac.kr)
*강원대학교 제어계측공학과 (Tel: +82-361-250-6346; Fax: +82-361-242-2059 E-mail: hspark@cc.kangwon.ac.kr)

Abstract This paper analytically derives the variance of service time of a station in the symmetric IEEE 802.4 token bus network with single access class and finite buffer capacity. This performance measure is represented in terms of the total number of stations, the token hold time, the arrival rate of frames, the service rate, and other parameters. Using computer simulations, presented performance measure is validated.

Keywords IEEE 802.4 Token Bus Network, Petri-net, Variance

1. 서론

최근에 통신 네트워크는 자동화를 위한 매우 중요한 부분중 하나가 되었다. 그래서 여러 회사의 자동화 기기들 사이의 통신 문제를 해결하기 위해 MAP(Manufacturing Automation Protocol)과 같은 통신 표준 규약이 제시되었다.

IEEE 802.4 토큰 버스 네트워크 규약(Token Bus Mechanism)[1]은 MAP에서 사용된다. 이 규약은 개념적으로 논리적 토큰 링 네트워크로 생각될 수 있다. 토큰을 가지는 스테이션만이 프레임을 전송할 수 있는 것이다. 따라서 IEEE 802.4 토큰 버스 네트워크에서는 토큰이 모든 스테이션을 한 번 회전하는데 걸리는 시간과 각 스테이션이 토큰을 가지고 데이터를 전송하는 시간이 성능과 관련된 중요한 요소라고 할 수 있다.

지금까지 토큰 버스 규약에 관한 많은 연구들이 있다. 토큰 보유 시간(Token Hold Time)이 지난 후에는 전송하지 않고 많은 부하가 걸린다는 가정하에 IEEE 802.4 토큰 버스 네트워크의 처리량(throughput)이 [2]에서 분석되었다.

몇몇 연구들[3][4][5]은 요구되는 처리량을 만족시키기 위해 각 타이머들에 어떤 값을 할당해야 하는 것에 관한 것들이다. 어떤 논문들은 성능 지표들과 관련된 값들의 범위를 유도한다. IEEE 802.4 토큰 버스 규약의 평균 토큰 회전 시간(Mean Token Rotation Time) 및 최대 토큰 회전 시간(Maximum Token Rotation Time)에 대한 상한이 [6]에서 유도되었다. 그리고 [7]에서는 평균 전송 시간(Mean Transmission Time) 및 평균 토큰 회전 시간에 대한 상한과 하한이 유도되었다. 참고문헌 [8]에서는 전송중에는 프레임이 발생되지 않는다는 가정하에 평균 토큰 회전 시간 및 평균 대기 시간(Mean Waiting Time)을 구하는 알고리즘이 제시되었다. 최근에 확률적 페트리네트를 이용하여 제한된 버퍼를 가지는 IEEE 802.4 토큰 버스 네트워크의 성능이 완전 소비 전송의 가정하에 해석되었다[9]. 이 연구에서는 평균 토큰 회전 시간과 한 스테이션의 평균 서비스 시간이 유도되었다. 그런데 때로 통신망의 성능 해석에서 평균만을 가지고는 불충분한 경우가 있을 수 있다. 평균 뿐만 아니라 분산을 구할 수 있다면 대상 통신망의 평균적 성능을 얻는데 큰 도움을 줄 수 있다.

본 연구에서는 IEEE 802.4 토큰 버스 네트워크의 한 스테이션의 서비스 시간의 분산을 구한다. 확률적 페트리네트 모델[9]을 이용하여 성능 지표를 유도한다. 역시 완전 소비 전송의 가정을 사용한다. 그리고 모멘트 발생 함수를 이용하는 해석 방법[10]을 사용하여 성능 지표를 유도한다. 먼저 성능 지표를 유도하기 위한 식을 제시한다. 그리고 큐의 크기가 1인 경우에 성능 지표를 구하는 것이 예로 제시된다. 그리고 토큰 보유 시간이 충분하다는 가정하에 큐의 크기를 K로 놓고 성능 지표를 구한다. 마지막으로 결과가 유효하다는 것을 보여주기 위해 시뮬레이션 결과와 비교한다.

