

실시간 응용시 Mini-MAP 의 성능해석에 관한 연구

○ 김 덕 우 정 범 진 권 옥 현  
서울대학교 제어계측공학과

Performance analysis of Mini-MAP Architecture in Realtime Applications

Deok Woo Kim Bum Jin Chung Wook Hyun Kwon  
Dept. of Control and Instrumentation, Seoul National University

Abstract

In this paper, the performance of Mini MAP architecture in realtime applications is analyzed by analytic methods. The approximate equations for token rotation time and waiting time are obtained and using these equations network performance is evaluated. According to the results obtained, Mini MAP architecture has good realtime performance and can be used to many realtime applications.

1. 서론

일정한 지역에 위치하고있는 관련기기들을 상호연결하여 자원의 공유를 가능하게 하여주는 통신망이 LAN(Local Area Network)이다. LAN은 최근 몇 년 동안 커다란 발전을 보여 왔으며, 이제 수십종류의 상업용 LAN이 등장하여 공장자동화에도 커다란 도음을 주고있다. 그러나 자동화 설비들이 공급업체의 독자적인 LAN규격에 적합하게 개발되어왔기 때문에, 설비간에 정보교환량이 증가하고 있음에도 불구하고 공급업체가 서로 다른 자동화 설비들 사이의 통신은 갈수록 어려워지고있는 실정이다[1]. 이에 따라서 공장자동화에 소요되는 비용도 급격하게 증가하여, 비용의 50%가 공급업체가 서로 다른 자동화설비들 사이의 통신문제 해결을 위하여 사용되고있으며, 공장자동화가 진행되어감에 따라 이는 더욱 증가할 추세이다[1].

이러한 문제점을 해결하기위하여 GM(General Motors)은 1982년 공장자동화에 관한 LAN의 표준규격으로써 MAP (Manufacturing Automation Protocol)을 제창하였다. MAP은 ISO (International Standard Organization)의 OSI 참고모델 (Open Systems Interconnection Reference Model)에 기초하여 제정된 것으로 표준화가 진행중이거나 표준화가 완료된 프로토콜(Protocol)을 따르고있다.

MAP의 구조는 Full MAP 구조와 Mini MAP 구조로 나뉜다[1]. Full MAP 구조는 OSI 참고모델의 7 계층 모두를 가지는 것으로

MAP 사양서에서 정식으로 제안된 것이다. 그러나 각 계층에서의 시간지연으로인하여 실시간 응용에는 많은 어려움이 따른다[1]. Mini MAP 구조는 OSI 참고모델의 3계층만을 갖는 구조로써 실시간 응용을 목적으로 MAP 사용자 그룹에 의하여 제안되어 표준화 작업이 진행중인 구조이다[1]. Mini MAP 구조는 시간지연이 크게 줄어들고 비용이 적게들어 node의 수가 수십개 정도되는 실시간 시스템에 적용이 크게 확대될 전망이다[2]. 그림-1은 MAP의 구조이다.

MAP의 성능에 결정적인 영향을 갖는 물리적 계층은 Token Passing Bus protocol이다. 토큰패싱 LAN의 성능해석에 관한 연구로는 J.R Pimentel [3], W. Bux [4], R Cherukuri [5], A.S. Sethi [6], M.E. Uing [7][8], H. Okada [9], 이도영 [10] 등의 연구가 있다. 그러나 위의 연구 결과들은 실제 Token Passing Bus에서의 여러가지 오버헤드를 고려하고 있지 않아 MAP의 성능해석에 적용해서는 만족할만한 결과를 얻을 수가 없다. 본 논문에서는 IEEE 802.4 Token Passing Bus에서의 여러가지 오버헤드를 고려하고, MAP이 공장형 LAN이라는 사실에 증정을 두어 실시간 응용시에 Mini-MAP 구조의 성능을 해석적 모델을 사용하여 분석한다.

