論文 92-29B-5-3

지연정보 되먹임을 이용하는 고속 MAN용 MAC 프로토콜

(A MAC Protocol Using Delay Information Feedback for High-Speed MAN)

金 星 元*, 鄭 東 根**, 崔 悰 鎬**

(Sung Won Kim, Dong Geun Jeong, and Chong-Ho Choi)

要 約

넓은 지역과 빠른 전송속도에서 사용될 수 있는 새로운 고속 MAN용 MAC 프로토콜을 제안한다. 제안된 프로토콜은 P_i-퍼시스턴트 전송방법을 기초로 하며, 각 전송노드는 지연정보를 사용하여 주기적으로 전송확률을 구한다. 또한 전송확률이 트래픽 변화에 빨리 적응될 수 있도록 하기위한 윈도우 메카니즘을 제안한다. 제안된 프로토콜의 최대 처리량을 이론적 분석으로 알아본다. 시뮬레이션 결과에 의하면 제안된 프로토콜은 부하가 증가하더라도 공평한 메세지 지연시간 특성을 나타내며 버스 길이나 전송속도의 증가에 따른 메세지 지연시간의 변화폭이 DQDB보다 작다.

Abstract

This paper proposes a new MAC protocol for high-speed MAN. The proposed protocol is based on the P_i-persistent transmission scheme and each node calculates the transmission probability periodically by using delay information(DI). A window mechanism for the calculation of message delay in each node is proposed to improve the adaptability of the proposed protocol to traffic changes. The capactiy of the proposed protocol is analyzed. The simulation results show that the proposed protocol gives fair message delay under heavy load conditions and, when the transmission speed or distance is increased, the message delay variation of the proposed protocol is less than that of IEEE 802.6 DQDB.

*準會員, 金星社 情報器機研究所
(Inform. Sys. R & D LAB., GoldStar Co.)
**正會員, 서울大學校 制御計測工學科
(Dept of Cont. & Inst. Eng., Seoul Nat'l Univ.)
制御計測新技術 研究센터
(ERC-ACI by KOSEF)
自動化 시스템 共同 研究所
(ASRI)
接受日字 : 1992年 1月 28日

I. 서 론

LAN(local area network)에 비해 넓은 지역에서 빠른 전송 속도로 동작해야 하는 MAN(metropolitan area network)의 MAC(medium access control) 프로토콜은 다음 조건들을 만족해야 좋은 성능을 나타낼 수 있다[1]-[3] 첫째, 버스 길이나 전송속도의 증가에 따른 처리량(throughput)이나 지연시간의 변화가 작아야 한다. 둘째, 모든 노드는 위치에 상관없이 공평한 지연시간과 처리량을 나타내야 한다. 그 이유는 MAN에서와

같이 노드간의 거리가 긴 경우에는 제어 정보가 버스를 통해 전달되는데 많은 시간이 걸리므로 노드간의 전송기회가 불공평해지는 경우가 많이 발생하기 때문이다. 세째, 동적으로 변화하는 트래픽 상황에 잘 대처해야 한다. 네째, 노드의 가입과 탈퇴가 쉬워야 하며 프로토콜 작동이 단순해야 한다. 다섯째, 중앙집중식 망관리 보다는 분산관리가 바람직하다. 즉 모든 노드에서 독자적으로 상황을 판단하고 전송을 관리하는 것이 좋다. 왜냐하면 중앙집중식 망관리에서는 관리 노드가 고장이 나면 전체망의 동작이 정지해 버리지만 분산관리에서는 한 노드의 고장이 전체망에는 영향을 미치지 않기 때문이다

이러한 조건들을 만족시키기 위해 많은 프로토콜들이 제시되었다. 이중 대표적인 것으로는 DQDB (distributed queue dual bus)가 있다. DQDB는 프로토콜의 간단함과 좋은 성능으로 인해 MAN표준으로 채택되었다[4] 그러나 부하가 증가하면 공평성이 떨어지며 버스 길이나 전송 속도가 증가함에 따라 성능이 나빠지는 단점이 있다[5][6]