2. 페트리네트 모델과 해석 방법

이 논문에서 사용하는 그림 1의 페트리네트 모델은 참고문헌 [9]에서 제시된 것이다. 각 플레이스(place) 및 트랜지션(transition)이 나타내는 의미는 표 에 나타나 있다[9]. 이 모델에서 프레임의 발생간의 시간(interarrival time)과 서비스 시간(service time)은 각각 매개변수 λ, μ의 익스포넨셜(exponential) 분포를 가진다고 가정된다.

이 논문에서 사용한 해석 방법[10]을 간단하게 소개한다. 이 방법은 모멘트 발생 함수(Moment Generating Function)를 사용한다. 모멘트 발생 함수 $M(s)$는 다음과 같이 정의된다.

$$M(s) = \int_{-\infty}^{\infty} e^{st} f(t) dt.$$

이 방법에서 사용하는 전달 함수(transfer function)는 각 트랜지션에 대해 정의되며 다음과 같다.

$$W_k(s) = P(k)M(s)$$

여기에서 $P(k)$는 트랜지션 t_k가 파이어(fire)할 확률이다.

이 방법을 이용하여 시스템을 해석하고자 할 경우에는 다음의 단계를 따라야 한다.

1) 시스템을 페트리네트를 이용하여 모델링한다.
2) 그 페트리네트에 대한 도달가능성 그래프(reachability graph)를 만든다.
3) 그 그래프로부터 상태 머신(state machine)을 유도한다.
4) 그 상태 머신의 각 트랜지션들에 대한 전달 함수들을 구한다. 다음과 같은 트랜지션들의 집합 $\{t_{n1}, t_{n2}, \cdots, t_{nd}\}$, $d > 1$이 동시에

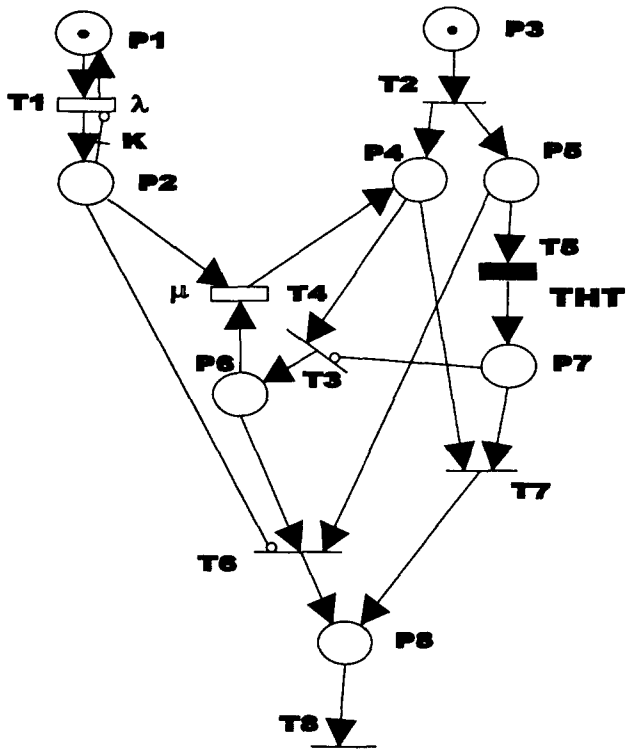

그림 1. 한 스테이션의 페트리네트 모델
Fig. 1. Petri-net model of a station.

표 1. 페트리네트 모델 설명
Table 1. Description of the Petri-net model.