	Full MAP 구조	Mini MAP 구조
7. Application	ISO CASE kernel	ISO CASE kernel
6. Presentation	Null	Null
5. Session	ISO Session Kernel	Null
4. Transport	ISO Transport class4	Null
3. Network	ISO CLNS	Null
2. Datalink	IEEE 802.2 LLC type1	IEEE 802.2 LLC type3
1. Physical	IEEE 802.4 Token bus	IEEE 802.4 Token bus

그림-1 MAP의 구조

2. 모델링

Mini MAP Architecture의 모델링은 크게 어플리케이션 계층의 모델링과 데이터링크 계층의 모델링, 물리적계층의 모델링, 그리고

네트워크의 사용자의 모델링으로 나눌수 있다. 그러나, 실제적으로 MiniMAP Architecture의 성능은 하위 두 계층에 크게 의존하고 있어, 본 논문에서는 어플리케이션 계층을 모델링에서 제외하고, 데이터링크 계층이하만을 고려하여 모델링을 하였다.

데이터링크 계층의 서비스는 Acknowledged Datagram 서비스 방식이다. 이 방식은 패킷을 수신한 스테이션이 response 프레임을 전송하도록함으로써 보다 믿음직한 서비스를 제공한다. 따라서 소프트웨어적인 오버헤드를 DL이라 할 때 데이터링크 계층의 모델링은 데이터링크 헤더와 인식 프레임의 전송기능을 모델링함으로써 이루어진다.

물리적 계층의 모델은 토큰캐싱서비스의 모델이 되며 가장 복잡한 구조를 갖는다. 그림-2는 IEEE 802.4 토큰캐싱서비스의 유한상태 다이어그램(finite state diagram)이다[1].

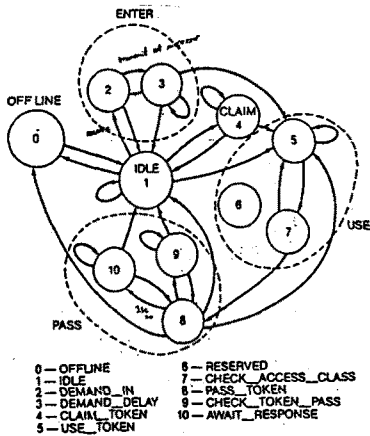


그림-2. 토큰캐싱서비스의 유한상태 다이어그램

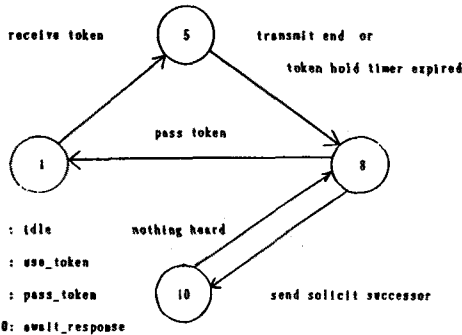


그림-3 간략화된 상태 다이어그램

그림-2의 상태 다이어그램은 여러의 검출 및 복구기능까지를 포함한 것이다. 그러나 정상상태에서의 성능을 해석하는 데는 아래와 같은 몇 가지 가정을 함으로써 처음의 상태 다이어그램을 그림-3과 같이 간략화 시킬 수 있다. 다음은 도입된 가정이다.

- 전송매체의 여러 발생음이 매우 낮아서 여러의 발생을 무시할 수 있다. 따라서 여러의 처리에 관계되는 상태(state)를 상태 다이어그램에서 제거할 수 있다.

- 논리적인 링(logical ring)에 새로 가입하거나 탈퇴하는 경우가 매우 적다. 따라서 이의 처리에 따른 오버헤드를 무시할 수 있다.

