

IEEE 802.4 토큰 패싱 버스 프로토콜의 성능에 관한 시뮬레이션 연구

(Simulation Study on the Performance of the IEEE 802.4 Token Passing Bus Protocol)

林 東 民*, 李 愧 洙*, 殷 鍾 官*

(Dong Min Lim, Hwang Soo Lee and Chong Kwan Un)

要 約

본 논문에서는 프로토콜의 시뮬레이션 모델을 통하여 IEEE 802.4 토큰 패싱 버스 프로토콜의 성능을 분석하였다. 정상상태 뿐만 아니라 초기상태, 과도상태, *fault recovery* 상태에서의 프로토콜 성능을 분석하기 위해 프로토콜의 기능을 다섯 개의 프로세스로 나누어 구성하였으며 각 프로세스는 프로토콜에서 정의된 파라미터들에 따라 프로토콜의 동작을 실제적으로 수행한다. 프로토콜의 시뮬레이션을 통한 연구로부터 망의 성능을 좌우하는 프로토콜 파라미터들을 알 수 있었으며 *priority*가 구현된 경우 토큰에 대한 파라미터 값의 선택이 이 프로토콜의 적절한 사용을 위하여 매우 중요함을 알 수 있었다.

Abstract

In this paper, we analyze the performance of the IEEE 802.4 token passing bus protocol through a simulation model of the protocol. In order to analyze the performance of the protocol in the initialization, transition and fault recovery states as well as in the steady state, the protocol functions are divided into five processes each of which can effectively simulate protocol behaviors according to the variations of protocol parameters. From the simulation study, we obtain protocol parameters which severely influence the protocol performance and find out that proper selection of the protocol parameter values for token passing is very important to obtain good performance of the protocol when the priority scheme is used.

I. 서 론

IEEE 802.4 토큰 패싱 버스 프로토콜⁽¹⁾은 여러 독자

적인 스테이션이 공유하는 하나의 전송매체를 정해진 순서에 따라 사용하도록 설계되었다. 이 프로토콜은 international standard organization(ISO)이 제정한 open system interconnection(OSI)의 physical, data link layer에 해당한다. IEEE 802 위원회가 제안한 local area network(LAN) 표준 프로토콜 세 가지 중에서, 802.4 토큰 패싱 버스 프로토콜은 정보의 실시간 처리를 요하는 공장 환경에서의 통신에 적합하다.

*正會員, 韓國科學技術院 電氣 및 電子工學科
(Dept. of Electrical Eng., KAIST)

接受日字: 1988年 10月 17日

(※ 본 연구는 한국과학재단 목적기초 연구비 지원에 의한 것임.)

LAN을 이용한 통신망으로 구성된 시스템의 실시간 응용을 위해서는 망에 대한 스테이션의 최대 access delay가 한정된 값을 가져야 한다. 일반적으로 토큰 패싱 방식은 한 스테이션이 토큰을 가지는 시간을 제한함으로써 실시간 응용에 적합한 특성을 지닌다.^{[2],[3]} 공장에서의 통신은 traffic의 양이 많고 변화가 적으며 실시간 처리를 요하는 정보가 많다. 더욱 문제가 되는 것은 이와같은 실시간 처리를 요하는 컨트롤 정보의 지연으로 시스템이 파괴될 수도 있다는 점이다. 이러한 환경에서는 carrier sense multiple access with collision detection(CSMA/CD) 방식^[4]은 정보의 전달시간이 통제적으로 주어지기 때문에 부적당하며 주어진 시간내에 망에 access가 가능한 토큰 패싱 방식이 이용된다. 또한 버스 형태의 망 구성은 공장의 생산라인 형태에 잘 적용될 수 있으며 전체 망에 큰 영향을 주지않고 스테이션을 추가설치 또는 제거할 수 있는 장점이 있다. 이러한 이유로 토큰 패싱 버스 방식은 general motors (GM)의 manufacturing automation protocol(MAP)의 physical, data link layer 프로토콜로 채택되었으며 이 MAP을 이용한 LAN이 실제로 공장자동화에 이용되고 있다.

이러한 표준 프로토콜은 복잡한 구조를 가지므로 수학적 해석 모델로는 성능을 분석하는데 한계가 있다. 이러한 경우에는 프로토콜 세부동작의 모델링이 가능한 시뮬레이션을 이용한 성능분석 방법이 효과적이다. Pimentel,^[5] Loyer와 Kolton^[6]은 시뮬레이션을 통하여 정상상태에서 여러가지 프로토콜 파라미터가 성능에 미치는 영향에 대하여 연구하였다. 그러나 이 프로토콜에서 필수적 기능인 링의 구성, 링의 관리, fault recovery시의 프로토콜 동작 및 성능에 대해서는 프로토콜의 복잡성으로 인해 연구결과가 많지 않다.

본 논문에서는 프로토콜의 중요기능들을 각각 프로세스로 모델링함으로써 프로토콜의 실제 동작을 구현하였으며 프로토콜의 정상상태에서 뿐만아니라 초기상태와 과도상태에서의 프로토콜의 동작과 여러 파라미터 변화에 따른 프로토콜의 성능을 연구하였다.

