

대규모 토큰-패싱 네트워크의 점근적 성능분석 및 적응제어

○ 심 광현 임 종태
한국과학기술원 전기및전자공학과

Asymptotic Performance Analysis and Adaptive Control of Large-Scale Token-Passing Networks

Kwang-Hyun Shim Jong-Tae Lim
Department of Electrical Engineering
Korea Advanced Institute of Science and Technology

Abstract

The main purpose of the paper is to derive asymptotic formulae for performance characteristics(throughput, delay) of large-scale token-passing network with buffered stations and to optimize the buffer capacity with respect to the probability of data generation. we consider two versions of token-passing network: uniform and nonuniform token-passing time interval. All results obtained are supported by simulations.

I. 서론

Token-passing network은 생산시스템의 data communication을 위한 표준통신방식인 manufacturing automation protocol (MAP)의 기본구조로서 CSMA/CD에 비해 부하가 큰 경우 특히 성능이 우수하며 유한 시지연을 갖는 등 많은 장점이 있다[1], [2]. 이에 성능 분석을 위한 많은 연구가 진행되어 왔으나[3]-[5] 거의 단일 buffer 및 무한 buffer의 경우가 대부분이며 유한 buffer의 경우에는 모델의 복잡성때문에 해석이 어렵다. 이에 점근적 해석 기법[6]을 이용하여 유한 buffer를 갖는 token-passing network의 성능을 평가하고 buffer 용량 및 data 발생율이 성능특성에 미치는 영향을 분석하여 원하는 성능특성을 갖도록 하는 buffer 용량을 구하고자 한다. 이제 본 논문에서 다루는 시스템을 하기와 같이 가정한다.

(1) 전체 시스템은 $1 \ll M < \infty$ 개의 station들로 구성되며 모든 station은 각각 $1 \leq N < \infty$ 개의 data를 저장할 수 있는 용량의 buffer를 가진다.

(2) Data의 전송은 한 station에서 network상의 인접한 다음 station으로 이루어지며 한쪽 방향으로만 전송이 가능하다.

(3) Network상에는 station의 network 사용을 제어하는 control bit들의 모임인 token이 존재하게 된다. 이 token은 유일하며 계속해서 network상을 순회하게 되는데 일단 token을 받은 station은 buffer에 data가 있으면 token을 가지고 있으면서 buffer의 최하위 cell에 있는 data만을 내 보내고 난 다음 다시 token을 다음 station에 전달한다. 그러나 만약 buffer가 비어 있는 때에는 일정 시지연 후에 token을 다음 station에 전달한다.

(4) Token이 임의의 한 station을 떠나 network상의 다음 station에 도달하는데 걸리는 시간(token 전달 시간)은 모든 station에 대해 동일하다.

(5) Data 발생(generation)에 대한 확률 분포는 모든 station에 대해 동일하며 buffer에 이미 들어있는 data의 양 및 시간에 대해 무관하다.

(6) 일단 들어온 모든 data에 대해 그것이 전송되는데 걸리는 시간은 동일하며 data 전송 시간과 token 전달 시간의 합을 $1 < \tau \ll M$ 로 한다.

(7) 들어오는 data들은 하위 cell에서부터 순서적으로 차 올라가게 된다. 그리고 하나의 data가 완전히 나가면 그 즉시 상위 cell에 있는 data들이 빈 cell을 채우면서 모두 아래로 내려오게 된다. 한편 data가 나가고 있는 동안에는 위에서 data들이 내려올 수 없다. 그리고 buffer의 모든 cell이 차있는 경우 더 이상의 data는 들어올 수 없다.

II. Uniform token-passing time interval의 경우의 성능 해석

이 경우는 token이 각각의 station에 일정시간 동안 머물러 있게 되는데 이것은 TDMA(time division multiple access)와 유사한 구조를 갖는다.

1. 모델링

다음과 같은 상태변수들을 정의한다.

$$\circ B_m^n(kT) \equiv B_m^n(k), \quad k = 0, 1, 2, \dots$$

여기서, n : Buffer's cell Index($BI = 1, 2, \dots, N$)
 m : Station Index($SI = 0, 1, \dots, M - 1$)
 T : Token 전달 시간 + Data 전송 시간

이 상태변수는 시간 kT 에서 SI (Station Index)가 m 인 station에 있는 buffer의 n 번째 cell($BI = n$)에 data의 유무를 나타내며 1 또는 0의 값을 갖는다. 그리고 이산화된 각각의 시간에 대해 token을