Mukherjee등은 P_i-퍼시스턴스(P_i-persistent) 프로토콜을 제안하였다[7][8] 이 프로토콜은 확률적 방법에 의해 전송을 결정하는 것으로 버스 길이나 트래픽 변화에 견실한 특성을 보인다. 그러나 이 프로토콜에는 몇 가지 문제점이 있다. 첫째, 모든 노드는 자신의 상대적 위치와 전체 동작노드의 갯수를 알고 있어야 한다. 둘째, 도착율을 계산하기 위해 모든 노드의 트래픽을 분리하여 기억하고 있어야 한다. 세째, 전체 인가부하(offered load)가 전부하(full load)를 넘으면 프로토콜의 안정성을 보장할 수 없게 된다. 한 예로 한 노드가 전부하로 동작한 후 다른 노드가 동작하게 되면 나중에 동작하게 된 노드는 전혀 전송하지 못하게 된다.

본 논문에서는 고속 MAN에서 비등시성(non-isochronous) 트래픽 전송을 위한 새로운 MAC 프로토콜을 제안한다. 제안된 프로토콜은 앞에서 제시된 MAN 프로토콜의 조건을 만족하도록 설계되었으며 버스 길이나 전송 속도의 변화에 따른 성능 변화가 작은 P_i-퍼시스턴트 전송기법을 사용한다. 그리고 Mukherjee 등이 제안한 프로토콜의 단점들을 없애는 방향으로 설계되었다.

Ⅱ. 제안된 프로토콜

제안된 프로토콜은 이중버스 구조에서 사용된다 (그림1). 각 노드는 서로 반대방향 전송에 이용되는 두 버스 각각에 전송탭과 수신탭으로 연결된다. 전

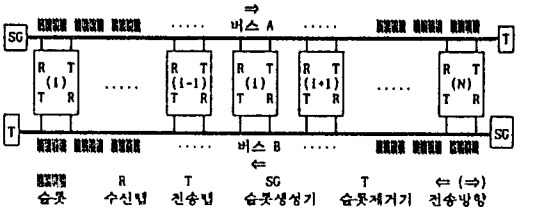


그림 1. 이중버스구조
Fig. 1. Dual bus architecture.

송탭은 버스로의 데이타 전송에 사용되고 수신탭은 버스로 전송되어 오는 데이타의 수신에 사용된다. 본 논문에서 노드의 구별은 가장 왼쪽의 노드를 1이라 하고 오른쪽으로 가면서 하나씩 증가하여 노드 지표(index)를 붙인다. 따라서 노드 i입장에서 보면 노드 $i+1$에서 부터 마지막 노드까지가 하위노드가 되고, 노드 1에서 노드 $i-1$까지가 상위노드가 된다. 각 노드는 전송할 데이타의 목적지에 따라 두 버스중 적당한 한 버스를 선택한다. 두 버스에 대한 동작은 대칭적이므로 앞으로는 한 버스에 대한 데이타 전송에 대해서만 설명하기로 한다. 버스의 시작부분에 슬롯 생성기(slot generator)가 있어 슬롯을 생성하며 버스의 마지막 부분에 슬롯 제거기(terminator)가 있어 슬롯을 제거한다. 각 노드에서의 전송은 슬롯(slot) 단위로 이루어 진다. 슬롯 생성기는 연속해서 슬롯을 보내므로 각 노드는 상위노드가 쓰고 남은 슬롯을 사용하게 된다. 제안된 프로토콜에서 노드 i는 빈 슬롯을 확률 P_i로써 사용하며, 전송확률 P_i를 계산하기 위해 하위노드들로 부터 되먹임 (feedback)되는 지연정보(delay information)를 사용한다. 따라서 제안된 프로토콜을 DIF(delay information feedback) 프로토콜이라 부르기로 한다.

슬롯은 용도에 따라 데이타 슬롯과 지연정보 슬롯으로 분류되며 그 구성은 그림2와 같다. 데이타 슬롯은 크게 헤더(header)와 정보부분(information field)으로 나누어 진다. 정보부분에는 전송하고자 하는 데이타들이 실리게 되며 헤더에는 제어정보가 실리게 된다. 제어정보에는 ACF(access control field), 수신 주소(destination address), 송신 주소(source address) 등이 있다. ACF에는 B(busy) 비트와 D (delay information) 비트가 있다. 슬롯 생성기에서는 데이타 슬롯의 B 비트를 0으로 하여 버스로 전송한다. 각 노드에서는 B 비트가 0인 데이타 슬롯의 정보부분에 자신의 데이타를 실으면서 B 비트를 1로 한다. B 비트가 1인 데이타 슬롯에는 데이타를 전