II. 토큰 패싱 버스 프로토콜의 개요

토큰 패싱 버스 프로토콜은 IEEE에 의해 채택된 LAN 표준 프로토콜 중의 하나이다. 토큰 패싱 버스 프로토콜의 전송매체는 여러 스테이션이 하나의 라인에 붙어있는 외형적인 버스 형태이다. 정상상태 동작에서 토큰(전송매체를 사용할 수 있는 권리)은 스테이션 address가 감소하는 순서에 따라 다음 스테이션으로 넘겨진다. 가장 낮은 address를 가진 스테이션은 가장 높은 address의 스테이션에 토큰을 넘

겨 줌으로써 외형적인 버스위에 그림 1과 같이 logical ring을 구성한다.

토큰은 전송매체를 사용할 수 있는 권리를 말한다. 토큰은 실제적으로 control frame중의 하나이다. 한 스테이션이 토큰 frame을 전체 버스에 전송하면 버스의 모든 스테이션은 수신된 토큰 frame의 destination address를 조사한다. 토큰 frame이 자신에게 보내진 것을 확인한 스테이션은 그 순간부터 전송매체를 사용할 권리를 갖게된다.

정상 상태의 동작은 데이터 전송과 토큰 패싱으로 나눌 수 있다. 다른 필수적이고 복잡한 기능은 초기에 링을 구성하고 유지하는 것과 fault recovery 이다.

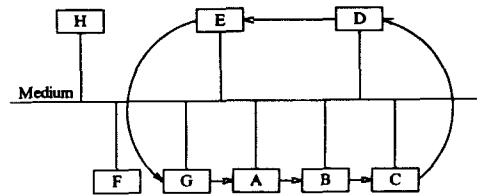


그림 1. Physical bus위에 구성된 logical ring
Fig. 1. Logical ring on physical bus.

III. 시뮬레이션 모델링

시뮬레이션 모델링은 수학적 해석 모델의 해석이 어려운 경우나 프로토콜의 세부적인 동작과 과도상태의 동작을 분석하기 위하여 사용된다. Priority가 구현된 IEEE 802.4 토큰 패싱 버스 프로토콜은 수학적 해석 모델로 해석하기가 쉽지 않다.^[7] 또한 이 프로토콜에서 필수적인 초기에 링을 구성하고 유지하는 기능과 fault recovery등에 대한 성능분석은 시뮬레이션 모델링을 통해서 가능하다. 본 논문에서는 프로토콜의 모델링에 systematic incremental modeling 방식^{[8],[9]}을 이용하였다. Incremental modeling 방식은 시스템의 작은 부분에 대한 모델을 구성하고 검증한 후 이를 바탕으로 전체 시스템의 모델을 구성하는 것을 말한다. Systematic modeling은 functional modeling 단계와 operational modeling 단계로 나눌 수 있으며 functional modeling은 실제 시스템의 기능에 대한 모델링이며 operational modeling은 정해진 목적에 따라 시스템 동작의 성능을 측정하기 위한 모델링을 말한다.

토큰 패싱 버스 프로토콜은 내부기능에 따라 여러 부분으로 나눌 수 있다. 각 부분은 서로 논리적으로

관련이 있는 동작을 수행한다. 이 논문에서는 simulation package인 SIMULA¹⁰⁾에서 제공되는 프로세스로 각 부분을 모델링 하였다. 여기서 프로세스는 운영체제에서 쓰이는 일반적인 프로세스 개념과 유사하다. SIMULA에서는 queue를 구현하는데 쓰이는 SIMSET class와 process description, scheduling, synchronization, time flow mechanism 등을 수행하는 SIMULATION class를 이용할 수 있다.

1. Physical Layer와 전송매체의 모델링

Physical layer와 전송매체에서 일어나는 세부동작은 고려하지 않았고 기능별로 나누어진 각 부분을 delay unit로 모델링 하였으며 신호의 중첩현상도 고려하였다. 잡음, 외부적인 간섭현상, 케이블에서의 신호의 감쇄와 왜곡현상은 고려하지 않았다.

버스에 연결된 스테이션의 실제 위치는 head-end remodulator 또는 terminator로부터의 거리로 나타내었다. 그림 2에 이러한 모델링을 도시하였다.

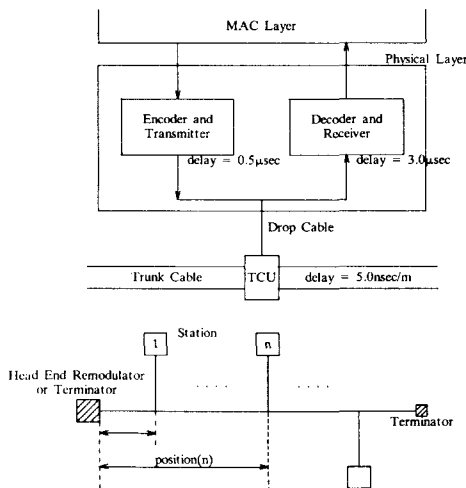


그림 2. Physical layer와 전송매체의 모델링
Fig. 2. Modeling of the physical layer and medium.