이름	설명
P1	프레임이 발생되었다
P2	발생된 프레임이 큐에 들어왔다
P3	토큰이 도착하였다
P4	토큰이 프레임 전송에 사용될 것이다
P5	토큰 보유 타이머 계산이 시작되었다
P6	프레임이 전송된다
P7	토큰 보유 타이머의 시간이 지났다
P8	토큰이 다음 스테이션으로 전송된다
T1	프레임의 큐에의 도착
T2	토큰 보유 시간 및 전송의 시작
T3	전송 시작
T4	프레임의 전송 시간
T5	토큰 보유 시간
T6	전송할 프레임이 없음
T7	토큰 보유 타임의 소진으로 인한 전송 중지
T8	토큰의 전송

인에이블(enable)되고 X_{ni}가 t_{ni} $(i = 1, 2, \cdots, d)$의 임의의 시간 지연이라고 할 때 전달 함수 W_{ni}는 다음과 같다.

$$W_{ni} = P(t_{ni})M(t_{ni}, s) \tag{1}$$

$$P(t_{ni}) = P\{X_{ni} \leq X_{nj}, j \neq i\}$$

$M(t_{ni}, s)$는 $Min\{X_{nj}, j = 1, \cdots, d\}$의 모멘트 발생 함수이다.
5) 평균 통과 시간(Mean Passage Time), 평균 재도달 시간(Mean Recurrence Time) 등과 같은 성능 지표들에 대한 대응(equivalent) 전달 함수를 Mason의 법칙[11]을 사용하여 구한다.
6) 이들 대응 전달 함수를 사용하여 성능 지표의 값을 구한다. 예를 들어 플레이스 S_i에서 S_j으로의 통과를 나타내는 대응 전달 함수가 W_E라고 할 때 평균 통과 시간 및 그 분산은 다음과 같이 계산된다.

$$T_p = \frac{d}{ds}M_E(s)|_{s=0} = \frac{d}{ds}\frac{W_E(s)}{p_E}|_{s=0}$$

$$\begin{aligned} V_p &= \frac{d^2}{ds^2}M_E(s)|_{s=0} - (\frac{d}{ds}M_E(s)|_{s=0})^2 \\ &= \frac{d^2}{ds^2}\frac{W_E(s)}{p_E}|_{s=0} - (\frac{d}{ds}\frac{W_E(s)}{p_E}|_{s=0})^2. \end{aligned}$$

정의에 의해 $M_E(0)$의 값은 0이므로 p_E는 $W_E(0)$와 같다.

3. 문제 정립 및 해석 결과

여기에서는 평균 토큰 회전 시간 및 평균 서비스 시간을 구하기 위한 문제를 정립하고 토큰 보유 시간이 충분하다는 가정하의 해석 결과를 유도한다.

3.1 문제 정립

한 스테이션의 평균 서비스 시간 T_s는 다음의 식으로 표현된다.

$$T_s = \vec{P}_A^T \vec{T} \tag{2}$$

여기에서

$$\vec{P}_A = [\ pa_{\to 0}\ \ pa_{\to 1}\ \ \cdots\ \ pa_{\to K}\ ]^T$$

$$\vec{T} = [\ 0\ \ T_1\ \ \cdots\ \ T_K\ ]^T$$

K : 버퍼의 수(큐의 크기)

이다. 원소 $pa_{\to i}$ $(i = 0, 1, \cdots, K)$는 큐에 i개의 프레임이 있을 때 토큰이 도착할 확률이다. T_i $(i = 0, 1, \cdots, K)$는 그 경우의 서비스 시간을 나타낸다.