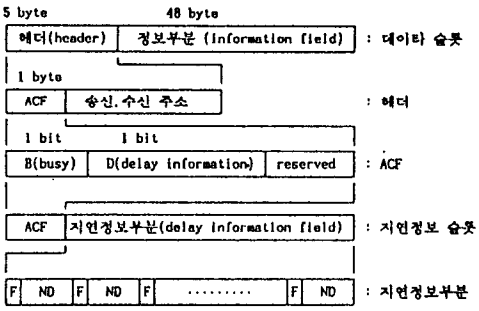


그림 2. 슬롯의 구성
Fig. 2. Slof format.

송하지 않는다. 그러므로 슬롯안에서 데이타의 충돌은 발생하지 않는다. B 비트와는 달리 D 비트는 슬롯 생성기에서 0 또는 1의 값을 쓰고 노드에서는 이 값을 읽기만 한다. D 비트가 0이면 데이타 슬롯임을 나타내고 1이면 지연정보 슬롯 (DI)임을 표시한다.

일반적으로 각 노드에서 전송하고자 하는 메세지의 길이는 한 슬롯의 정보부분의 길이와 같지 않다. 메세지의 길이가 정보부분의 길이보다 작을때는 한 슬롯에 메세지를 모두 보낼 수 있다. 그러나 한 메세지의 길이가 정보부분의 길이보다 길때는 메세지를 몇개의 패킷(packet)으로 나눈다. 한 패킷은 한 슬롯에 전송되는 길이가 된다. 그러므로 데이타를 받는 수신 노드는 이러한 패킷들을 받은 후, 한 메세지로 재결합한다.

지연정보 슬롯은 ACF와 지연정보 부분으로 이루어져 있다. 지연정보 슬롯의 ACF는 데이타 슬롯의 ACF와 같은 구성을 가지며 B 비트는 사용되지 않는다. 지연정보부분은 다시 여러개의 작은 슬롯(mini slot)으로 나눈다. 작은 슬롯은 F(full) 비트와 노드지연부분(ND:node delay)으로 이루어지는데, F 비트들은 노드지연부분이 사용되었는지를 표시하며 그 동작은 B 비트와 비슷하다.

각 노드는 지연정보 슬롯이 지나갈때 마다, 그때의 메세지 지연시간 값 '*WMD*'를 비어있는 노드지연 부분에 기록한다. 한 버스를 이용하는 전송에 대한 *WMD*는 ND에 기록되어 반대방향 버스를 통해 전달된다. 지연정보 슬롯은 슬롯 생성기에서 주기적으로 생성되며, 이 주기를 정보주기라 한다. *WMD*는 각 노드의 버퍼 크기와 메세지의 평균길이에 따라 달라진다. 어느 순간에는 *WMD*가 한 노드지연부분에 전송할 수 있는 크기보다 커져 전송할 수 없게 될 수

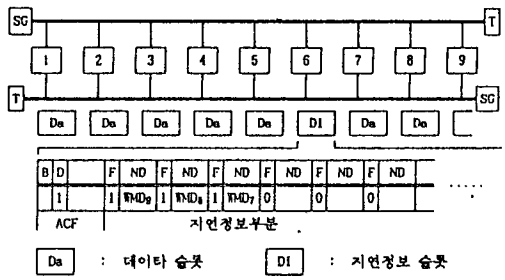


그림 3. 지연정보 슬롯의 사용 예
Fig. 3. Example of Delay informaton slot.

도 있다. 그래서 *WMD*의 최대값을 D_{max}로 제한한다.