2. Medium Access Control(MAC) Layer의 모델링

MAC layer의 시뮬레이션 모델은 그림 3과 같이 서로 정보를 주고 받는 다섯가지 종류의 프로세스로 이루어져 있다. 각 스테이션당 interface machine (IFM) 프로세스와 access control machine(ACM) 프로세스가 각각 하나씩 create되고 logical ring이 구성된 후 네가지 access class에 대해 LLC_gen 프로

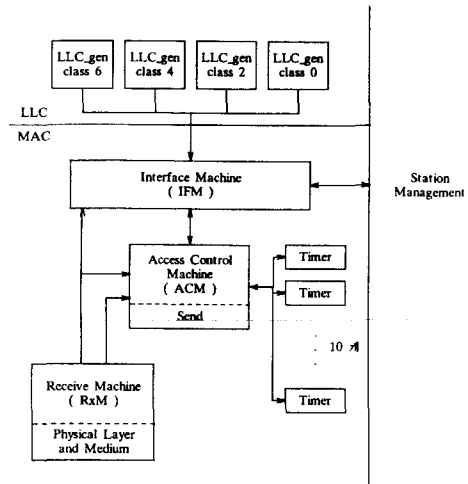


그림 3. MAC layer의 프로세스 모델
Fig. 3. Process model of the MAC layer.

세스가 하나씩 create된다. Timer 프로세스는 모두 열 개이며 slot 단위로 동작하는 것이 다섯개, octet 단위로 동작하는 것이 다섯 개이다. Timer 프로세스는 프로토콜에 의해 정해진 순간에 ACM에 의해 create되며 timeout시 terminate된다. Receive machine(RxM) 프로세스는 frame 전송시 마다 create된다. 한 스테이션이 ACM의 "send" procedure를 통하여 frame을 전송하면 모든 스테이션의 RxM 프로세스가 create된다. Frame의 전송시간이 지나면 RxM 프로세스는 수신된 frame의 상태에 따라 IFM 프로세스와 ACM 프로세스에 상황을 전한 후 terminate된다. Main program의 흐름도를 그림 4에 나타내었다.

1) RxM 프로세스

반대 방향으로 진행되는 두 개의 전기적 신호는 어느 순간 동시에 한 곳을 지나게 된다. 이와같은 위치에 있는 스테이션은 의미없는 신호를 수신하게 된다. 이와같은 신호를 "noise-burst"라고 부르기로 한다. Cable propagation delay로 인한 이러한 신호의 중첩현상을 physical layer와 전송매체의 modeling시 고려하였으며 프로그램의 효율을 위하여 RxM 프로세스 속에 포함시켰다.

ACM의 "send" procedure에 의해 create된 RxM 프로세스는 신호가 중첩되어 있는지를 조사한다. Frame 전송시간만큼 기다린후(hold되어 있음) 버스에 더 이상의 신호가 없으면 frame의 도착을 ACM 프로세스에 알린다. LLC 데이터 frame인 경우에는

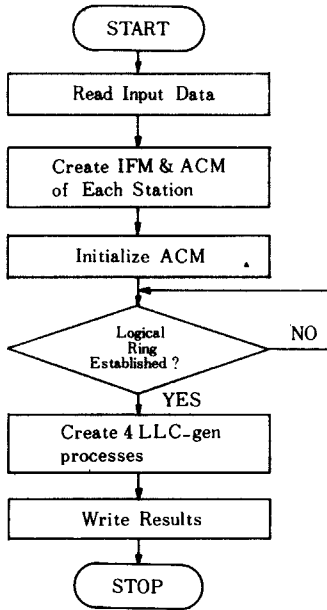


그림 4. Main program의 흐름도
Fig. 4. Flow chart for the main program.

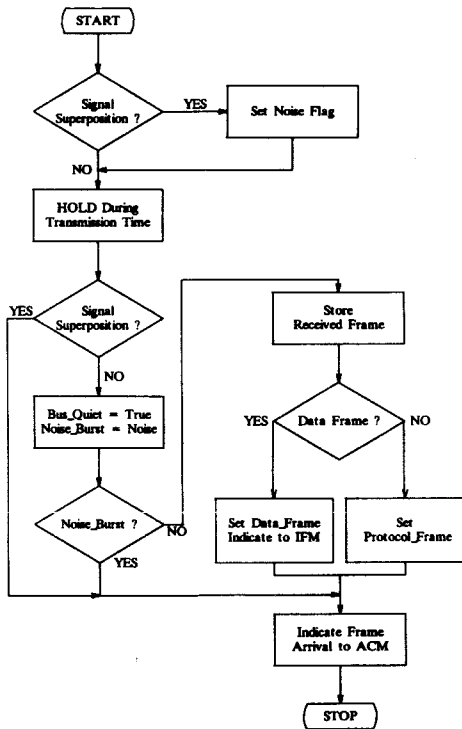


그림 5. RxM 프로세스의 흐름도
Fig. 5. Flow chart for the RxM process.

IFM 프로세스에도 frame의 도착을 알린다. RxM 프로세스의 흐름도를 그림 5에 나타내었다.

2) Timer 프로세스

Timer는 slot 단위와 octet 단위로 동작하는 두 가지 종류가 있다. ACM 프로세스에서 사용되는 timer는 모두 10개이며 ACM initialization시 create 된다.