가지는 station의 SI 를 0으로한다. 그리고 token이 인접한 station으로 계속해서 전해질 때마다 모든 station의 SI 는 $M-1$ 이 다시 0이 되는 것을 제외하고는 모두 1씩 증가한다.

$$\circ A_m(kT) \equiv A_m(k), \quad k = 0, 1, 2, \dots$$

이 상태변수는 $[k, k+1]$ 시간 동안 SI 가 m 인 station에서의 data의 발생을 나타낸다. T 시간 동안 2개 이상의 data는 발생할 수 없으며 일단 발생된 data는 buffer에 비어있는 cell이 있을 때에만 buffer로 들어올 수 있다. 정의한 상태변수들을 사용하여 시스템을 모델링하면 다음과 같은 상태방정식을 얻을 수 있는데 각각의 station에 대해 buffer의 첫 번째 cell에 관한 상태방정식들은

$$\begin{aligned} B_m^1(k+1) &= B_{m+1}^1(k) + [1 - B_{m+1}^1(k)] A_{m+1}(k) \\ B_{M-1}^1(k+1) &= B_0^2(k) + [1 - B_0^2(k)] A_0(k) \end{aligned} \quad (1)$$

여기서, $m = 0, 1, \dots, M-2$

와 같이 표현되며 buffer의 두 번째에서 $N-1$ 번째까지의 cell에 관한 상태방정식들은

$$\begin{aligned} B_m^n(k+1) &= B_{m+1}^n(k) + [B_m^{n-1} - B_{m+1}^n(k)] A_{m+1}(k) \\ B_{M-1}^n(k+1) &= B_0^{n+1}(k) + [B_0^n(k) - B_0^{n+1}(k)] A_0(k) \end{aligned} \quad (2)$$

여기서, $m = 0, 1, \dots, M-2, \quad n = 2, 3, \dots, N-1$

그리고 끝으로 buffer의 마지막 cell에 관한 상태방정식들을 나타내면 다음과 같다.

$$\begin{aligned} B_m^N(k+1) &= B_{m+1}^N(k) + [B_m^{N-1} - B_{m+1}^N(k)] A_{m+1}(k) \\ B_{M-1}^N(k+1) &= \begin{cases} 0, & N \neq 1 \\ [1 - B_0^N(k)] A_0(k), & N = 1 \end{cases} \end{aligned} \quad (3)$$

여기서, $m = 0, 1, \dots, M-2$

(1)식에서 $B_0^2(k)$ 는 $[k, k+1]$ 시간동안 token을 가지고 있는 station이 network를 사용하고 있는지 아닌지를 나타내는 성능분석을 위한 기준이 되므로 이 상태변수를 출력 $Y(k)$ 로 놓았다.

$$Y(k) = B_0^2(k) \quad (4)$$

한편, 가정(5)에 의해 아래의 식이 만족되므로

$$\begin{aligned} E\{A_m(k)\} &= \text{constant} \\ E\{A_m(k)B_m^n(k)\} &= E\{A_m(k)\}E\{B_m^n(k)\} \end{aligned}$$

(1), (2), (3)식의 양변에 기대치를 취하여 이를 matrix형태로 정리하여 나타내면 다음과 같은 식들을 얻을 수 있다.

$$B(k+1) = G \cdot B(k) + H$$

$$y(k) = E \cdot B(k) \quad (5)$$

○ $N = 1$ 인 경우

$$\begin{bmatrix} b_0^1(k+1) \\ b_1^1(k+1) \\ \vdots \\ b_{M-2}^1(k+1) \\ b_{M-1}^1(k+1) \end{bmatrix} = \begin{bmatrix} 0 & Qa & 0 & \dots & 0 \\ 0 & 0 & Qa & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & Qa \\ -Pa & 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} b_0^1(k) \\ b_1^1(k) \\ \vdots \\ b_{M-2}^1(k) \\ b_{M-1}^1(k+1) \end{bmatrix} + \begin{bmatrix} Pa \\ Pa \\ \vdots \\ Pa \\ Pa \end{bmatrix}$$

$$y(k) = b_0^1(k)$$

여기서,

$$b_m^n(k) = E\{B_m^n(k)\}, \quad Pa = E\{A_m(k)\}, \quad Qa = 1 - Pa, \quad y(k) = E\{Y(k)\}$$

