

우선순위 및 제한 서서비스를 갖는 대규모 토큰-패싱 네트워크의 점근적 성능해석 및 적응제어

° 심광현, 임종태
한국과학기술원 전기 및 전자공학과

Asymptotic Performance Analysis and Adaptive Control of Large Scale Limited Service Token-Passing Networks with Priorities

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

Abstract

In this paper asymptotic formulae for performance characteristics(throughput, delay) of large scale token-passing networks with priorities and limited service are given. In particular, adaptive control procedures for obtaining optimal buffer capacity with respect to each priority and optimal limited service are shown. All results obtained are supported by simulations.

I. 서론

생산시스템의 완전 무인자동화 및 생산성의 극대화를 실현시키기 위해 만들어진 CIM(Computer Integrated Manufacturing)을 실제 적용시키기 위해서는 많은 어려움이 있다. 그중 하나가 여러 시스템간의 통신에 관한 문제이다. 이에 표준통신방식의 필요성이 대두되어 MAP(Manufacturing Automation Protocol)이 제정되었다[7]. 이러한 MAP의 기본 구조로써 토큰-패싱 네트워크(token-passing network)가 쓰이고 있는데 이것은 CSMA/CD 방식에 비해 부하가 큰 경우 특히 성능특성이 우수하며 유한 시지연을 갖는 등 많은 장점을 가지고 있어서[1],[2]. 성능특성 분석을 위한 많은 연구가 진행되어 왔으나[3],[4] 거의 단일 버퍼용량(buffer capacity) 및 무한 버퍼용량의 경우가 대부분이며 유한 버퍼용량의 경우에는 모델의 복잡성 때문에 정확한 해석이 어려워 점근적 해석기법을 이용한 성능특성 분석이 이루어지고 있다[5]~[7]. 그리고 데이터들이 발생되는 순서가 아닌 우선순위에 따라 전송순서가 정해지는 특정한 수의 우선순위들을 갖는 경우[8]~[10]나 단일 우선순위에서 특정한 서어비스 형태를 갖는 경우[11],[12] 및 각 스테이션이 우선순위의 수 만큼의 버퍼를 가지며 각 버퍼가 유한버퍼용량을 가지는 경우[7]등에 성능특성 분석에 대한 연구가 많이 이루어지고 있다.

본 논문에서는 점근적 해석기법을 더욱더 실제적인 면으로 확대 적용하여 유한 버퍼용량을 가지며 데이터들이 우선순위(priority)를 가지고 발생되며 토큰을 받은 스테이션이 한번에 전송할 수 있는 데이터의 수가 제한되는 제한 서어비스 토큰-패싱 네트워크(limited

service token-passing network)를 대상으로 성능특성 분석을 위한 해석적 공식을 유도한다. 그리고 각 우선순위에 해당하는 버퍼의 용량을 주어진 성능사양에 부합하도록 최적화시켜준후 제한 서어비스에서 최적의 제한값을 최적화시켜주기 위한 최적화 적응제어기법을 제시하고자 한다. 그리고 본 논문에서는 임의의 우선순위 및 임의의 유한 버퍼용량과 특히 여러가지 서어비스 형태를 갖는 토큰-패싱 네트워크를 대상으로하여 각각의 시간에 대한 상태변화를 상태방정식을 세워 표현해 줌으로써 보다 해석적인 방법으로 성능특성을 해석해줄 수 있고 또한 정상상태 뿐만 아니라 시간에서의 시스템의 상태변화 또한 추정이 가능하다는 특징이 있다. 우선 본 논문에서 다루는 시스템을 하기와 같이 설정한다.

(1) 전체 시스템은 $1 \leq M < \infty$ 개의 스테이션들로 구성되며 각 스테이션에서 발생하는 데이터는 $1 \leq L < \infty$ 개의 우선순위를 갖고며 모든 스테이션은 각 우선순위의 데이터들을 위해 L 개의 버퍼를 가지며 각각의 버퍼는 $1 \leq N_i < \infty$ 개의 데이터를 저장할 수 있는 용량을 가진다. ($i = 1, 2, \dots, L$)

(2) 데이터의 전송은 한 스테이션에서 네트워크상의 다음 순서의 스테이션으로 이루어지며 한쪽 방향으로만 전송이 가능하다.