- (1) bus-idle-timer (slot 단위)
- (2) claim-timer (slot 단위)
- (3) response-window-timer (slot 단위)
- (4) contention-timer (slot 단위)
- (5) token-pass-timer (slot 단위)
- (6) token-rotation-timer(4) (octet 단위)
- (7) token-rotation-timer(2) (octet 단위)
- (8) token-rotation-timer(0) (octet 단위)
- (9) token-rotation-timer (-2) (octet 단위)
- (10) token-holding-timer (octet 단위)

Timer 프로세스에는 다음과 같은 세가지 procedure가 있다.

- start (quantum)

Timer의 동작시간(quantum)을 정해서 동작시킨다.

- Residual-value

Timer의 종료때까지 남은 시간을 알려준다.

- Expired

Timer의 종료를 알려준다.

Timer 프로세스는 종료되기 전에 다시 재동작이 가능하다. Slot 단위의 timer 프로세스는 종료시 이 사실을 ACM 프로세스에 알려준다. 그림 6은 Timer 프로세스의 흐름도이다.

3) ACM 프로세스

ACM 프로세스는 11개의 내부 state를 가진다. State의 처음부분에서는 여러가지 내부 또는 외부 변수를 바탕으로 정해진 procedure를 수행한다. 그림 7은 ACM 프로세스의 state 내부동작의 흐름도이다. Procedure의 수행시간(state_time)은 다음과 같이 정하였다.

$$\begin{aligned}
 \text{state_time} &= \frac{\text{station_delay}}{10} & (1) \\
 &= \frac{\text{slot_time-margin}}{2 \times 10}
 \end{aligned}$$

Station_delay는 스테이션이 PHY_symbol을 받아서 다시 내보내는데 걸리는 시간이다. 프로토콜에서 이러한 응답에 걸리는 최대 state의 수는 10개이다.

Frame의 전송을 모델링한 "send" procedure는 프로그램의 효율을 높이기 위하여 ACM 프로세스에 포함시켰다. 이 procedure는 frame을 보내는 순간부터

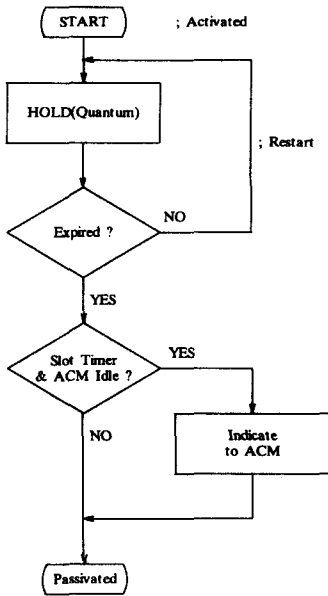


그림 6. Timer 프로세스의 흐름도
Fig. 6. Flow chart for the timer process.

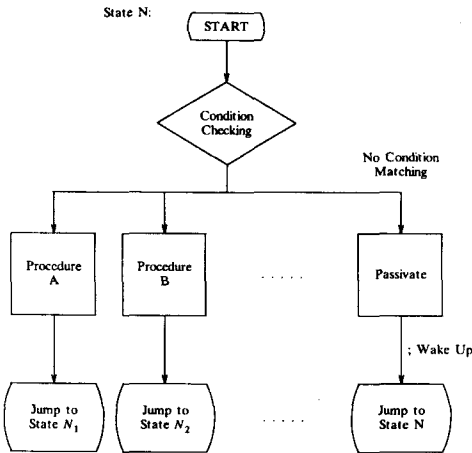


그림 7. ACM 프로세스 내부 state의 흐름도
Fig. 7. Flow chart for the internal state of ACM process.

받는 순간까지의 total delay를 구하여 이 시간이 지나면 RxM 프로세스를 create시킨다.

$$\begin{aligned} \text{total delay} = & \text{propagation delay} \\ & + \text{transmitter modem delay} \\ & + \text{receiver modem delay} \end{aligned} \quad (2)$$

4) IFM 프로세스

IFM 프로세스에는 여러가지 event가 발생할 수 있다. 본 논문에서는 세 가지 event를 고려하였다.

- (1) Logical link control(LLC) connectionless data transfer service를 위한 service request primitive
- (2) 전체 MAC sublayer를 reset 하거나 MAC entity의 MAC address를 선택하기 위한 service primitive
- (3) frame의 수신

이러한 event를 위한 event queue가 필요하며 LLC 데이터를 위한 네개의 queue가 있어야 한다.

그림 8에 LLC, station management function (SMF), RxM 등이 event를 처리하는 과정을 나타내었고 그림 9에 IFM 프로세스의 흐름도를 나타내었다.

5) LLC_gen 프로세스

LLC_gen 프로세스는 LLC protocol data unit (PDU)를 Poisson 분포에 따라 만들어 낸다. LLC PDU는 MA_DATA.req event를 통하여 IFM에 넘겨진다. 그림10에 LLC_gen 프로세스의 흐름도를 나타내었다.

IV. 시뮬레이션의 결과 및 분석

일반적으로 LAN의 성능은 stability, throughput, delay 등으로 표시된다.^[1] 이 논문에서는 주로 throughput에 대한 delay로 성능을 평가하였다. Delay에 대한 측정으로 waiting time을 선택하였다. Waiting time은 frame이 queue에 도착한 순간부터 전송이 시작되기 바로 직전까지의 시간을 말한다.