○ $N > 1$ 인 경우

$$\begin{bmatrix} b_0^1(k+1) \\ b_1^1(k+1) \\ b_2^1(k+1) \\ \vdots \\ b_{M-1}^1(k+1) \\ b_0^2(k+1) \\ b_1^2(k+1) \\ \vdots \\ b_{M-2}^2(k+1) \\ b_{M-1}^2(k+1) \end{bmatrix} = \begin{bmatrix} 0 & Qa & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & Qa & \dots & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \vdots & 0 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & Qa & 0 & \dots & 0 & 0 \\ 0 & Pa & 0 & \dots & 0 & 0 & Qa & \dots & 0 & 0 \\ 0 & 0 & Pa & \dots & 0 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & Pa & 0 & 0 & \dots & 0 & Qa \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 \end{bmatrix} \begin{bmatrix} b_0^1(k) \\ b_1^1(k) \\ b_2^1(k) \\ \vdots \\ b_{M-1}^1(k) \\ b_0^2(k) \\ b_1^2(k) \\ \vdots \\ b_{M-2}^2(k) \\ b_{M-1}^2(k) \end{bmatrix} + \begin{bmatrix} Pa \\ Pa \\ Pa \\ \vdots \\ Pa \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

$$y(k) = b_0^1(k)$$

2. 정상상태 해석

$$b_m^n(k) = b_m^n, \quad y(k) = y \quad \text{as } k \rightarrow \infty$$

로 놓고 (5)식을 출력(throughput) y 에 대해 풀면

$$y = E(I - G)^{-1}H = \frac{E \text{adj}(I - G)H}{|I - G|}$$

여기서, $E = [1 \ 0 \ \dots \ 0]^T$, I : NM 차 단위 행렬.

○ $N = 1$ 인 경우

$$y = \frac{1 - (1 - Pa)^M}{1 + Pa(1 - Pa)^{M-1}} \quad (6)$$

여기서, $M = 1, 2, \dots$

○ $N > 1$ 인 경우

$$y = \begin{bmatrix} Pa - Qa & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 \\ Pa & 1 & -Qa & \dots & 0 & 0 & 0 & \dots & 0 & 0 \\ Pa & 0 & 1 & \vdots & 0 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ Pa & 0 & 0 & \dots & 1 & -Qa & 0 & \dots & 0 & 0 \\ 0 & -Pa & 0 & \dots & 0 & 1 & -Qa & \dots & 0 & 0 \\ 0 & 0 & -Pa & \dots & 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -Pa & 0 & 0 & \dots & 1 & -Qa \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 - Qa & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & -Qa & \dots & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \vdots & 0 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & -Qa & 0 & \dots & 0 & 0 \\ 0 & -Pa & 0 & \dots & 0 & 1 & -Qa & \dots & 0 & 0 \\ 0 & 0 & -Pa & \dots & 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -Pa & 0 & 0 & \dots & 1 & -Qa \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix} \quad (7)$$

(7)식의 분모항에서 첫 행과 열 그리고 마지막 행과 열을 제외한 부분을 DL_M^N 으로 놓고 DL_M^N 에서 첫번째부터 m 번째까지의 행과 열을 제외한 부분을 $DL_M^N(m)$ 로 놓으면 분자항을 $DL_M^N(m)$ 의 함수로서 나타낼 수 있다. 이를 대입하여 출력 y 을 다시 쓰면

$$y = \frac{Pa \sum_{m=0}^{M-1} Qa^m DL_M^N(m)}{DL_M^N(0)} \quad (8)$$

가 된다. 여기서 $DL_M^N(m)$ 는 다음과 같은 iterative equation을 만족한다.

$$DL_M^N(m) = \begin{cases} DL_M^N(m+1) - PaQa^{M-1} DL_M^N(m+M), & m=0, \dots, (N-1)M-2 \\ 1, & \text{otherwise} \end{cases}$$

m 을 $(N-1)M-1$ 에서부터 0까지 순차적으로 대입하여 $DL_M^N(m)$ 을 구한뒤 출력을 구하면

$$y = \frac{Pa \sum_{m=0}^{M-1} \{Qa^m [1 + \sum_{n=0}^{N-2} S_n^m (-PaQa^{M-1})^{n+1}]\} + S_{N-2}^{M-1} (-PaQa^{M-1})^N}{1 + \sum_{n=0}^{N-2} S_n^0 (-PaQa^{M-1})^{n+1}}$$

여기서,

$$S_n^m = \begin{cases} (N-1)M-1-m, & n=0 \\ \sum_{i_n=1}^{(N-1-n)M-1-m} \dots \sum_{i_2=1}^{i_3} \sum_{i_1=1}^{i_2} i_1, & n=1, \dots, N-2 \\ m=0, 1, \dots, M-1, & N=2, 3, \dots, M=1, 2, \dots \end{cases}$$