(3) 네트워크상에는 스테이션의 네트워크 사용을 제어하는 제어 비트(control bit)들의 모임인 토큰(token)이 존재하게 된다. 이 토큰은 유일하며 계속해서 네트워크상을 순회하게 되는데 일단 토큰을 받은 스테이션은 버퍼에 데이터가 있으면 그 중 우선순위가 가장 높은 데이터들중에서 가장 먼저 들어온 하나의 데이터만이 토큰에 의해 서어비스를 받는다. 그러나 만약 버퍼가 비어 있는 때에는 일정 시지연 후에 토큰을 다음 스테이션에 전달한다. 여기서, 토큰을 재전송하기 전에 토큰을 가진 스테이션이 얼마나 많은 데이터를 전송할 수 있는가에 따라 다음의 4가지 서어비스 시스템으로 구분된다.

1. R-limited service - 토큰을 가진 스테이션은 데이터를 하나 이상 가지고 있는 버퍼들중에서 우선순위가 가장 높은 버퍼에 있는 데이터중 최대 R개까지 전송할 수 있다.

2. Nonexhaustive service - 토큰을 가진 스테이션은 해당 버퍼에

있는 데이터중 단 하나만을 전송할 수 있다(1-limited service).

3. Gated service-토큰을 가진 스테이션은 토큰이 도착했을때 그 스테이션의 해당 버퍼에 들어있는 모든 데이터를 전송할 수 있다. 단 토큰이 도착한 후에 들어온 데이터는 전송할 수 없다.

4. Exhaustive service - 토큰을 받은 스테이션은 해당 버퍼가 완전히 빌때까지 모든 데이터를 전송할 수 있다(∞ - limited service).

(4) 토큰이 임의의 한 스테이션을 떠나 네트워크상의 다음 스테이션에 도달하는데 걸리는 시간(토큰 전달 시간)은 모든 스테이션에 대해서 동일하다.

(5) 같은 우선순위의 데이터 발생(data generation)에 대한 확률 분포는 모든 스테이션에 대해 동일하며 버퍼에 이미 들어있는 데이터의 양 및 시간에 대해 무관하다.

(6) 같은 우선순위의 모든 데이터에 대해 그것이 전송되는데 걸리는 시간은 스테이션에 관계없이 동일하다.

이상에서 가정된 시스템을 두가지 관점에서 성능특성 분석을 하고자 한다. 그 첫 번째는 네트워크 즉 서버(server)관점에서의 성능 특성으로서 출력(throughput)을 구하고 두 번째로는 스테이션 즉 사용자(client)관점에서의 성능특성으로서 시지연(time delay)을 구한다. 한편, 본 논문의 모든 모의실험은 C 언어에 의해 소프트웨어(software)로 구현된 시뮬레이터(simulator)에 의한 결과이다.

II. 본론

1. 정의

T : 토큰이 임의의 스테이션을 떠나 다음 스테이션에 도달하는데 걸리는 시간. 즉, 토큰 전달시간으로서 모든 시간은 이 시간을 단위시간슬롯(unit time slot)으로하여 이산화됨.

M : 네트워크상의 스테이션의 수.

L : 데이터들의 구분되는 우선순위의 수. 즉, 각각의 스테이션이 가지는 버퍼의 갯수.

R : 토큰을 받은 스테이션이 한번에 전송할 수 있는 최대 데이터의 수(R -limited service).

N_i : 각각의 스테이션에서 i 번째 우선순위의 데이터를 위해 할당된 버퍼의 용량.

τ_i : i 번째 우선순위 데이터의 전송시간.

P_{ai} : 임의의 한 스테이션에서 단위시간슬롯(T)동안 i 번째 우선순위의 데이터가 발생할 확률.

t^k : 임의의 한 스테이션이 k 번째로 토큰을 받은 시간.

$T_c(k)$: 토큰이 네트워크 상을 한 바퀴 도는데 걸리는 시간. 즉, 토큰순회시간($= t^{k+1} - t^k$).

$x_i(k)$: 토큰순회시간($T_c(k)$)동안에 토큰이 만난 스테이션들중에서 i 번째 우선순위의 데이터가 전송된 스테이션의 수.

$A_{ij}(k)$: 토큰순회시간($T_c(k)$)동안에 i 번째 우선순위의 데이터가 j 개 들어올 확률.

$$C_{ij}(k) : 1 - \sum_{n=0}^{j-1} A_{in}(k)$$

$b_{ij}(k)$: 시간 t^k 에 토큰을 막 받은 스테이션의 i 번째 우선순위 버퍼의 j 번째 셀에 데이터가 들어있을 확률.