$\vec{P}_A$는 프레임의 발생률(arrival rate) 및 평균 토큰 도착간 시간(mean token intervisit time)에 따라 좌우된다. 도착간의 시간이라는 것은 토큰이 떠난 후 다시 돌아오는 때까지 경과된 시간을 말한다. 그러면 평균 토큰 도착간 시간은 상수 T_v로 놓을 수 있다. 그리고 $p[x(t) = y]$는 시간 t 동안에 y개의 프레임이 발생될 확률이라고 하자. 그러면 $\vec{P}_A$는 다음과 같이 된다.

$$\vec{P}_A = TM_{AD}\vec{P}_D \tag{3}$$

여기에서

$$TM_{AD} = \begin{bmatrix} p(X(T_v)=0) & 0 & \cdots & 0 \\ p(X(T_v)=1) & p(X(T_v)=0) & \cdots & 0 \\ p(X(T_v)=2) & p(X(T_v)=1) & \cdots & 0 \\ . & . & \cdots & . \\ . & . & \cdots & . \\ . & . & \cdots & . \\ p(X(T_v)\geq K) & p(X(T_v)\geq K-1) & \cdots & 1 \end{bmatrix}$$

$$\vec{P}_D = [\ pd_{0\rightarrow}\ \ pd_{1\rightarrow}\ \ \cdots\ \ pd_{(K-1)\rightarrow}\ \ 0\ ]^T$$

이다. 확률 $pd_{j\rightarrow}$는 큐에 j개의 프레임이 있을 때 토큰이 떠날 확률을 나타낸다. 프레임의 발생은 매개변수 λ의 포아송(Poisson) 분포를 따른다고 가정했으므로 $p(X(T_v) = M)$는 다음의 식

$$p(X(T_v) = M) \ = \ \frac{(\lambda T_v)^M e^{-\lambda T_v}}{M!}$$

에 의해 쉽게 얻을 수 있다. 그러므로 만약 T_v의 값을 구할 수 있다면 행렬 TM_{AD}는 쉽게 얻을 수 있다.

$\vec{P}_A$를 구하기 위해서는 $\vec{P}_D$를 알아야 하는데 $\vec{P}_D$는 $\vec{P}_A$의 함수로 다음과 같이 표현된다.

$$\vec{P}_D = TM_{DA}\vec{P}_A \tag{4}$$

TM_{DA}는 다음과 같다.

$$TM_{DA} = \begin{bmatrix} 1 & p_{10} & p_{20} & \cdots & p_{K0} \\ 0 & p_{11} & p_{21} & \cdots & p_{K1} \\ . & . & . & \cdots & . \\ . & . & . & \cdots & . \\ . & . & . & \cdots & . \\ 0 & p_{1,K-1} & p_{2,K-1} & \cdots & p_{K,K-1} \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

확률 p_{ij}는 큐에 i개의 프레임이 있을 때 토큰이 도착한 경우 j개의 프레임이 있을 때 떠날 확률을 나타낸다. 위의 식들에 따르면 결국 $\vec{P}_A$를 구하기 위해서는 T_v와 TM_{DA}를 알아야 한다는 것이 된다. 이 TM_{DA}를 바로 모멘트 발생 함수를 이용하는 해석 방법에 의해 구한다. 그러면 결국 $\vec{P}_A$는 평균 토큰 도착간 시간 T_v만 구한다면 식 (3), (4)과 다음의 조건

$$pa_{\rightarrow 0} \ + \ pa_{\rightarrow 1} \ + \ \cdots \ + \ pa_{\rightarrow K} \ = \ 1$$

에 의해 구할 수 있다.

T_{ij}는 큐에 i개의 프레임이 있을 때 토큰이 도착해서 j개의 프레임이 있을 때 떠나는 경우의 서비스 시간을 나타낸다고 하자. 이제 $\vec{T}$를 생각해보자. $\vec{T}$는 스테이션의 프레임 발생률, 프레임 서비스율, 그리고 토큰 보유 시간의 함수이다. $\vec{T}$의 원소 T_i는 다음과 같이 표현된다.

$$T_i = \sum_{j=0}^{K-1} p_{ij}T_{ij} \ (i = 1, \cdots, K)$$

여기에서 T_{ij} 역시 모멘트 발생 함수 해석 방법에 의해 구하게 된다. 그리고 이 T_{ij}를 구하는 과정에서 p_{ij} (즉, TM_{DA})는 거의 자동적으로 나오게 되므로 미리 p_{ij}를 구할 필요가 없다.