시뮬레이션 모델에서 사용한 파라미터는 다음과

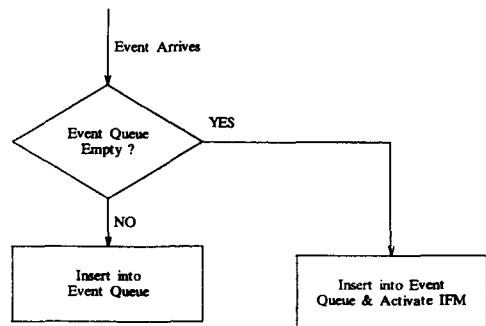


그림 8. IFM event의 처리과정
Fig. 8. IFM event handling procedure.

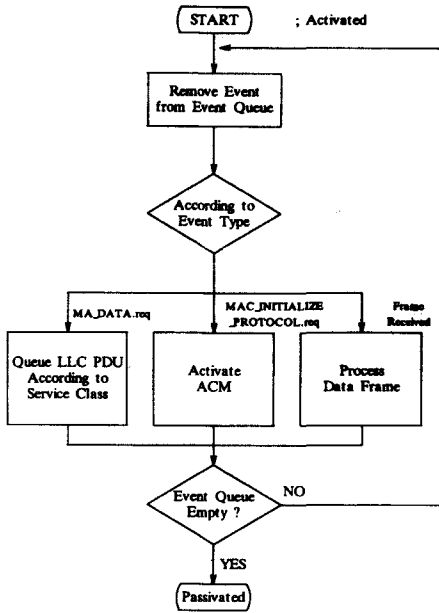


그림 9. IFM 프로세스의 흐름도
 Fig. 9. Flow chart for the IFM process.

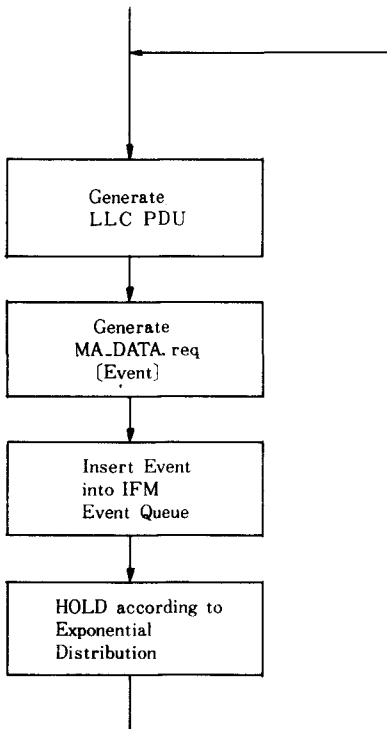


그림10. LLC_gen 프로세스의 흐름도
 Fig. 10. Flow chart for the LLC_gen process.

같다.^{[6],[11]}

Physical layer

- Data rate = 10Mbits/sec
- Cable propagation delay = 5nsec/m
- Delay at head-end remodulator = 7μsec
- Transmitter modem delay = 0.5μsec
- Receiver modem delay = 3μsec

MAC layer

- Length of preamble = 4bytes
- Length of address field = 6bytes
- Frame arrival = Poisson distribution

그밖에도 시뮬레이션시 정해야 하는 여러가지 파라미터는 다음과 같다.^[11]

- 스테이션의 수
- 스테이션의 위치
- MAC service data unit (SDU)의 길이
- "Response window"를 개방하는 빈도
- Token holding time
- Target rotation time
- Slot time
- ACM state 처리시간

시뮬레이션을 통하여 링의 구성, 링의 관리, fault recovery, 정상상태의 동작을 분석하였다. 프로토콜에서 제시된 동작이 수행되는 것을 확인한 후 파라미터 값의 변화가 성능에 미치는 영향을 알아보았다.

1. 링의 구성

Logical ring은 링을 구성하는데 참가하기를 원하는 스테이션 중 가장 높은 address의 스테이션이 먼저 토큰을 얻어 스테이션 address가 감소하는 순서로 스테이션이 링에 포함됨으로써 이루어진다. 링의 구성과정에서 스테이션의 address가 사용되어 링을 구성하는데 걸리는 시간은 스테이션 address에 크게 영향을 받는다. 스테이션 address가 정해지면 링 구성시간은 스테이션의 수와 slot time에 따라 변한다. 그림11과 그림12에서와 같이 링 구성시간은 slot_time과 스테이션의 수에 비례하는 관계를 가진다.

Address가 같은 두 개의 스테이션이 존재할 때는 동시에 두 개의 스테이션이 데이터를 전송하는 현상이 생길 수 있다. 이 경우에는 먼저 Idle 상태로 들어간 스테이션이 Off_line 상태가 되어 하나의 스테이션만이 존재하게 된다.