$u_{ij}(k)$: 시간 t^k 에 토큰을 막 받은 스테이션에서 $i-1$ 번째 우선순위 버퍼까지는 비어있고 i 번째 우선순위 버퍼의 j 번째 셀에는 데이터가 들어있을 확률.

$w_{ij}(k)$: $[kT, (k+1)T]$ 시간동안에 i 번째 우선순위 버퍼의 j 번째 셀에서 데이터가 나갈 확률.

$X_{ij}(k)$: 시간 kT 에서 모든 스테이션의 i 번째 우선순위 버퍼의 j 번째 셀에 들어있는 데이터들의 합 of 평균.

TP_i : 정상상태에서 단위시간슬롯 동안에 전송되는 i 번째 우선순위 데이터의 평균 갯수(출력).

TD_i : 정상상태에서 i 번째 우선순위의 데이터가 버퍼에 들어온 후 나가는데까지 걸리는 평균 시간(시지연).

α_i : i 번째 우선순위 데이터의 출력(TP_i)최적화 지수.

β_i : i 번째 우선순위 데이터의 시지연(TD_i)최적화 지수.

2. 모델링

첫 번째 우선순위의 데이터 버퍼에 대한 상태를 모델링을 하면 하기와 같은 상태방정식을 얻을 수 있다.

$$\begin{aligned} b_{11}(k+1) &= A_{10}(k)b_{1R+1}(k) + C_{11}(k) \\ b_{12}(k+1) &= A_{11}(k)b_{1R+1}(k) + A_{10}(k)b_{1R+2}(k) + C_{12}(k) \\ &\vdots \\ b_{1N_1}(k+1) &= A_{1N_1-1}(k)b_{1R+1}(k) + \dots + A_{11}(k)b_{1N_1+R-1}(k) \\ &\quad + C_{1N_1}(k) \end{aligned} \quad (1)$$

여기서, $N_1 = 1, 2, \dots$

그리고, 이것을 일반화하면 시간 t_{k+1} 에 i 번째 우선순위 버퍼의 j 번째 셀에 데이터가 들어있을 확률인 $b_{ij}(k+1)$ 를 구하면

$$\begin{aligned} b_{ij}(k+1) &= C_{ij}(k) + \left(1 - \prod_{n=0}^{i-1} [1 - b_{n1}(k)]\right) \sum_{n=0}^{j-1} A_{in}(k)b_{ij-n}(k) \\ &\quad + \prod_{n=0}^{i-1} [1 - b_{n1}(k)] \sum_{n=0}^{j-1} A_{in}(k)b_{iR+j-n}(k) \end{aligned} \quad (2)$$

여기서, $i = 1, 2, \dots, L, \quad j = 1, 2, \dots, N_i$

$$A_{ij}(k) = 0 \quad \text{for } j < 0$$

$$b_{ij}(k) = 0 \quad \text{for } i \leq 0 \text{ or } j > N_i$$

가 된다. 이식에서 보면 토큰순회시간동안에 j 개 이상의 데이터가 들어오면 즉 $C_{ij}(k) = 1$ 이면 $b_{ij}(k+1)$ 는 시간 t^k 에서의 상태에 상관

없이 항상 1이 된다. 그렇지 않고 j 개 이하의 데이터가 들어왔을 때에는 두가지 경우로 나누어진다. 첫번째로 시간 t^k 에 i 번째 보다 높은 우선순위의 데이터가 있을 경우에는 i 번째 우선순위 버퍼에 있는 데이터는 전송되어 나가지 못하므로 그 갯수는 토큰순회시간동안 들어온 i 번째 우선순위 데이터의 수만큼 증가하게 된다. 그러므로 $b_{ij}(k+1) = 1$ 이 되기 위해서는 시간 t^k 에 i 번째 우선순위 버퍼에 있는 데이터들과 토큰순회시간동안 들어온 i 번째 우선순위 데이터들의 합이 j 개 이상이면 된다. 그리고 두번째로 i 번째 보다 높은 우선순위의 데이터가 없을 경우에는 i 번째 우선순위의 데이터가 R 개까지 전송되어 없어질 수 있으므로 $b_{ij}(k+1) = 1$ 이 되기 위해서는 시간 t^k 에 i 번째 우선순위 버퍼에 있는 데이터들과 토큰순회시간동안 들어온 i 번째 우선순위 데이터들의 합이 $j + R$ 개 이상이어야 한다. 그리고 $X_{ij}(k)$ 에 대한 상태방정식을 세우면

$$X_{ij}(k+1) = X_{ij}(k) + y_{ij+1}(k) + [1 - (1 - P_{ai})^{X_{ij-1}(k) - X_{ij}(k)}] - y_{ij}(k) \quad (3)$$