각 노드에서 지연정보 슬롯을 사용하는 예가 그림 3에 있다. 그림3의 예에서는 지연정보 슬롯이 노드 6에 도착하고 있을때의 상황을 보여주고 있다. 지연정보 슬롯에는 노드 9부터 차례로 노드 7까지의 지연정보가 실려있다. 지연정보 슬롯이 슬롯 제거기에 도착하면 모든 노드의 지연정보가 실려있게 된다. 각 노드는 F 비트가 0인 ND부분을 사용하므로 하위노드부터 차례로 사용하게 된다. 그러므로 노드 i는 하위 노드들의 *WMD*를 지연정보 슬롯을 통해 알게 된다. 이렇게해서 받아들인 값과 자신의 WMD_i를 사용하여 확률전송 P_i를 다음과 같이 결정한다.

$$P_i = \frac{WMD_i}{WMD_i + RDT_i} \qquad (1)$$

여기서 RDT_i는 가장 최근에 갱신한 지연정보 슬롯의 값, 즉 최근의 DI를 통해 파악한 하위노드들의 *WMD*의 합이다. RDT_i는 매 정보주기마다 갱신되며 WMD_i는 매 슬롯 시간마다 바뀐다. 그러므로 P_i는 매 슬롯시간마다 새로 계산된다.

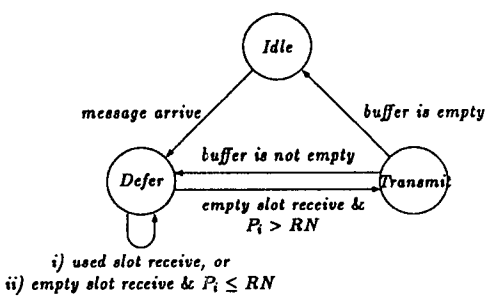


그림 4. DIF의 FSM(finite state machine)
Fig. 4. FSM(finite state machine) of DIF.

이상에서 설명한 DIF 프로토콜의 동작을 FSM (finite state machine)으로 나타내면 그림4와 같다. DIF에서 각 노드는 전송할 메시지가 없을 경우 '휴지(idle)' 상태에 있게 된다. '휴지' 상태에서 전송할 메시지가 도착하면 '연기(defer)' 상태가 된다. '연기' 상태에서 빈 슬롯이 지나갈때 P_i가, [0, 1]에 균일분포된(uniformly distributed) 난수(random variable)인 RN보다 크면 '전송(transmit)' 상태가 되고, 그 이외의 경우에는 계속 '연기' 상태로 있게 된다. '전송' 상태에서는 한 패킷의 전송이 이루어 지며, 전송 후 노드 버퍼가 비게되면 '휴지' 상태로 되고 전송할 패킷이 남아 있으면 '연기' 상태가 된다. 한편 각 노드는 어떤 상태에 있더라도 매 정보주기마다 하위 노드들의 WMD를 사용하여 P_i를 갱신하고, 지나가는 지연정보슬롯에 자신의 지연정보를 기록하는 작업을 한다.

Ⅲ. 윈도우 메시지 지연시간 및 구현

실제 트래픽은 매우 변화가 심하다. 한동안 메시지가 없다가 갑자기 대규모의 파일(file)을 전송하기도 하고, 부하의 크기가 급격히 줄어들기도 한다. 이런 상황변환에 대응하자면 노드 i의 전송확률 P_i도 트래픽 변화에 따라 변해야 좋은 성능을 나타낼 수 있다. DIF는 지연정보에 의해 전송확률을 결정한다.

한 메세지가 노드버퍼에 도착해서 완전히 전송되기까지 걸리는 시간이 메세지 지연시간(message delay)이 된다. 그림5는 노드 i에서 두개의 패킷으로 구성된 메세지의 지연시간을 나타내고 있다. $\tau 1$은 메세지가 도착한 후 새로운 슬롯이 노드 i에 도착할 때까지의 시간이며, $\tau 2$는 메세지 도착후 새로운 슬롯의 도착시간 부터 메세지 전송이 끝날때 까지의 시간으로 $\tau 1+\tau 2$가 메세지 지연시간이 된다. 일반적으로 $\tau 1$은 한 슬롯내에 균일하게 분포하며(uniform distrubution) 평균값은 한 슬롯시간의 1/2이 된다. 보통 $\tau 1$은 $\tau 2$에 비해 매우 작은 값이며, 본 논문에서는 계산의 복잡성을 피하기 위해 $\tau 2$를 메시지 지연시간으로 근사한다.

현재로부터 일정시간 이내에 버퍼에 존재했던 메세지에 대해 지정된 시간 내에서의 평균 지연시간을 윈도우 메세지 지연시간이라 한다. 이때 메세지 지연시간 계산에 고려되는 시간의 길이를 윈도우 크기라 한다. 본 논문에서 앞으로 사용될 기호를 다음과 같이 정의한다.