결과적으로 평균 토큰 도착간의 시간 T_v만 알면 $\vec{P}_A$, $\vec{T}$, 그리고 T_s는 구할 수 있게 된다. 이 T_v는 평균 토큰 회전 시간과 밀접한 관계가 있는 값이므로 평균 토큰 회전 시간을 표시하는 식에 의해 구하게 된다.

평균 서비스 시간 T_s는 T_v의 함수이므로 $T_s(T_v)$로 나타낼 수 있다. 그리고 평균 평균 토큰 전달 시간(mean switch-over time)을 T_o라고 하자. 이 T_o는 상수이다. 네트워크가 대칭적이라고 가정하였으므로 스테이션의 수를 N이라고 할 때 평균 토큰 회전 시간 T_r은 다음과 같이 쓸 수 있다.

$$T_r = N(T_s(T_v) + T_o) \tag{5}$$

평균 토큰 회전 시간은 평균 서비스 시간과 안정 상태의 도착간의 시간의 합이므로 평균 토큰 회전 시간은 다른 방법으로 표시될 수 있다.

$$T_r = T_s(T_v) + T_v \tag{6}$$

결국 식 (5)와 (6)을 사용하면 안정 상태 도착간의 시간 T_v를 구할 수 있다. 그러면 식 (2)와 (5)를 통해 평균 서비스 시간 T_s와 평균 토큰 회전 시간 T_r을 구할 수 있게 된다.

3.2 해석 결과

서론에서 언급했듯이 여기에서는 토큰 보유 시간이 충분하기 때문에 시간이 부족하여 전송을 하지 못하고 큐에 프레임이 있는 채로 토큰이 떠나는 경우가 없다는 가정하에 평균 토큰 회전 시간 및 평균 서비스 시간을 구한다. 큐의 크기는 K로 놓는다.

먼저 위의 가정을 사용하면 TM_{DA}는 다음과 같이 간단한 형태로 된다.

$$TM_{DA} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ . & . & . & \cdots & . & . \\ . & . & . & \cdots & . & . \\ . & . & . & \cdots & . & . \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

그러면 $TM_{AD}TM_{DA}$는

$$TM_{AD}TM_{DA} = \begin{bmatrix} p(X(t)=0) & \cdots & p(X(t)=0) \\ p(X(t)=1) & \cdots & p(X(t)=1) \\ . & \cdots & . \\ . & \cdots & . \\ . & \cdots & . \\ p(X(t)=K-1) & \cdots & p(X(t)=K-1) \\ p(X(t)\geq K) & \cdots & p(X(t)\geq K) \end{bmatrix}$$

과 같다. 그러면 다음의 식을 유도할 수 있다.

$$pa_{\rightarrow i} = \begin{cases} \frac{(\lambda t)^i e^{-\lambda t}}{i!}, & \text{i=0,1,}\cdots\text{,K-1} \\ 1 - \sum_{i=0}^{K-1} \frac{(\lambda t)^i e^{-\lambda t}}{i!}, & \text{i=K} \end{cases} \tag{7}$$

이제 $\vec{T}$를 구해야 한다. 참고문헌 [9]에 의하면 $\vec{T}$ 및 평균 서비스 시간 T_s는 다음의 형태로 된다.

$$T_i = \frac{\lambda^{K-1} + 2\lambda^{K-2}\mu + \cdots + i\lambda^{K-i}\mu^{i-1} + \cdots + i\mu^{K-1}}{\mu^K} \tag{8}$$

$$T_s = \sum_{i=1}^{K-1} \{ (\frac{(\lambda T_v)^i e^{-\lambda T_v}}{i!}) T_i \} + (1 - \sum_{j=0}^{K-1} \frac{(\lambda T_v)^j e^{-\lambda T_v}}{j!}) T_K \tag{9}$$