- 모든 프레임이 동일한 우선순위를 갖는다  
네트워크 사용자는 일반적으로 도착하는 패킷의 수와 패킷의 크기시켜 모델링 되어지며, 이는 네트워크의 사용목적에 따라 달라지게 된다. 실시간 시스템에 응용된 네트워크의 가장 큰 특징은 정보의 처리에 시간 제한이 존재하는 것이며 그 값은 약 50ms 정도이다. 이외에 실시간 네트워크에서 볼 수 있는 특징들은 다음과 같은 것들이 있다.

- 발생하는 패킷(packet)의 크기가 대부분 일정하며, 그 크기 또한 작은 것이 대부분이다.
- 패킷의 발생 간격이 일정한 경우가 대부분이다.
- 제어나 감시목적에 쓰이는 경우에 패킷의 목적지가 몇 개의 스테이션에 집중되는 경우가 많다.

이러한 특징들을 고려하여 본 논문에서는 네트워크의 사용자를 다음과 같이 모델링 하였다.

- 사용자는 A, B 두 형태로 나뉘며  $N_A$ 는 A 형태의 사용자가 사용하는 스테이션의 수,  $N_B$ 는 B 형태의 사용자가 사용하는 스테이션의 수도한다. 이때 이들 사이에는 다음식이 성립한다.

$$N_A = \alpha N \quad (0 < \alpha < 1), \quad N = N_A + N_B$$

- A 형태의 사용자는 평균  $L$  (p/sec)의 Poisson 분포를 가지며 도착하고, B 형태의 사용자는 평균  $\beta L$  (p/sec)의 Poisson 분포를 가지고 도착한다. ( $0 < \beta$ )

- 패킷의 크기는  $M$  bytes로 고정되어있다.

### 3. 토큰회전시간

토큰이 현재 토큰을 소유하고 있는 스테이션에서 다음 스테이션으로 넘겨지는 것은 서비스할 패킷이 없거나, token\_hold\_timer가 종료되었을 경우에 발생한다. 그러므로 token\_hold\_timer가 충분히 큰 값을 갖는다고 가정하면 token을 받기전 까지 도착한 패킷은 모두 서비스를 받게되어 exhausted 서비스 방식이 된다. 그러므로  $T_r$ 을 토큰회전시간이라하면 다음의 식이 성립한다.

$$T_r = T_r \text{ 동안 도착한 패킷의 서비스시간} + \text{Ring matenance overhead} \quad (3.1)$$

$$= T_i + T_o \quad (3.2)$$

식 (3.2)의 항을 차례대로 구해보면 다음과 같다.

A 형태의 스테이션에는  $T_r$  동안 평균  $L \cdot T_r$  개의 패킷이 도착하게되며 B 형태의 스테이션에는  $T_r$  동안 평균  $\rho \cdot L \cdot T_r$  개의 패킷이 도착하게된다. 또한 도착한 패킷은 protocol에 따른 오버헤드가 불계되고 전달을 확인하기위한 response 프레임을 수신해야

하므로 하나의 패킷을 서비스하는데 걸리는 시간  $x$  는 식 (3.3) 과 같이 주어진다.

$$x = \frac{8}{C} (M + 26) + x_0 + T_s \quad (3.3)$$

C: channel speed (5 Mbps)

이 때  $x_0$  는 response 프레임의 서비스에 걸리는 시간이고  $T_s$  는 slot time으로써 식 (3.4)와 같이 주어진다[11].

$$T_s = 2(D_L + D_p) + \text{safety margin} \quad (3.4)$$

$D_L$ : 스테이션 지연시간

$D_p$ : 전파지연시간

그러므로  $T_i$  은 식 (3.5)와 같이 주어진다.

$$\begin{aligned} T_i &= N_A \cdot L \cdot Tr \cdot x + N_B \cdot \beta \cdot L \cdot Tr \cdot x \\ &= (N_A + \beta N_B) \cdot L \cdot Tr \cdot x \end{aligned} \quad (3.5)$$