위에서 구한 (6)과 (8)의 출력식은 가정 (1)에서 언급한 충분한 큰 M 에 대해서 뿐만 아니라 모든 M 에 대해서 적용될 수 있는 일반적인 식이다. 그리고 buffer로 들어온 data가 나가기까지 기다리는 평균 시간인 TD(time delay)는 현재 network의 모든 buffer에 들어있는 data들의 합을 출력 y 로 나누어주면 된다.

$$TD = \frac{\sum_{n=1}^N \sum_{m=0}^{M-1} h_n^m}{y} \quad (9)$$

하기의 그림 1과 2는 (6),(8),(9)식과 simulation 결과를 비교한 것으로서 두 결과가 거의 같음을 알 수 있다.

III. Nonuniform token-passing time interval의 경우의 성능 해석

이 경우는 token이 어떤 station에 도착했을 때 전송할 data가 있으면 data를 전송한 후 다음 station으로 가고 전송할 data가 없으면 바로 다음 station으로 간다. 이때 token이 어떤 station를 떠나 다음 station에 도착하는데 걸리는 시간을 T 라 하고 이 시간과 data 전송 시간과의 합을 τT 라고 놓는다.

1. 모델링

다음과 같이 정의 한다.

- $DT(k)$: Token이 k 번째로 station을 만날 때에서 부터 $k+1$ 번째로 station을 만날 때까지의 시간(T 또는 τT).
- $CT(k)$: 시간 t^k 에 어떤 station에 있는 token이 다시 그 station으로 돌아오는데 걸리는 시간(cycle time).
- $b_n(k)$: 시간 kT 에서 token이 임의의 station에 도착했다고 가정했을 때 그 station의 buffer에 n 개 이상의 data가 있었을 확률.

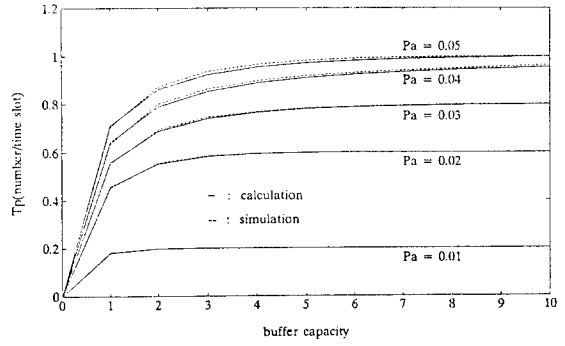


그림 1. Uniform token-passing time interval의 경우의 TP(throughput) 특성 ($M=20$)

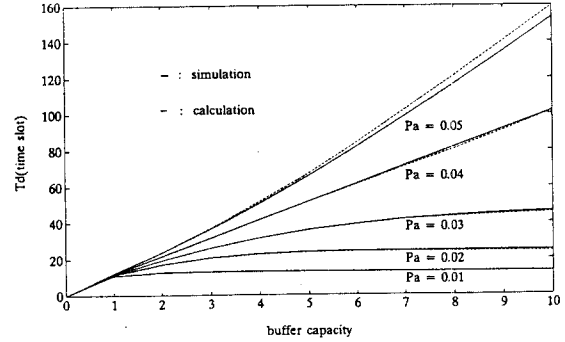


그림 2. TD(time delay) 특성 ($M=20$)

속확률. 이것은 $[k, k+1)$ 시간 동안 network이 사용중일 확률을 의미한다.

- $y_n(k)$: 시간 kT 에서 임의의 station이 token을 받고 동시에 그 station의 buffer에 n 개 이상 data가 있을 확률. 여기서 $y_1(k)$ 는 $[k, k+1)$ 시간 동안 내보내는 data의 평균 갯수를 의미하므로 이것이 출력(throughput)이 된다.

$$b_n(k+1) = b_n(kT + DT(k+1)), \quad n=1, 2, \dots, N$$

로 놓고 위의 정의들을 사용하여 상태 방정식을 세우면 다음과 같다.