이제 토큰이 도착한 순간에 i개의 프레임이 큐에 저장되어 있는 경우 서비스 시간의 분산을 V_i하고 하자. 약간 복잡한 계산 과정을 통해 본 연구에서 다음과 같이 유도되었다.

$$\begin{aligned} V_i \ = \ & \frac{\sum_{j=1}^{i} j\lambda^{K-j}\mu^{j-1} + i\sum_{j=i+1}^{K} \lambda^{K-j}\mu^{j-1}}{\mu^K} \\ & \times \frac{\sum_{j=1}^{K} j\lambda^{K-j}\mu^{j-1} + \sum_{j=i+1}^{K} (j-i)\lambda^{K-j}\mu^{j-1}}{\mu^K} \\ & - \frac{2(\sum_{j=1}^{i}(\sum_{l=1}^{j} l)(K-j)\lambda^{K-1-j}\mu^{j-1}}{\mu^K} \\ & - \frac{\sum_{j=i+1}^{K-1}(\sum_{l=j-i+1}^{j} l)(K-j)\lambda^{K-1-j}\mu^{j-1})}{\mu^K} \end{aligned} \tag{10}$$

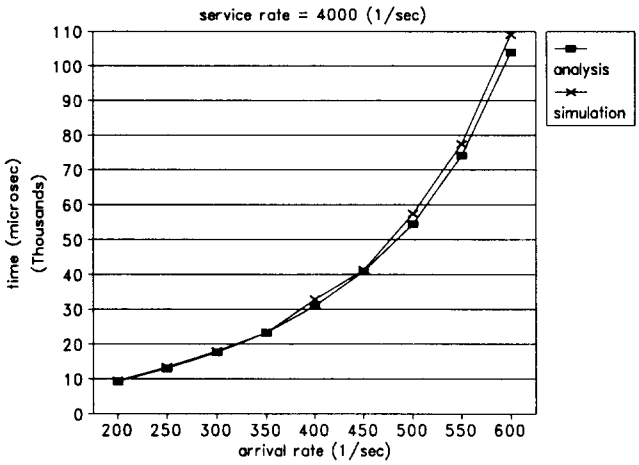

그림 2. 버퍼가 10개인 경우 결과
Fig. 2. Results for 10 buffer capacity.

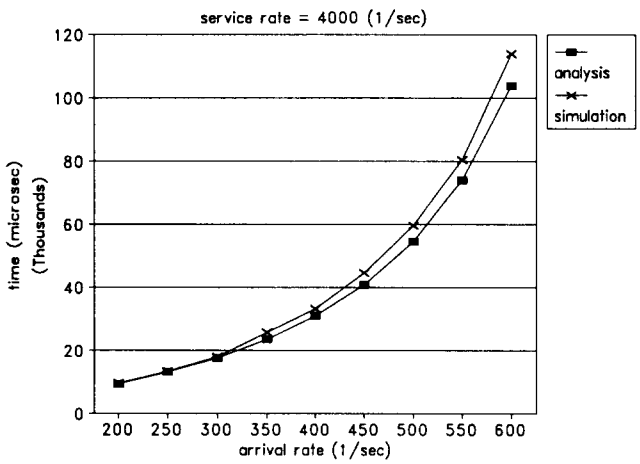

그림 3. 버퍼가 20개인 경우 결과
Fig. 2. Results for 20 buffer capacity.

위의 T_i와 V_i를 이용하면 하나의 스테이션의 평균 서비스 시간 및 그 분산을 구할 수 있다. V_i를 유도하는 과정은 매우 복잡하므로 이 논문에서는 생략한다.

4. 시뮬레이션 결과

여기에서는 시뮬레이션을 통하여 해석 결과가 정확하고 유효하다는 것을 보인다. 그림 2는 큐의 크기가 10인 경우에 대해 시뮬레이션과 해석 결과를 비교한 것이다. 그림 3은 큐의 크기가 20인 경우의 결과를 나타낸다. 그림에서 보면 쉽게 알 수 있듯이 최대 오차가 5 퍼센트 정도로 매우 작음을 알 수 있다. 그러나 본 연구에서는 완전 소비 전송 가정을 사용하였기 때문에 프레임의 도착율이 커질수록 오차가 커진다는 것을 알 수 있다.