2. 링의 관리

망내의 각 스테이션은 logical ring의 구성원이 될 수도 있으며 필요에 따라 링에서 빠져나올 수도 있다. 링의 구성원이 되기를 원하는 스테이션은 토큰

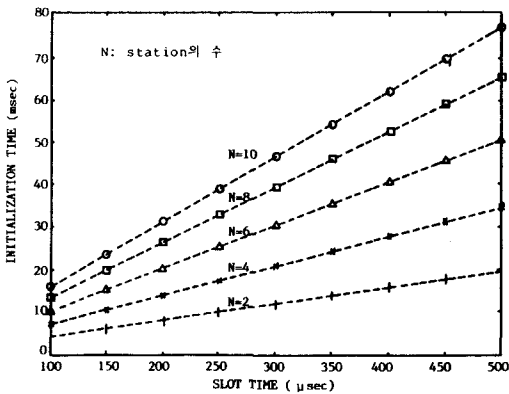


그림11. Logical ring의 구성에 걸리는 시간 (slot_time 변화에 따라)

Fig. 11. Logical ring establishment time. (vs. slot_time).

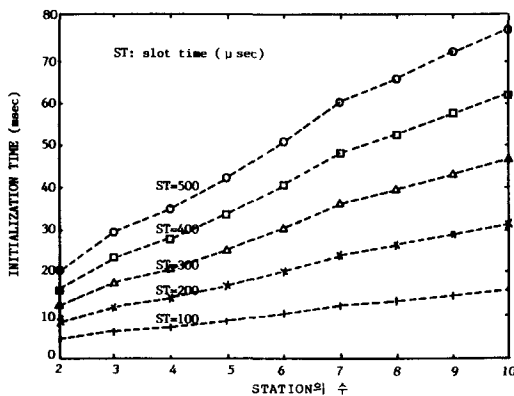


그림12. Logical ring의 구성에 걸리는 시간 (스테이션 수의 변화에 따라)

Fig. 12. Logical ring establishment time. (vs. number of station).

을 가지고 있는 스테이션이 링에 참가를 허용하는 frame을 전송할 때 이 frame에 응답을 보냄으로써 링의 구성원이 될 수 있다. 또한 각 스테이션은 자기에게 보내진 토큰에 응답하지 않으므로써 링에서 빠져나와 단지 frame을 수신하는 스테이션이 될 수 있다. 스테이션이 링에 참가되는데 걸리는 시간은 max_ring_maintenance_time, max_inter_solicit_count, 스테이션의 수에 영향을 받는다.

표 1 과 표 2 에서 알 수 있듯이 max_inter_solicit_

count가 증가함에 따라 또 max_ring_maintenance_time 이 감소함에 따라 링에 참가되는데 걸리는 시간이 커짐을 알 수 있다.

스테이션이 링에서 빠져나오는 것은 전체 망의 동작에 크게 영향을 주지 않으며 빠져나오는데 걸리는 시간도 별로 문제가 되지 않는다. 스테이션의 수가 여섯인 경우에 빠져나오는데 걸리는 시간의 평균은 4400μsec 정도였다.

표 1. 스테이션이 링에 참가되는데 걸리는 시간 (μsec) (max_inter_solicit_count 변화에 따라, 스테이션의 수=5)

Table 1. Ring insertion time(μsec). (vs. max_inter_solicit_count, 5 stations).

max_inter_solicit_count	time [worst case]
50	56228
100	120613
150	167825
200	277406
250	348508

표 2. 스테이션이 링에 참가되는데 걸리는 시간 (μsec) (max_ring_maintenance_time 변화에 따라, 스테이션의 수=5)

Table 2. Ring insertion time(μsec). (vs. max_ring_maintenance_time, 5 stations).

max_ring_maintenance_time	time
1000	22580.6
5000	4932.7

3. Fault Recovery

망에서는 여러가지 error가 발생할 수 있으며 회복가능한 error에 대하여서는 정상상태로의 회복이 요구된다. 본 논문에서는 아래에 열거된 상황에서의 프로토콜 동작을 살펴보았으며 정상상태로의 회복을 확인하였다.

- Lost token
- Multiple tokens
- Deaf station
- Duplicate station addresses

4. 정상상태의 동작

정상상태에서는 frame length, slot time, token holding time이 waiting time에 미치는 영향을 알아보았다. 그림13에서 나타나듯이 frame length가 길어지면 waiting time이 줄어드나 250-500byte에서는 거의 변화가 없었다. Waiting time은 slot time에 비례하는 관계가 있으며 traffic이 많은 경우에는 token holding time(THT)을 충분히 크게 해야함을 그림14와 그림15로부터 알 수 있다. 그림13과 그림14의 analysis 결과는⁽¹²⁾의 결과를 이용하였다.

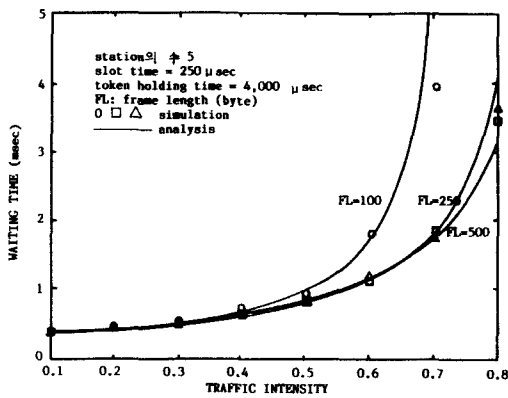


그림13. Frame의 길이가 frame의 waiting time에 미치는 영향(class 6 frame만이 존재하는 경우)

Fig. 13. Effect of the frame length on frame waiting time. (class 6 frame only).