- $N=1$ 인 경우

$$\begin{aligned} b_1(k+M) &= \sum_{i=0}^{M-1} [1 - b_1(k+i)]^M (1 - Qa^M) \\ &\quad + b_1(k) \sum_{i=1}^{M-1} [1 - b_1(k+i)]^M (1 - Qa^{M-1}) \\ &\quad + \dots + \sum_{i=0}^{M-1} b_1(k+i)^{M-1} [1 - Qa^{(M-1)(\tau-1)}] \\ &= f(b_1(k), b_1(k+1), \dots, b_1(k+M-1)) \end{aligned} \quad (10)$$

- $N > 1$ 인 경우

$$B(k+M) = G(k)B(k) + H(k) \quad (11)$$

여기서,

$$B(k) = [b_1(k) \ b_2(k) \ b_3(k) \ \dots \ b_{N-1}(k) \ b_N(k)]^T$$

$$H(k) = [S_1(k) \ S_2(k) \ S_3(k) \ \dots \ S_{N-1}(k) \ S_N(k)]^T$$

$$G(k) = \begin{bmatrix} 0 & A_0(k) & 0 & \dots & 0 & 0 \\ 0 & A_1(k) & A_0(k) & \dots & 0 & 0 \\ 0 & A_2(k) & A_1(k) & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & A_{N-2}(k) & A_{N-3} & \dots & A_1(k) & A_0 \\ 0 & A_{N-1}(k) & A_{N-2}(k) & \dots & A_2(k) & A_1(k) \end{bmatrix}$$

$$A_n(k) = \Pr\{\text{CT}(k)\text{동안 } n\text{개의 data가 들어옴}\}$$

$$= \sum_{\text{all } t} \Pr\{t \text{ 시간동안 } n\text{개의 data가 들어옴} / \text{CT}(k) = t\} \Pr\{\text{CT}(k) = t\}$$

$$= \sum P_n((M+i(\tau-1))(b_1)^i(1-b_1)^{M-i-1}(M-1)C_i)$$

$$S_n(k) = 1 - \sum_{i=0}^{n-1} A_i(k)$$

$$P_n(k) = P\{kT \text{ 시간동안 buffer로 } n\text{개의 data가 들어옴}\}$$

$$= {}_k C_n (Pa)^n (1-Pa)^{k-n}$$

2. 정상상태 해석

$$b_j(k) = b_j, \ S_j(k) = S_j, \ A_j(k) = A_j \quad \text{as } k \rightarrow \infty$$

으로 놓고 (10),(11)식들을 풀면

o $N = 1$ 인 경우

$$b_1 = \sum_{i=0}^{M-1} (b_1)^i (1-b_1)^{M-i} (1-Qa^{M+i(\tau-1)})_{M-1} C_i$$

$$+ \sum_{i=0}^{M-1} (b_1)^{i+1} (1-b_1)^{M-i-1} (1-Qa^{M+i(\tau-1)-1})_{M-1} C_i \quad (12)$$

여기서,

$${}_{M-1} C_i = \frac{(M-1)!}{(i)!(M-i-1)!}, \quad i = 0, 1, \dots, M-1$$

o $N > 1$ 인 경우

(11)식으로부터

$$b_1 = E(I-G)^{-1}H = \frac{Eadj(I-G)H}{|I-G|} \quad (13)$$

여기서, $E = [1 \ 0 \ \dots \ 0]^T$, I : N 차 단위 행렬.

(12), (13)식의 양변에 b_1 이 들어가 있으므로 양변의 식의 교점을 구해서 b_1 을 구할 수 있다. 그리고 우리가 구하고자하는 출력(throughput) y_1 은

$$y_1 = b_1 \cdot \Pr\{\text{token arrival}\}$$

$$= \frac{b_1}{1 + (\tau-1)b_1} \quad (14)$$

로 표현될 수 있다. 그리고 TD(time delay)를 구하기 위해 다음과 같은 상태변수들을 정의한다.

o $X_i(k)$: 시간 kT 에서 모든 station의 buffer의 i 번째 cell에 있는 data들 합. $X_i(k+1) = X_i(kT+T)$, $i = 1, 2, \dots, N$

o $\alpha_i(k)$: $[kT, (k+1)T)$ 시간 동안에 모든 station의 buffer의 i 번째 cell로 들어오는 data들의 합.

o $\beta_i(k)$: $[kT, (k+1)T)$ 시간 동안에 모든 station의 buffer의 i 번째 cell에서 나가는 data들의 합.