다음  $T_0$  를 살펴보면 보른은 실제로 보른 프레임의 전송이 종료된후 전파지연시간이 지난후에야 다음스테이션으로 전달된다. 또한 가장 낮은 어드레스를 갖는 스테이션을 제외하면 inter\_sollicit\_successor\_counter(ISSC)의 값이 0이 될 때마다 한 번씩 slot time만큼의 window를 개방하게 된다. 그리고 가장 낮은 어드레스를 갖는 스테이션은 slot time의 두 배 크기를 갖는 window를 개방하게 된다. 그러므로 한 번의 보른회전시간 동안에 고려해야할 오버헤드는 식 (3.6)으로 주어진다.

$$T_0 = N(x_0 + D_p) + \frac{T_s}{ISSC} (N + 1) \quad (3.6)$$

$x_0$ : 보른 프레임 전송시간

따라서 식 (3.5), (3.6) 을 이용하여 정리하면 보른회전시간  $Tr$  는 식 (3.7)과 같이 주어진다.

$$Tr = \frac{T_0}{1 - (N_A + \beta N_B) \cdot L \cdot x} \quad (3.7)$$

#### 4. 대기시간

대기시간은 스테이션에 도착한 패킷이 서비스를 받기까지 기다려야 하는 시간이다. 먼저 스테이션에 대기하고 있는 패킷이 하나도 없는 경우에 스테이션에 도착한 패킷의 경우를 살펴보자. 만일 도착과 동시에 스테이션이 보른을 넘겨받았다면 패킷의 대기시간은 0 이된다. 그러나 스테이션이 도착과 동시에 다른 스테이션으로 보른을 넘겨 주었다면 대기시간은  $Tr$  이 된다. 그러므로 이 경우에 패킷의 평균대기시간은  $0.5 Tr$  가 될 것이다. 다음 스테이션에 1 개의 패킷이 대기하고 있었다고 가정하면 새로 도착한 패킷의 대기시간  $W$  는  $0.5 Tr + x$  가 될 것이다. 그

러므로 스테이션에  $k$  개의 패킷이 존재하는 경우에 새로 도착하는 패킷의 대기시간은 식 (4.1)로 주어진다.

$$W = 0.5 Tr + k x \quad (4.1)$$

이 때 패킷의 도착이 Poisson 분포를 갖으므로 A 형태의 스테이션에서의 평균대기시간  $W_A$  와 B 형태의 스테이션에서의 평균대기시간  $W_B$  는 각각 (4.2), (4.3)의 식으로 주어진다.

$$W_A = \sum_{k=0}^{\infty} \frac{\beta^{k+1} (LTr)^k}{k!} (0.5 Tr + kx) \quad (4.2)$$

$$W_B = \sum_{k=0}^{\infty} \frac{\beta^{k+1} (\beta LTr)^k}{k!} (0.5 Tr + kx) \quad (4.3)$$

(4.2), (4.3) 식을 전개하면 식 (4.4), (4.5)를 얻는다.

$$W_A = 0.5 Tr + L \cdot Tr \cdot x \quad (4.4)$$

$$W_B = 0.5 Tr + \beta \cdot L \cdot Tr \cdot x \quad (4.5)$$

#### 5. 성능해석

성능을 알아보기 위하여 다음과 같은 실시간 네트워크를 가정하였다.

- 전체 스테이션중 10 % 가 A 형태의 스테이션으로 제어와 감시를 위하여 사용되며 나머지 스테이션은 데이터의 수집과 독립된 작업을 행한다. 이 때 B 형태 스테이션에서의 패킷 발생율은 A 형태 스테이션에서의 패킷 발생율의 1/2 이다.

식 (3.7)과 식 (4.4)를 이용하여 A 형태의 스테이션에서의 보른회전시간과 대기시간을 그림 -4 ~ 그림 -7 에 나타내었다. 이 때 사용된 값들은  $\alpha = 0.1$ ,  $\beta = 0.5$ ,  $D = 10 \mu s$  이며  $N, L, M, D$  등은 그림에 나타난 값들이다. safety margin은 2ps로 하였다.