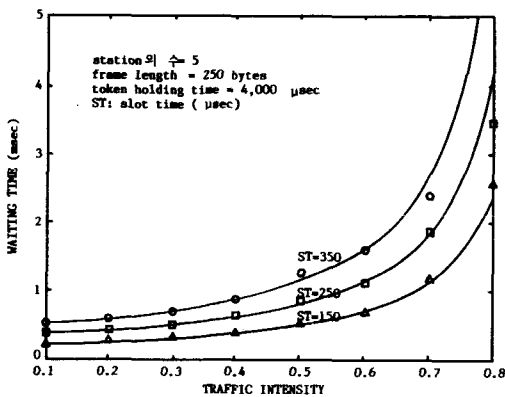


그림14. Slot time이 frame의 waiting time에 미치는 영향(class 6 frame만이 존재하는 경우)

Fig. 14. Effect of the slot time on frame waiting time. (class 6 frame only).

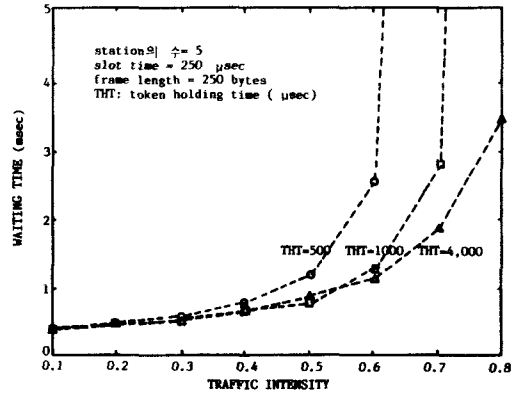


그림15. Token holding time이 frame의 waiting time에 미치는 영향(class 6 frame만이 존재하는 경우)

Fig. 15. Effect of the token holding time on frame waiting time. (class 6 frame only).

그림16, 그림17, 그림18을 비교하여 볼 때 target rotation time(TRT)은 waiting time에 크게 영향을 미치며 토큰 패싱 버스 프로토콜이 사용되는 상황에 따라 waiting time을 조절하는데 이용될 수 있다.

V. 결 론

본 논문에서는 IEEE LAN 표준 프로토콜의 하나인 토큰 패싱 버스 프로토콜의 성능을 시뮬레이션을 통하여 연구하였다. 프로토콜의 정상상태 동작 뿐 아니라 초기상태와 과도상태의 성능을 분석하기 위해 프로토콜을 여러가지 세부 기능으로 나누어 프로세스

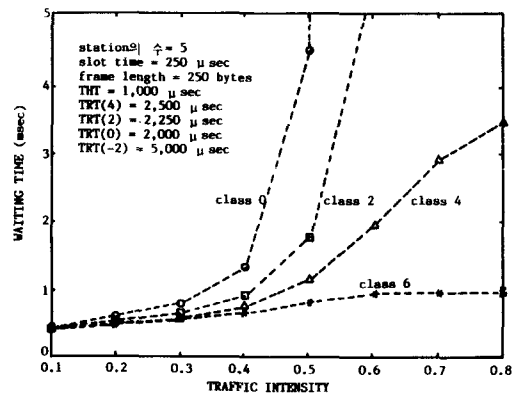


그림16. 각 access class frame의 waiting time
Fig. 16. Frame waiting time of each access class.

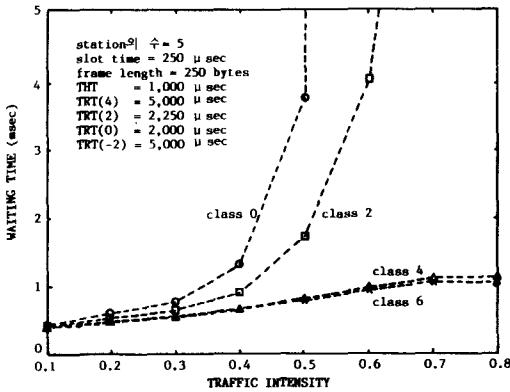


그림17. 각 access class frame의 waiting time
Fig.17. Frame waiting time of each access class.

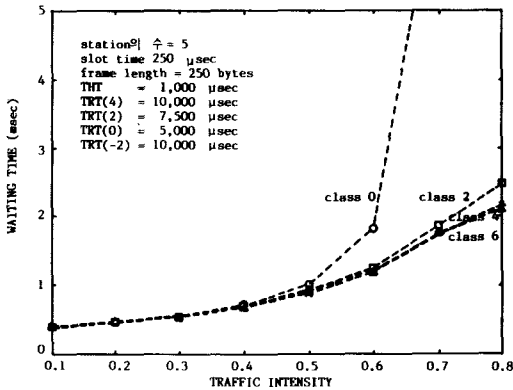


그림18. 각 access class frame의 waiting time
Fig.18. Frame waiting time of each access class.