T: 한 슬롯시간
N: 패킷을 전송하는 전체 노드 갯수
M: 윈도우 크기(단위: 슬롯시간)
B: 각 노드의 버퍼크기(단위: 패킷)
R: 정보주기
ρ (rho): 인가부하(offered load)
L: 두 노드간의 거리(단위: 슬롯거리)
n: 1, 2, …
$DS(nT)$: 시각 nT에서 윈도우내의 지연시간의 합
$MS(nT)$: 시각 nT에서 윈도우내에 버퍼에 존재했던 메세지 갯수
$Arr(nT)$: 시각 $(n-1)T$에서 nT사이에 도착한 메세지 갯수
$Dep(nT)$: 시각 nT에서 전송이 완료되는 메세지 갯수, 0또는 1
$MN(nT)$: 시각 $(n-1)T$에서 nT사이에 전송이 끝나지 않아 대기중인 메세지 갯수
$WMD(nT)$: 시각 nT에서 윈도우 메세지 지연시간 $(=DS(nT)/MS(nT))$

윈도우 메세지 지연시간을 구하는 한 예를 그림 6을 통해 설명한다. 그림6에서 모든 메세지의 길이는 두 패킷이며 윈도우 크기는 5라고 가정한다. 시각 nT에서 두번째 메세지($M(2)$)는 윈도우 내에서 도착과 전송이 완료되었으므로 메세지 지연시간은 4가 된다. 그런데 첫번째 메세지($M(1)$)와 세번째 메세지($M(3)$)는 윈도우 내에서 완전한 메세지 지연시간을 알 수 없다. 이때는 윈도우 내에서 걸린 지연시간만을 고려하므로 첫번째 메세지 지연시간은 1, 세번째 메세지 지

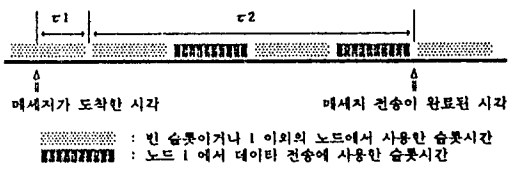


그림 5. 노드 i에서의 메세지 지연시간 예
Fig. 5. Example of message delay at node i.

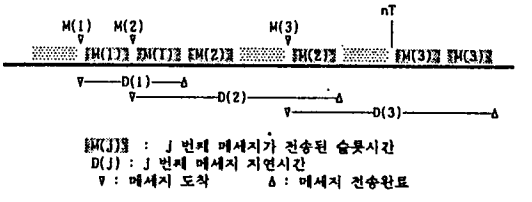


그림 6. 윈도우 메세지 지연시간의 예
Fig. 6. Example of window message delay.

연시간은 2가 된다. 그러므로 $DS(nT)$는 7이 되고 $MS(nT)$는 3이 된다.

윈도우 메세지 지연시간 계산을 실제 각 노드에 구현하는 방법은 다음과 같다. $WMD(nT)$를 구하기 위해서는 $DS(nT)$와 $MS(nT)$를 알아야 한다. $DS(nT)$와 $MS(nT)$는 다음과 같이 나타낼 수 있다.

$$DS(nT)=\sum_{i=n-W+1}^{n} MN(iT) \qquad (2)$$

$$MS(nT)=\sum_{i=1}^{n} Arr(iT) \ -\sum_{i=1}^{n-W} Dep(iT) \qquad (3)$$

여기서 (2)와 (3)은 다시 다음과 같은 반복식(recursive equation)으로 표현할 수 있다.

$$\begin{aligned} &DS(nT)\\ &=\sum_{i=n-W}^{n-1} MN(iT)+MN(nT)-MN[(n-W)T]\\ &=DS[(n-1)T]+MN(nT)-MN[(n-W)T] \end{aligned} \qquad (4)$$

$$\begin{aligned} &MS(nT)\\ &=\sum_{i=1}^{n-1} Arr(iT)+Arr(nT)\\ &\quad -\sum_{i=1}^{n-W-1} Dep(iT)-Dep[(n-W)T]\\ &=MS[(n-1)T]+Arr(nT)\\ &\quad -Dep[(n-W)T] \end{aligned} \qquad (5)$$

그러므로 $MS(nT)$와 $DS(nT)$를 구하기 위해서는 $MN(\cdot)$과 $Dep(\cdot)$를 윈도우 크기만큼 기억하고 있으면 되므로 두개의 카운터(counter)와 윈도우 크기만큼의 두개의 시간지연 기억장치(shift register)가 사용된다. (4)와 (5)를 실제 카운터로 구현한 것이 그림 7에 있다. $MN(\cdot)$은 0에서 버퍼 크기 만큼의 값을 가질수 있고, $Dep(\cdot)$은 0또는 1의 값을 가진다.