정의된 상태변수들을 사용하여 상태방정식을 세우면

$$X_i(k+1) = X_i(k) + \alpha_i(k) - \beta_i(k), \quad i = 1, 2, \dots, N$$

이 되는데 여기서 각각의 기대치를 다음과 같이 놓으면

$$E\{X_i(k)\} = X_i, \quad E\{\alpha_i(k)\} = \alpha_i$$

$$E\{\beta_i(k)\} = \beta_i, \quad i = 1, 2, \dots, N$$

양변에서 X_i 가 소거되어

$$\alpha_i = \beta_i, \quad i = 1, 2, \dots, N$$

의 식을 얻게 된다. 여기서,

$$\alpha_1 = y_2 + (M - X_1)Pa$$

$$\alpha_i = y_{i+1} + (X_{i-1} - X_i)Pa, \quad i = 2, \dots, N-1$$

$$\alpha_N = (X_{N-1} - X_N)Pa$$

$$\beta_i = y_i, \quad i = 1, 2, \dots, N$$

이 식을 X_i 에 대해 풀면

$$X_i = M - \frac{y_1 - y_{i+1}}{Pa}, \quad i = 1, \dots, N-1$$

$$X_N = M - \frac{y_1}{Pa}$$

이 된다. 따라서 time delay, TD 는

$$TD = \frac{1}{y_1} \sum_{i=0}^N X_i$$

$$= \frac{1}{y_1} [NM - \frac{1}{Pa}(-Ny_1 + \sum_{i=2}^N y_i)] \quad (15)$$

하기의 그림 3과 4는 (14), (15)식과 simulation 결과를 비교한 것으로서 두 결과가 거의 같음을 알 수 있다.

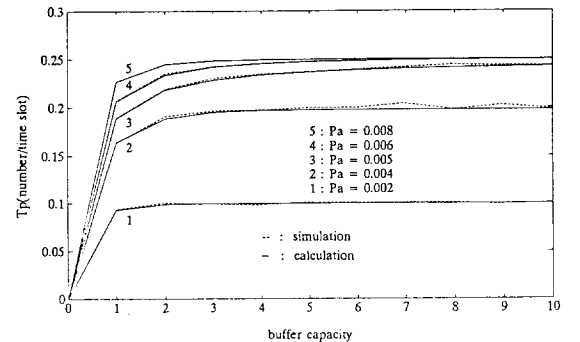


그림 3. Nonuniform token-passing time interval의 경우의 TP(throughput) 특성($M = 50, \tau = 4$)

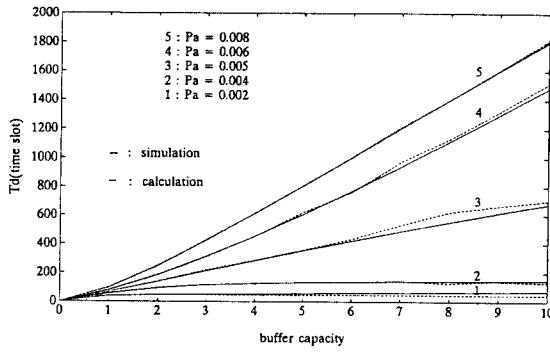


그림 4. TD(time delay) 특성 ($M = 50, r = 4$)

IV. buffer size의 적응 제어

앞에서 얻어진 결과는 buffer 용량에 따른 성능의 변화를 보여주는 데 그것으로부터 buffer 용량이 늘어 나면 TP(throughput)의 특성은 좋아지며 TD (time delay)의 특성은 나빠짐을 알 수 있다. 그러므로 시스템을 설계할때 이 두 특성을 모두 고려하여 원하는 성능특성에 부합하는 buffer 용량으로 station의 buffer 용량을 설계해 주면 된다. 그러나 이 값은 P_a (data generation probability)값에 따라 달라지는데 P_a 값은 실제에 있어서는 주어지는 값이 아니라 상황에 따라 바뀔 수 있는 미지의 값이므로 estimation을 통해 얻은 P_a 값을 근거로 각 station의 buffer 용량을 적당한 값으로 조정해 주면 된다. 이제 몇가지 가정을 시스템에 덧붙인다.

(8) 각각의 station은 자기 자신의 시스템 정보만을 알 수 있을뿐 나머지 다른 station에 대해서는 전혀 알 수 없다. 그리고 시스템 내에는 supervisor가 없어 각각의 station이 모두 자기 자신의 buffer capacity만을 제어하는 분산 제어 형태가 된다.

(9) 각각의 station은 buffer로 data가 들어왔느냐 아니냐에 대한 정보만 가질뿐 발생한 data중에 buffer가 가득차있어 들어오지 못한 것에 대한 정보는 알 수 없다.

(10) P_a 값에 대한 최적의 buffer 용량은 앞 절의 data로부터 표로 만들 수 있을 것이다. 이 표는 성능특성들(TP,TD)중 어느것에 더 큰 비중을 두느냐에 따라 달라질 수 있으므로 일단 이 논문에서는 P_a 에 대해 요구된 성능을 만족하는 buffer 용량을 나타내는 표는 만들어져 있다고 가정한다.