로 모델링하였으며 프로토콜에서 정의된 파라미터가 프로토콜의 성능에 미치는 영향에 대해 연구하였다. 망의 초기상태에서 slot time과 스테이션 수의 변화에 따른 링 구성시간의 변화를 알아보았다. 링이 구성된 후 링 밖의 스테이션이 링에 첨가되는데 걸리는 시간은 max_inter_solicit_count와 max_ring_maintenance_time의 영향을 받으며 token rotation time이 증가하여도 일정한 시간 한도내를 유지하도록 여유있는 파라미터 값의 선택이 필요하다. 정상상태에서는 frame의 waiting time에 대한 frame의 길이, slot time, token holding time의 영향에 대하여 알아보았다. Frame의 waiting time은 frame의 길이가 길수록, slot

time이 작을수록, token holding time이 클수록 작아지는 현상을 보인다. 특히 token holding time의 경우 스테이션 간의 공정성이 크게 문제가 되지 않으면 값을 충분히 크게하는 것이 바람직하다. Priority scheme이 구현된 경우에는 낮은 traffic에서는 각 access class에 대한 waiting time에 큰 차이가 없으나 traffic이 증가함에 따라 낮은 access class에 대한 target rotation time을 적절히 증가시킴으로써 전체적인 성능을 높일 수 있다.

参 考 文 献

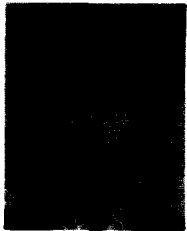
- [1] IEEE Std 802.4-1985, Local area network standard-Token passing bus access method and physical layer specifications.
- [2] W. Stallings, Local networks, Macmillan Inc., New York, N.Y., 1984.
- [3] C.K. Miller and D.M. Thompson, "Making a case for token passing in local networks," Data Comm., pp. 79-88, Mar. 1982.
- [4] IEEE Std 802.3-1985, Local area network standard-Carrier sense mutple access with collision detection (CSMA/CD) access method and physical layer specifications.
- [5] J.R. Pimentel, "Performance simulation of the IEEE token bus protocol," Workshop on analytic and simulation modeling of IEEE 802.4 token bus local area networks, pp. 5-34, Jun. 1985.
- [6] B.A. Loyer and D. Kolton, "Performance issues of 802.4 token bus LAN's," Workshop on analytic and simulation modeling of IEEE 802.4 token bus local area networks, pp. 153-167, Jun. 1985.
- [7] S.R. Sachs, K.L. Kan and J.A. Silvester, "Token-bus protocol performance analysis and comparison with other LAN protocols," Proc. of GLOBECOM, 1985.
- [8] J.W. Yeh, "Simulation of local computer networks," Proc. of 4th conf. on Local Computer Networks, 1979.
- [9] S.K. Pahimi and G.D. Jelatis, "LAN protocol validation and evaluation," IEEE J. of Seleted Areas in Comm., vol. SAC-1, pp. 790-802, Nov. 1983.
- [10] G.M. Birtwistle, O-J Dahl, B. Myhrhaug and K. Nygaard, SIMULA BENGIN, Studentlitterature, 1976.
- [11] J.Y. Chien, "Detailed tests show how well industrial local network performs," Data

Comm., Aug. 1985.

[12] 임동민, "Token passing LAN protocol의

델링 및 성능에 관한 연구" 석사학위논문, 한국과학기술원 전기 및 전자공학과, 1988. *

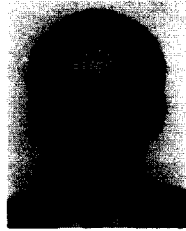
著 者 紹 介



林 東 民(正會員)

1962年 10月 9日生. 1986年 2月 서울대학교 제어계측공학과 졸업 공학사학위 취득. 1988年 2月 한국과학기술원 전기 및 전자공학과 졸업 공학석사학위 취득. 1988年 3月~현재 한국과학기술원 전기

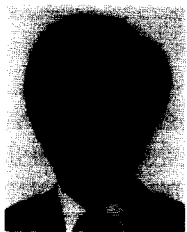
및 전자공학과 박사과정 재학중. 주관심분야는 Local area network 등임.



殷 鍾 官(正會員)

1940年 8月 25日生. 1964年 6月 미국 University of Delaware 전자공학과 졸업 공학사학위 취득. 1966年 6月 동대학원 공학석사학위 취득. 1969年 6月 동대학원 공학박사학위 취득. 1969年 9月~

1973年 5月 미국 University of Maine 전자공학과 조교수. 1973年 5月~1977年 6月 미국 스탠포드연구소(SRI) 책임연구원. 1977年 6月~현재 한국과학기술원 전기 및 전자공학과 교수 IEEE Fellow. 1986年 9月~현재 디지콤 정보통신연구소 고문. 주관심분야는 디지털통신 및 디지털 신호처리 등임.



李 愷 洙(正會員)

1952年 9月 19日生. 1975年 2月 서울대학교 공과대학 전기공학과 졸업. 1975年 1月~10月 현대조선중공업 주식회사 전장설계부. 1978年 8月 한국과학기술원 전기 및 전자공학과 공학석사학위 취득.

1983年 2月 한국과학기술원 전기 및 전자공학과 공학박사학위 취득. 1983年 3月~현재 한국과학기술원 전기 및 전자공학과 조교수. 1984年 4月~1985年 5月 Post Doc. 연구원(Information System Lab., Dept. of EE, Stanford University). 주관심분야는 디지털 신호처리, 컴퓨터통신망 등임.