여기서, $i = 1, 2, \dots, L, j = 1, 2, \dots, N_i$

$$X_{i0}(k) = M, \quad X_{iN_i+1}(k) = 0$$

가 된다. 이식을 살펴보면 $X_{ij}(k+1)$ 는 $X_{ij}(k)$ 에다가 들어온 데이터의 갯수를 더해주고 나간 데이터의 갯수($y_{ij}(k)$)를 빼준 것이 된다. 그리고 새롭게 들어온 데이터는 또다시 위의 버퍼셀에서 내려온 것($y_{ij+1}(k)$)과 외부로부터 발생되어 들어온 것으로 나누어진다.

3. 점근적 성능해석

모든 상태변수들이 정상상태에 도달하였다고 가정하고 상태변수들의 정상상태 값을 다음과 같이 놓으면

$$b_{ij}(k) = b_{ij}, \quad u_{ij}(k) = u_{ij}, \quad y_{ij}(k) = y_{ij}, \quad X_{ij}(k) = X_{ij}$$

$$A_{ij}(k) = A_{ij}, \quad C_{ij}(k) = C_{ij}, \quad x_i(k) = x_i, \quad k \rightarrow \infty$$

(2)식은 다음과 같이 쓰여질 수 있다.

$$b_{ij} = C_{ij} + \left(1 - \prod_{n=0}^{i-1} [1 - b_{n1}]\right) \sum_{n=0}^{j-1} A_{in} b_{ij-n} + \prod_{n=0}^{i-1} [1 - b_{n1}] \sum_{n=0}^{j-1} A_{in} b_{iR+j-n} \quad (4)$$

여기서, $i = 1, 2, \dots, L, j = 1, 2, \dots, N_i$

$$A_{ij} = 0 \quad \text{for } j < 0$$

$$b_{ij} = 0 \quad \text{for } i \leq 0 \text{ or } j > N_i$$

여기서 토큰순회시간동안 j 개의 데이터가 발생할 확률인 A_{ij} 를 구하기 위해서는 먼저 다음과 같은 것을 고려해야 한다. 그것은 토큰 순회시간이 그 시간동안 토큰이 나머지 스테이션에게 얼마나 많은 서비스를 제공하느냐에 따라 달라진다는 것이다. 이에 A_{ij} 는

$$A_{ij} = \sum_{x_1=0}^M \sum_{x_2=0}^{M-x_1} \dots \sum_{x_L=0}^{M-x_1-\dots-x_{L-1}} \frac{M!}{x_1! \dots x_L! (M - \sum_{n=1}^L x_n)!}$$

$$\times (u_{11})^{x_1} \dots (u_{L1})^{x_L} \left(1 - \sum_{n=1}^L u_{n1}\right)^{M - \sum_{n=1}^L x_n} \times P_{ij} \left\{ M + \sum_{n=1}^L \tau_n x_n \left(\frac{\sum_{m=1}^R u_{nm}}{u_{n1}} \right) \right\} \quad (5)$$

여기서, $i = 1, 2, \dots, L, j = 1, 2, \dots, N_i$

$$P_{ij}(n) = {}_n C_j (P_{ai})^j (1 - P_{ai})^{n-j}$$

$$u_{ij} = \left[\prod_{n=1}^{i-1} (1 - b_{n1}) \right] b_{ij}$$

가 되어 (4),(5)식으로부터 b_{ij} 를 구하여 u_{ij} 를 구할 수 있다. 그리고 단위시간슬롯동안 모든 스테이션의 i 번째 우선순위 버퍼의 j 번째 셀에서 데이터가 나갈 확률인 y_{ij} 는 u_{ij} 를 토큰이 임의의 한 스테이션에 머무르는 평균 시간으로 나누어주면 된다. 즉,

$$y_{ij} = \frac{u_{ij}}{1 + \sum_{n=1}^L \tau_n u_{n1}}$$

가 된다. 이때, i 번째 우선순위 데이터의 출력 TP_i 는

$$TP_i = y_{i1} \quad (6)$$

이다. 그리고 전체 출력 TP 는

$$TP = \sum_{i=1}^L TP_i$$

가 된다. 한편 (3)식은 정상상태에서 다음과 같이 쓰여질 수 있다.