그림들을 통해서 주어진 조건하에서 스테이션의 수, 패킷의 도착률 등이 대기시간과 보른회전 시간에 큰 영향을 준다는 사실을 관찰할 수 있다. 그리고 [3]-[10]에서 고려되지 않았던 스테이션에서의 소프트웨어적인 지연시간도 매우 큰 영향을 준다는 사실을 관찰할 수 있다.

그러나 전체적으로 볼 때  $N < 50$ ,  $L < 40$  p/sec 정도이면  $\alpha$  나  $\beta$  에 관계없이, 모델링에서 고려되지 않았던 어플리케이션 계층의 오버헤드를 비롯한 네트워크 인터페이스의 구현에 따른 오버헤드를 고려하더라도 안정된 동작을 한다는 사실을 알 수 있다. 이는 식 (3.7)에서 유도되는 네트워크의 안정성에 관한 식에 의해서도 나타난다. 식 (3.7)이 의미를 가지기 위해서는 분모가 0 보다 커야한다. 그러므로 도착률  $L$  은 다음식을 만족해야한다.

$$L < \frac{1}{(N_A + \beta N_B) x} \quad (5.1)$$

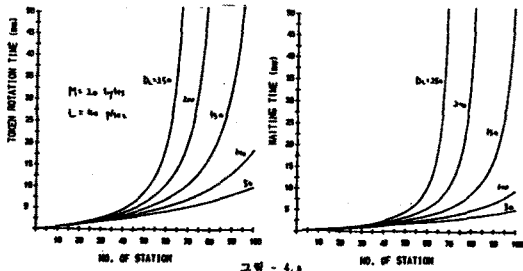


그림 - 4. a

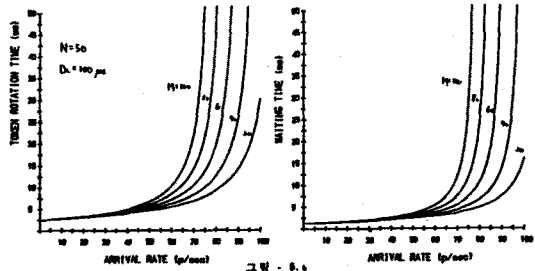


그림 - 6. a

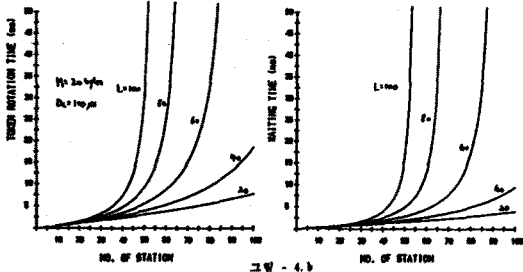


그림 - 4. b

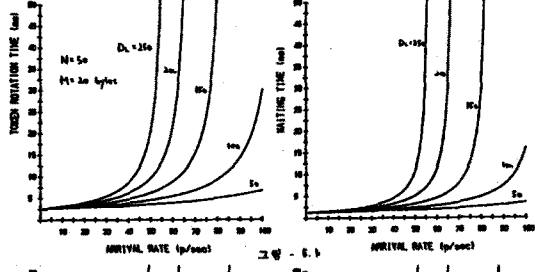


그림 - 6. b

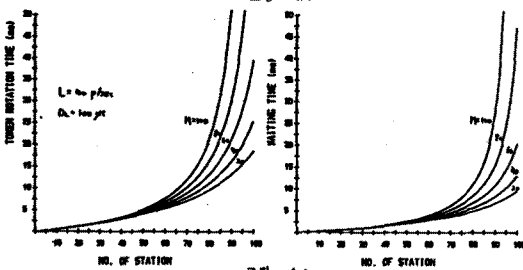


그림 - 4. c

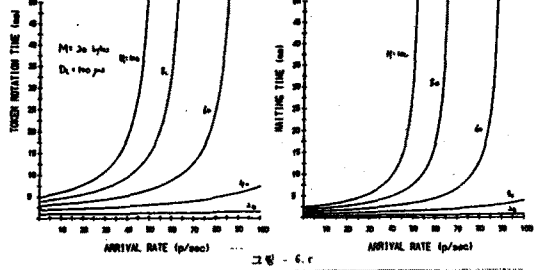


그림 - 6. c

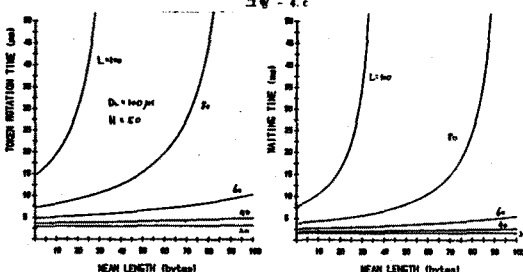


그림 - 5. a

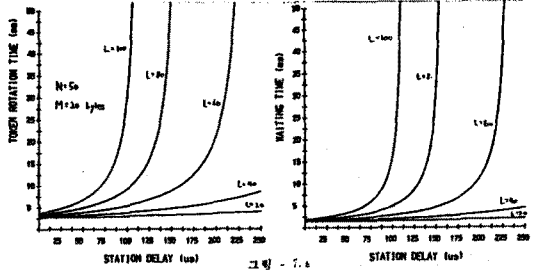


그림 - 7. a

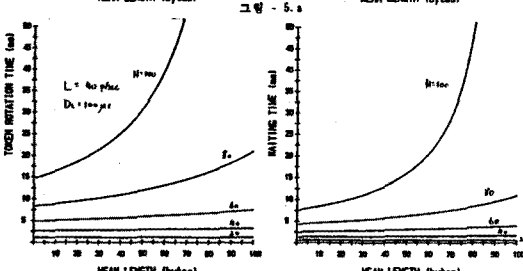


그림 - 5. b

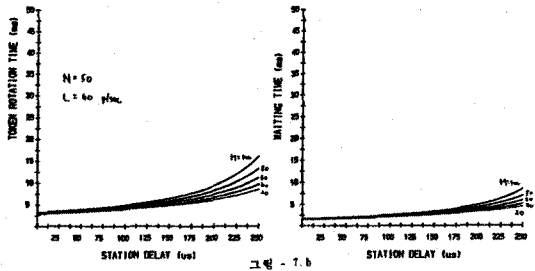


그림 - 7. b