5. 결론

이 논문에서는 하나의 접근 등급과 제한된 수의 버퍼를 가지는 대칭의 IEEE 802.4 토큰 버스 네트워크를 해석하였다. 프레임은 임의적으로 발생되며 전송된다고 가정되었다. IEEE 802.4 토큰 버스 네트워크에 대한 페트리네트 모델을 이용하여 한 스테이션의 전송시간의 분산을 유도하였다. 시뮬레이션을 통해 유도한 결과가 유효하다는 것을 보였다.

이 논문에서 제시된 해석은 필드버스, IEEE 802.3 등의 IEEE 802.4 토큰 버스 규약 외의 다른 네트워크 규약에도 쉽게 적용될 수 있다. 앞으로 이들 네트워크 규약에의 적용이 필요할 것으로 생각된다. 그리고 여러 개의 접근 등급을 사용하는 네트워크 및 비대칭 네트워크(네트워크에 연결된 스테이션들의 매개변수들이 다른 경우)에 대한 적용도 앞으로 연구되어야 할 과제이다.

참고문헌

[1] *Token Passing Bus Access Method and Physical Layer Specification*, ANSI/IEEE Standard 802.4, 1985.

[2] Joseph W. M. Pang and Fouad A. Tobagi, "Throughput Analysis of a Timer Controlled Token Passing Protocol Under Heavy Load," *IEEE Trans. on Communications*, Vol. 37, No. 7, pp.694-702, July 1989.

[3] Paolo Montuschi, Adriano Valenzano, Luigi Ciminiera, "Selection of Token Holding Times in Timed-Token Protocols," *IEEE Trans. on Industrial Electronics*, Vol. 37, No. 6, pp.442-451, December 1990.

[4] R. Mangala and Alfred C. Weaver, "Setting Target Rotation Times in an IEEE Token Bus Network," *IEEE Trans. on Industrial Electronics*, Vol. 35, No. 3, pp.366-371, August 1988.

[5] Anura P. Jayasumana and Geetha G. Jayasumana, "On the Use of the IEEE 802.4 Token Bus in Distributed Real-Time Control Systems," *IEEE Trans. on Industrial Electronics*, Vol. 36, No. 3, pp.391-397, August 1989.

[6] P. Montuschi, L. Ciminiera, A. Valenzano, "Time Characteristics of IEEE 802.4 Token Bus Protocol," *IEE Proceedings-E*, Vol. 139, No. 1, pp.81-87, January 1992.

[7] Hong Seong Park, Sang Chul Ahn, and Wook Hyun Kwon, "Performance and Parameter Region for Real-Time Use in IEEE 802.4 Token Bus Network," *IEEE Trans. on Industrial Electronics*, Vol. 40, No. 4, pp.412-420, August 1993.

[8] D. W. Kim, H. S. Park, and W. H. Kwon, "The Performance of a Timer-Controlled Token Passing Bus Mechanism with Finite Buffers in an Industrial Communication Network," *IEEE Trans. on Industrial Electronics*, Vol. 30, No. 4, pp.421 -427, August 1993.

[9] 문상용, 박홍성, 권욱현, "페트리 네트를 이용한 IEEE 802.4 토큰 버스 규약의 성능 해석," 1994 한국자동제어학술회의, pp.661-666, 1994.

[10] DianLong Guo, Frank DiCesare, and MengChu Zhou, "A Moment Generating Function Based Approach for Evaluating Extended Stochastic Petri Nets," *IEEE Trans. on Automatic Control*, Vol.38, No.2, February 1993.

[11] Benjamin C. Kuo, *Automatic Control Systems*, Prentice-Hall, 1987.