$$X_{ij} = X_{ij} + y_{ij+1} + 1 - (1 - P_{ai})^{X_{ij-1} - X_{ij}} - y_{ij}$$

여기서, $i = 1, 2, \dots, L, j = 1, 2, \dots, N_i$

$$X_{i0} = M, \quad X_{iN_i+1} = 0$$

일반적으로 $P_{ai} \ll 1$ 이고 $(X_{ij-1} - X_{ij})P_{ai} \ll 1$ 이므로 다음과 같이 근사화시킬 수 있다.

$$y_{ij} = y_{ij+1} + (X_{ij-1} - X_{ij})P_{ai}$$

이 식을 $j = 1$ 에서부터 n 까지 대입하여 모두 더해주면

$$y_{i1} = y_{in+1} + (M - X_{in})P_{ai}, \quad n = 1, 2, \dots, N_i$$

가 된다. 이것을 X_{in} 에 대해 풀어주면

$$X_{in} = M - \frac{y_{i1} - y_{in+1}}{P_{ai}}$$

이다. 이때 i 번째 우선순위 데이터의 총 갯수는

$$\sum_{n=1}^{N_i} X_{in} = N_i M - \frac{1}{P_{ai}} [N_i y_{i1} - \sum_{n=2}^{N_i} y_{in}]$$

이다. i 번째 우선순위 데이터의 평균시지연인 TD_i 는 위에서 구한 정상상태에서 구한 i 번째 우선순위 데이터의 총수를 i 번째 우선순위 데이터의 출력으로 나누어 주면 된다[6].

$$TD_i = \frac{\sum_{n=1}^{N_i} X_{in}}{TP_i} = \frac{1}{y_{i1}} \left[N_i M - \frac{1}{P_{ai}} (N_i y_{i1} - \sum_{n=2}^{N_i} y_{in}) \right] \quad (7)$$

그리고 전체 평균 시지연 TD는

$$TD = \frac{\sum_{i=1}^L TP_i TD_i}{TP} \quad (8)$$

이다. 그림[1]~[4]는 (6),(7)식에서의 출력과 시지연의 해석된 결과와 모의 실험 결과를 제한 서어비스의 제한값에 대해 나타낸 것으로서 해석된 결과의 타당성을 보여주고 있다.

4. 버퍼용량 및 제한 서어비스의 적응제어기법

우리가 각각 스테이션들에게 버퍼를 부가하는 이유는 출력 향상을 위해서이다. 그러나, 모든 데이터들이 동일한 우선순위를 갖는 경우에 일반적으로 출력은 항상 버퍼용량의 증가에 대해 포화현상을 나타내지만 시지연은 데이터 발생율이 어느 정도 이상이면 포화현상을 보이지 않고 계속해서 증가하게 되므로 지나치게 버퍼용량을 크게하는 것은 출력의 향상없이 시지연만을 증가시켜 전체적으로 성능특성을 저하시키는 효과를 가져오므로 출력향상을 위한 버퍼용량의 증가는 출력곡선이 포화가 되기 시작하는 곳으로 제한하는 것이 바람직하다[7]. 출력의 포화현상과 우선순위를 고려하여 버퍼용량을 최적화하기 위해 다음과 같은 절차를 제안한다.

◦ 버퍼용량의 최적화 절차(Procedure)

i) 출력(TP_k)의 극대화

$$SP_k = \{(n_1, \dots, n_L) | (n_1, \dots, n_L) \in SP_{k-1}, TP_k(n_1, \dots, n_L) \geq \alpha_k \times \max_{(n_1, \dots, n_L) \in SP_{k-1}} [TP_k(n_1, \dots, n_L)]\} \quad (9)$$

여기서, $k = 1, \dots, L$

$$SP_0 = \{(n_1, \dots, n_L) | 0 \leq n_i < \infty, i = 1, \dots, L\}$$

ii) 시지연(TD_k)의 극소화

$$SD_k = \{(n_1, \dots, n_L) | (n_1, \dots, n_L) \in SD_{k-1}, n_{k-L} = \min_{(n_1, \dots, n_L) \in SD_{k-1}} [n_{k-L}]\} \quad (10)$$