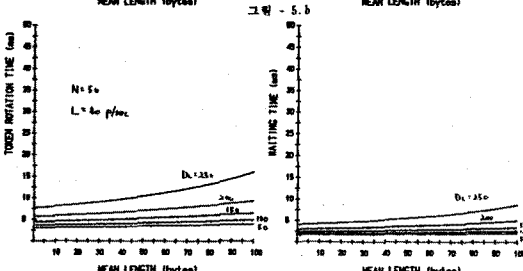


그림 - 5. c

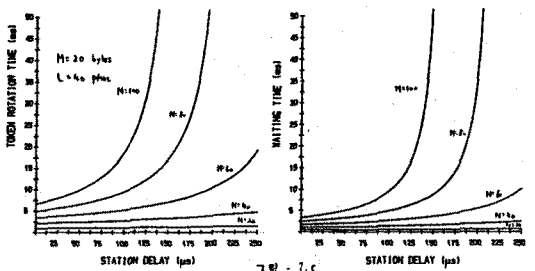


그림 - 7. c

참 고 문 헌

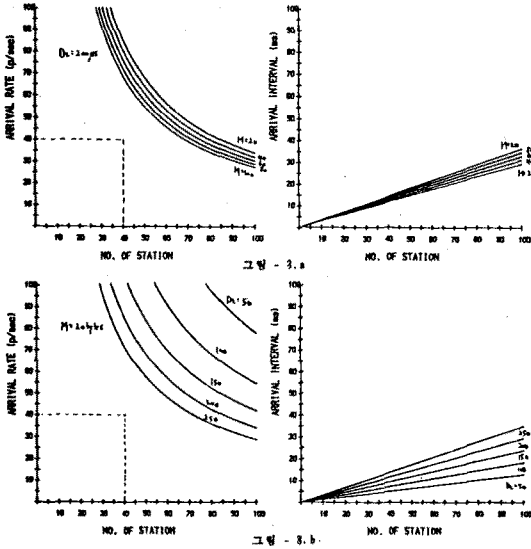


그림 - 3 은 네트워크가 안정하게 동작하기 위한 패킷의 도착률과 N, D 사이의 관계를 (5.1)식을 이용하여 나타낸 것이다. 앞의 그림 - 4 ~ 그림 - 7 에서와 같이  $N < 50$ ,  $L < 40$  p/sec 정도이면 네트워크 인터페이스의 구현에 따른 오버헤드를 고려한다 하더라도 원하는 실시간 응답을 얻을 수 있음을 알 수 있다.

6. 결론

본 논문에서는 실시간 응용시에 Mini MAP 구조를 사용한 네트워크의 성능을 해석적 방법으로 알아보았다. 분석을 통하여 모든 회전시간에 관한 식과 대기시간에 관한 식을 유도하였으며 그를 이용하여 네트워크가 안정하게 동작할 수 있는 조건을 구하였다. 유도된 식들을 통하여 스테이션에서의 소프트웨어적인 지연시간의 영향이 매우 큼을 알 수 있으며 실시간 응용에서와 같이 패킷의 크기가 작은 경우는 패킷길이의 영향이 그리 크지 않다는 사실을 알 수 있었다. 그리고 스테이션 지연시간들을 모두 고려하더라도 실시간 시스템에 응용된 네트워크와 같이 스테이션 수가 적은 경우 ( $N < 50$ )에는 충분히 원하는 실시간 응답을 얻을 수 있음을 알 수 있었다.

- [1] MAP Task Force, Manufacturing Automation Protocol, version 2.2, General Motors, 1986.1.
- [2] CONTROL ENGINEERING Special edition : Manufacturing Automation Protocol, Oct. 1986
- [3] Juan R. Pimentel, "Performance evaluation of MAP networks", IECON85, 1985.