1. P_a 의 estimation

data가 buffer로 들어오기 위해서는 P_a 의 확률로 발생한 후 buffer에 비어있는 cell이 하나 이상 있어야 한다. 이것을 식으로 나타내면

$$A'(k) = A(k)[1 - B(k)]$$

여기서,

$A'(k)$: data arrival, $A(k)$: data generation, $B(k)$: buffer full
가 되는데 양변에 기대치를 취하면 하기와 같다.

$$P_a'(k) = P_a[1 - P_b(k)] \quad (16)$$

여기서 우리가 estimation하고자 하는 P_a 에 대한 정보는 직접적으로는 알 수 없으므로 $P_a'(k)$ 와 $P_b(k)$ 를 estimation한 후 위 식으로부터 P_a 의 estimation값을 구한다.

o $P_a'(k)$ 및 $P_b(k)$ 의 estimation

다음과 같은 $P_a'(k)$ 의 estimation식을 세운다.

$$\hat{P}_a'(k+1) = \hat{P}_a'(k) + \alpha(k) - \beta(k) \quad (17)$$

여기서, $\alpha(k)$ 와 $\beta(k)$ 의 결정을 위한 몇가지 근거를 들어보면 첫째, $\hat{P}_a'(k)$ 는 확률값을 estimation한 것이므로 정확한 estimation이라면 $\hat{P}_a'(k)$ 또한 확률의 기본 조건인 $0 \leq \hat{P}_a'(k) \leq 1$ 을 만족할 필요가 있다. 이를 위해 각 time slot의 증가분 $\alpha(k)$ 와 감소분 $\beta(k)$ 는 다음의 조건

$$0 \leq \alpha(k) \leq 1 - \hat{P}_a'(k), \quad 0 \leq \beta(k) \leq \hat{P}_a'(k)$$

을 만족해야한다. 둘째로 $A'(k) = 1$ 의 경우에는 $\alpha(k)$ 가 그리고 $A'(k) = 0$ 인 경우에는 $\beta(k)$ 가 $\hat{P}_a'(k)$ 에 기여하여야 함 것이다. 셋째로 $\alpha(k)$ 와 $\beta(k)$ 는 $\hat{P}_a'(k)$ 에 비해 충분히 작은 값이어야 한다. 그리고 넷째로 정상 상태에서 $\hat{P}_a'(k) = P_a'(k)$ 이 만족되어야 함 것이다.

이상의 몇가지 사항들을 고려하여 $\alpha(k)$ 와 $\beta(k)$ 를 정하면

$$\alpha(k) = \begin{cases} \frac{\hat{P}_a'(k)[1 - \hat{P}_a'(k)]}{Z}, & A'(k) = 1 \\ 0, & A'(k) = 0 \end{cases}$$

$$\beta(k) = \begin{cases} 0, & A'(k) = 1 \\ \frac{[\hat{P}_a'(k)]^2}{Z}, & A'(k) = 0 \end{cases} \quad (18)$$

여기서, $Z \gg 1$

(16)식에 $\alpha(k)$ 와 $\beta(k)$ 를 대입하여 양변에 기대치를 취하면

$$\hat{P}_a'(k+1) = \hat{P}_a'(k) + \frac{\hat{P}_a'(k)}{Z} [P_a'(k) - \hat{P}_a'(k)] \quad (19)$$

정상상태에서는 $\hat{P}_a'(k+1) = \hat{P}_a'(k)$ 이므로 $P_a'(k)$ 가 먼저 정상상태의 값 P_a' 에 도달한 후에 $\hat{P}_a'(k)$ 가 P_a' 이 됨을 알 수 있다.

$$\hat{P}_a'(k) \rightarrow P_a' \quad \text{as } k \rightarrow \infty$$

그리고, 위의 estimation식에서 Z 는 변화율을 조정해주는 factor로서 크면 클수록 delay가 커지고 작으면 작을수록 mean square error가 커진다.

o $\hat{P}_a'(k)$ 의 mean convergence.