여기서, $k = 1, \dots, L$

$$SD_0 = SP_L$$

그리고 일반적으로 단일 우선순위의 경우에는 제한 서어비스의 제한값 R을 크게해주면 출력과 시지연특성이 모두 향상되게 된다. 왜냐하면 제한값 R이 크면 클수록 한 스테이션이 한번에 보낼 수 있는 데이터의 수가 많아지므로 토큰전달시간에 의해 소비되는 시간을 줄일 수 있다 그래서 출력특성이 좋아지게 되며 이로 인해 시지연특성 또한 좋아지게 된다. 그러나 단일 우선순위가 아닌 경우 다른 양상을 보임을 알 수 있다. 즉 성능특성들이 서로 연관이 되어 제한값(R)의 증가에 따라 성능특성이 좋아질 수도 나빠질 수도 있다는 것이다. 이에 본 논문에서는 먼저 제한값을 1로한 경우에 대해 위의 버퍼용량의 최적화기법을 이용하여 최적의 버퍼용량을 구한다. 그런다음, 그와같은 최적의 버퍼용량하에서 제한값을 변화시켜 주어 최적의 제한 서어비스를 구해주는 다음과같은 기법을 제시

한다.

◦ 제한 서어비스의 최적화 절차(Procedure)

i) 출력(TP_k)의 극대화

$$WP_k = \{R | R \in WP_{k-1}, TP_k(R) \geq \alpha_k \times \max_{R \in WP_{k-1}} [TP_k(R)]\} \quad (11)$$

여기서, $k = 1, \dots, L$

$$WP_0 = \{1, 2, \dots, \max_{i=1,2,\dots,L} [N_i]\}$$

ii) 시지연(TD_k)의 극소화

$$WD_k = \{R | R \in WD_{k-1}, TD_k(R) \leq \beta_k \times \min_{R \in WD_{k-1}} [TD_k(R)]\} \quad (12)$$

여기서, $k = 1, \dots, L$ $WD_0 = WP_L$

그림[5]~[8]은 시간에 따라 데이터 발생율이 서서히 변하는 경우 ($Pa_1 = 5 \times 10^{-9}t$, $Pa_2 = 2 \times 10^{-8}t$)에 대해 데이터발생율을 추정하고 최적화 기법들을 이용하여 버퍼용량 및 제한 서어비스를 제어해준 경우와 버퍼용량만을 제어해준 경우($R=1$) 그리고 전혀 제어하지 않은 경우($N_1 = N_2 = 9$, $R=1$)에 대한 성능특성을 모의 실험을 통해 비교분석한 것이다.

III. 결론

이상에서 데이터들이 우선순위를 가지고 발생하고 유한 버퍼용량을 갖는 대규모 제한 서어비스 토큰-패싱 네트워크의 성능특성을 분석하였고 각 우선순위에 해당하는 버퍼의 용량을 최적화하는 기법과 제한 서어비스의 제한값을 최적화하는 기법을 제시하였다. 그리고 이러한 기법을 통한 버퍼용량과 제한 서어비스의 적응제어로 성능특성의 향상을 도모할 수 있었다. 끝으로 앞의 모든 결과들은 다양한 종류의 여러 데이터 통신망(data communication network)에 확대 적용할 수 있을 것이다.

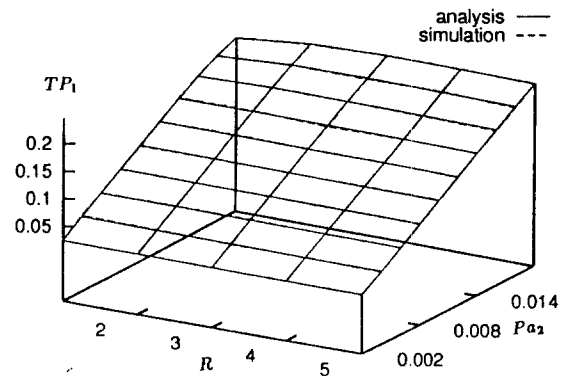


그림 [1]. 제한 서어비스에 대한 1번째 우선순위 데이터의 출력 ($M = 50$, $L = 2$, $\tau_1 = \tau_2 = 3$, $N_1 = N_2 = 5$, $Pa_1 = \frac{Pa_2}{4}$)