- [4] Werner Bux, "Local-Area Subnetworks : A Performance Comparison", IEEE Tran. on COMMUNICATIONS, vol.COM-29, No.10, Oct. 1981.
- [5] R. Cherkuri, L. Li and L. Louis, "Evaluation of Token passing Schemes in Local Area Networks", Proceedings. Computer Networking Symposium, 1983.
- [6] A. S. Sethi and T. Saydan, "Performance Analysis of Token ring Local Area Networks", 3th Conference on Local Computer Networks, Oct. 1984.
- [7] M.E. Ulug, "Calculation of Waiting Times for a Dynamic Token Passing Bus", Proceedings Computer Networking Symposium, 1984.
- [8] M.E. Ulug, "Comparison of Token Holding Time Strategies for a Static Token Passing Bus", Proceedings Computer Networking Symposium 1984.
- [9] H. Okada, "Comparative Evaluation of Token Ring and CSMA/CD Medium - Access Control Protocols in LAN Configurations", Proceedings Computer Networking Symposium, 1984.
- [10] 이 도영, "보른 버스 근거리 통신망의 성능분석", 한국과학기술원 석사학위논문, 1985.
- [11] IEEE 802.4 Token Passing Bus Access Method and Physical Layer Specifications revision E, 1983.
- [12] ANSI/IEEE Standard 802.2, 1986.1.