(18)식으로부터 $0 \leq \alpha(k) < 1 - \hat{P}_a'(k)$ 이고 $0 \leq \beta(k) < \hat{P}_a'(k)$ 이므로 $\hat{P}_a'(k)$ 가 0과 1사이의 값이라면 $\hat{P}_a'(k+1)$ 또한 항상 0과 1사이의 값이 된다. 이것은 모든 k 에 대해 성립하므로 초기 상태 $\hat{P}_a'(0)$ 가 0과 1사이의 값이라면 $t = 0$ 후의 모든 시간에 대해 $\hat{P}_a'(k)$ 은 0과 1사이에 항상 머물러 있을 것이다.

$$0 < \hat{P}_a'(k) < 1, \quad k = 0, 1, \dots \quad (20)$$

한편, (19)식에서 $P_a'(k)$ 가 먼저 정상상태에 도달했다고 했을 때 식을 다시 쓰면 다음과 같은 형태로 쓸 수 있다.

$$\hat{P}_a'(k+1) - P_a' = \left[1 - \frac{\hat{P}_a'(k)}{Z} \right] [\hat{P}_a'(k) - P_a']$$

이 상태방정식은 두 개의 평형점 $\hat{P}a'(k) = 0, Pa'$ 를 가진다. 그리고 $Z \gg 1$ 이므로 $\hat{P}a'(k) \neq 0$ 이면 contraction의 조건

$$\left| 1 - \frac{\hat{P}a'(k)}{Z} \right| < 1, \quad k = 0, 1, \dots$$

을 만족한다. 그리고, (20)식에 의해 $\hat{P}a'(k)$ 는 0이 될 수 없으므로 위의 contraction조건에 의해 $\hat{P}a'(k)$ 는 평형점 Pa' 으로 수렴하게 된다.

동일한 방법으로 Pb 의 값 또한 estimation할 수 있으며 estimation된 두 값 $\hat{P}a'(k)$ 와 $\hat{P}b(k)$ 로부터 구하고자 하는 $\hat{P}a$ 은 아래 식

$$\hat{P}a = \frac{\hat{P}a'(k)}{1 - \hat{P}b(k)}$$

에 대입함으로써 구해진다.

그리고 그림 5에서 보듯이 Pa 가 변화하더라도 충분히 느리게 변한다면 Pa 의 estimation값 $\hat{P}a(k)$ 은 Pa 를 따라간다는 것을 알 수 있다.

이상에서 구해진 $\hat{P}a(k)$ 를 이용하여 원하는 성능에 맞는 buffer 용량을 찾을 수 있다.

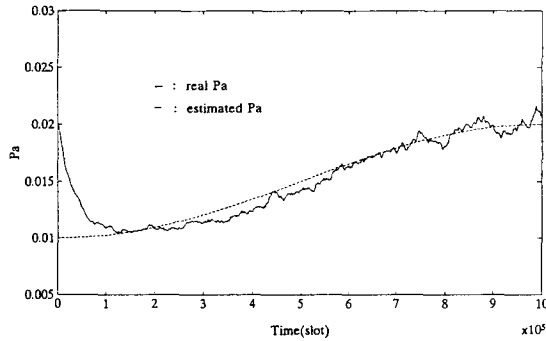


그림 5. Pa 의 estimation 특성 ($M = 10, N = 2, \tau = 3, Z = 500$)

V. 결론

이상에서 token-passing network의 두 가지 경우에 대한 성능분석을 위한 analytical formulae를 도출하였고 buffer 용량이 성능특성에 미치는 영향을 분석하였으며 system parameter인 Pa (data generation probability)의 estimation을 통한 주어진 성능을 만족하기 위한 buffer 용량의 제어에 대해 논하였다. 여기서 개발된 해석 기법은 생산시스템의 다양한 data communication에 확대 적용할 수 있을 것이다.

참고 문헌

- [1] A. S. Tanenbaum, "Computer network", Prentice-Hall, 1981.
- [2] S. R. Sachs, K. -I. Kan and J. A. Silvester, "Token-bus protocol performance analysis and comparison with other LAN protocols", Proc. IEEE GLOBECOM, 1985, pp. 1492 ~ 1498.

- [3] M. A. Colvin and A. C. Weaver, "Performance of single access classes on the IEEE 802.4 token-bus", IEEE Trans. 1986, COM-34, pp. 1253 ~ 1256.
- [4] M. Choi and C. M. Krishna, "An adaptive algorithm to ensure differential service in a token-ring network", IEEE Trans. Comput., 1990, 39, pp. 19 ~ 33.
- [5] P.-C. Wong and T. -S. P. Yum, "An integrated services token-controlled ring network", IEEE Trans. Commun., 1989, 7, pp. 670 ~ 679.
- [6] J. -T. Lim, S. M. Meerkov and T. Zeng, "Simplified description of slow-in-the-average Markov walks", J. Math. Anal. and Appls., 1991, 158, pp. 476 ~ 486.