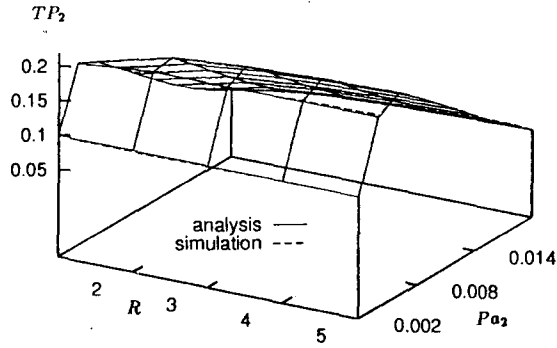


그림 [2]. 제한 서비스에 대한 2번째 우선순위 데이터의 출력
 ($M = 50, L = 2, \tau_1 = \tau_2 = 3, N_1 = N_2 = 5, Pa_1 = \frac{Pa_2}{4}$)

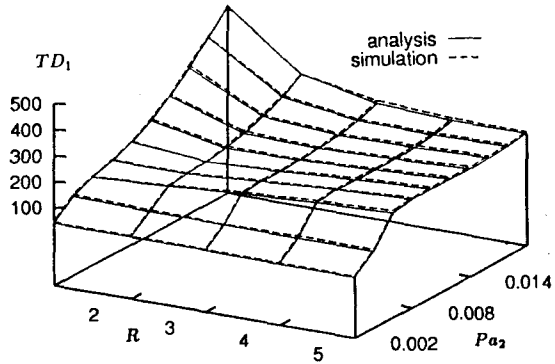


그림 [3]. 제한 서비스에 대한 1번째 우선순위 데이터의 시지연
 ($M = 50, L = 2, \tau_1 = \tau_2 = 3, N_1 = N_2 = 5, Pa_1 = \frac{Pa_2}{4}$)

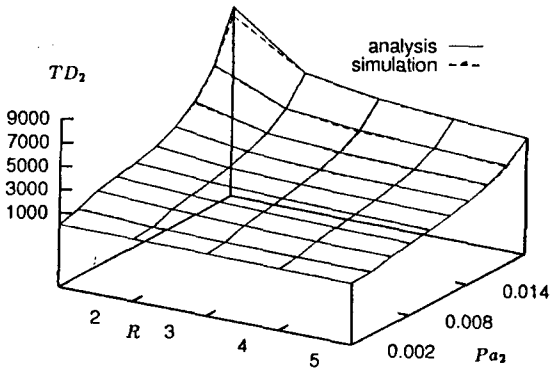


그림 [4]. 제한 서비스에 대한 2번째 우선순위 데이터의 시지연
 ($M = 50, L = 2, \tau_1 = \tau_2 = 3, N_1 = N_2 = 5, Pa_1 = \frac{Pa_2}{4}$)

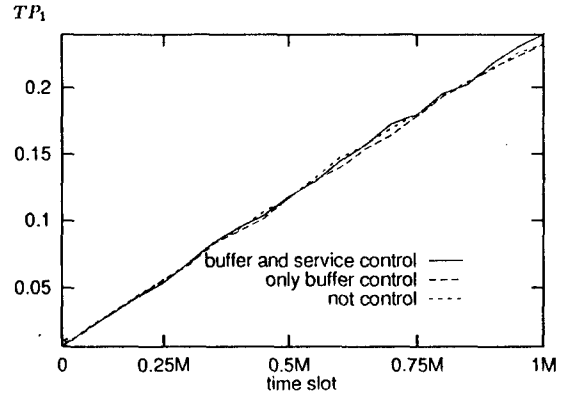


그림 [5]. 시간에 따른 1번째 우선순위 데이터의 출력
 ($M = 50, L = 2, \tau_1 = \tau_2 = 3, \alpha_1 = \alpha_2 = 0.95, \beta_1 = 1.02$)

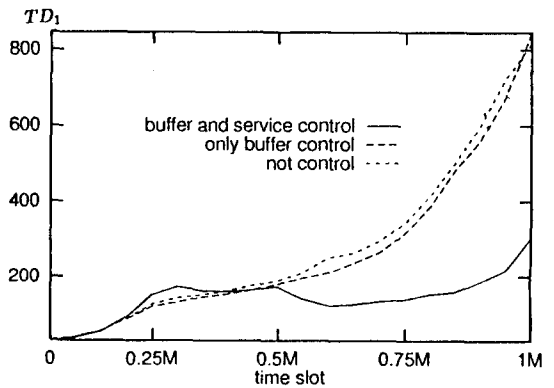


그림 [6]. 시간에 따른 1번째 우선순위 데이터의 시지연
 ($M = 50, L = 2, \tau_1 = \tau_2 = 3, \alpha_1 = \alpha_2 = 0.95, \beta_1 = 1.02$)

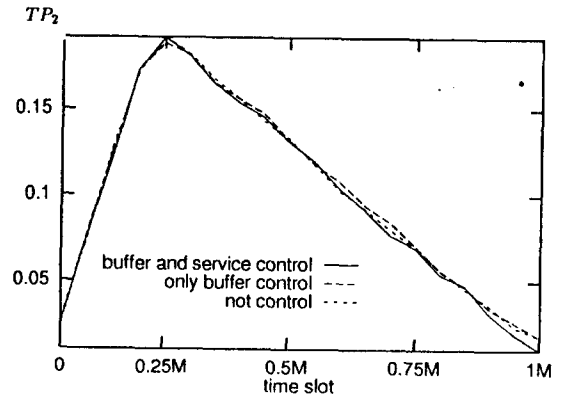


그림 [7]. 시간에 따른 2번째 우선순위 데이터의 출력
 ($M = 50, L = 2, \tau_1 = \tau_2 = 3, \alpha_1 = \alpha_2 = 0.95, \beta_1 = 1.02$)

TD_2

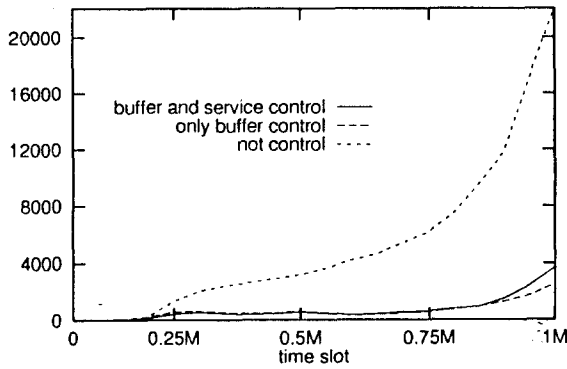


그림 [8]. 시간에 따른 2번째 우선순위 데이터의 시지연
 $(M = 50, L = 2, \tau_1 = \tau_2 = 3, \alpha_1 = \alpha_2 = 0.95, \beta_1 = 1.02)$

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] P.-C. Wong and T. -S. P. Yum, "An integrated services token-controlled ring network," IEEE Trans. Commun., 1989, 7, pp. 670 ~ 679.
- [5] 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.
- [6] 심 광현, 임 종태, "대규모 토큰-패싱 네트워크의 점근적 성능 분석 및 적응제어," KACC논문집, 1992, 158, pp. 37 ~ 42.
- [7] 심 광현, 임 종태, "우선순위를 갖는 대규모 토큰-패싱 네트워크의 점근적 성능해석 및 설계," 제어계측연구회 합동학술연구발표회 논문집, 1993, pp. 119 ~ 122.
- [8] B. Grella-M'Poko, M. Mehmet Ali and J. F. Hayes, "Approximate Analysis of Asymmetric Single-Service Prioritized Token Passing Systems," IEEE Trans. Comm. vol. 39, no. 10, July, 1981, pp. 1037 ~ 1040.
- [9] D. R. Manfield, "Analysis of a Priority Polling System for Two-Way Traffic," IEEE Trans. on Comm. COM-33, no. 9, 1985, pp. 1001 ~ 1006.

- [10] J. Gianini and D.R. Manfield, "An Analysis of Symmetric Polling Systems with Two Priority Classes," Perform. Eval., vol. 8, 1988. pp. 93 ~ 115.
- [11] H. Takagi, "Mean Message Waiting Times in Symmetric Multi-Queue Systems with Cyclic Service," Performance '85 (North-Holland), 1985, pp. 271 ~ 277.
- [12] O. J. Boxma and B. W. Meister, "Waiting-Time Approximations in Multi-Queue Systems with Cyclic Service", Performance Evaluation 7(1), 1987, pp 59~70.