Ⅳ. 성능평가

DIF의 성능을 평가할 수 있는 정확한 수학적 모델을 만드는 것은 매우 어려운 일이다. 그러므로 DIF의 최대 처리량(capacity)만을 이론적 분석으로 밝히고 다른 성능은 시뮬레이션을 통해 보인다.

1. 최대 처리량의 이론적 분석

모든 노드에 계속해서 트래픽이 발생하고 있는 과부하(over load) 상태에서 DIF의 최대 처리량(capacity)을 이론적으로 분석한다. 모든 메세지는 한 패킷으로 이루어 졌으며 모든 노드의 버퍼 크기는 같다고 가정한다. D_{max}의 제한이 없을때 각 노드에서

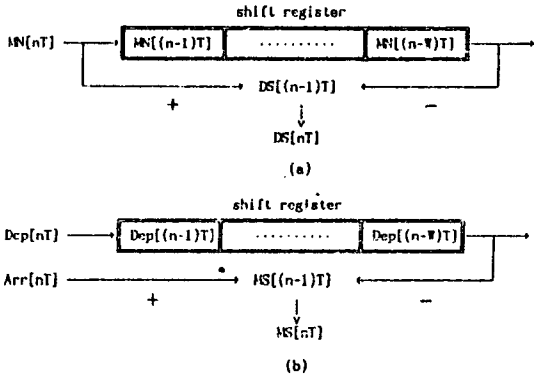


그림 7. DS(·) 및 MS(·) 카운터의 구현
(a) DS(·) 카운터 (b) MS(·) 카운터
Fig. 7. Implementation of DS(·) and MS(·) counter.
(a) DS(·) counter, (b) MS(·) counter.

계산되는 WMD는 모두 같게 된다.

$$WMD=B\cdot W \qquad (6)$$

노드 i에서 받게되는 하위 노드들의 지연시간 합 RDT_i는 항상 일정하며 다음과 같다.

$$RDT_i=\sum_{k=i+1}^{N} B\cdot W=B\cdot W\cdot(N-i) \qquad (7)$$

노드 i의 전송확률 P_i는 (1), (6), (7)에 의해 다음과 같다.

$$\begin{aligned} P_i&=\frac{WMD}{WMD+RDT_i}=\frac{B\cdot W}{B\cdot W\cdot(N-i+1)}\\ &=\frac{1}{N-i+1} \end{aligned} \qquad (8)$$

노드 i에서 사용되는 데이타 슬롯의 비율 즉, 처리량을 T_i, 노드 i에서 받게되는 빈 데이타 슬롯의 비율을 E_i라 하면 각 노드의 처리량은 다음과 같다.

$$\begin{aligned} T_1&=E_1\cdot P_1=E_1/N\\ T_i&=E_i\cdot P_i\\ &=[E_1-T_1-T_2-\cdots\cdots-T_{i-1}]\cdot P_i\\ &=E_1\cdot[1-(i-1)/N]/(N-i+1)=E_1/N\\ &\vdots\\ T_N&=E_N\cdot P_N\\ &=[E_1-T_1-T_2-\cdots\cdots-T_{N-1}]\cdot P_N\\ &=E_1\cdot[1-(N-1)/N]=E_1/N \end{aligned}$$

그러므로 모든 노드의 처리량은 E_1/N으로 같다.

한편 데이타 슬롯의 크기와 지연정보 슬롯의 크기가 같다고 가정하면, E_1은 $R/(R+1)$이므로 DIF의 최대 처리량은 다음과 같다.

$$\sum_{i=1}^{N} T_i = \sum_{i=1}^{N} E_i/N = R/(R+1) \qquad (9)$$

2. 시뮬레이션 및 고찰

시뮬레이션에서 모든 노드는 등간격으로 위치한다고 가정하며 다음과 같은 두 종류의 트래픽 모델이 사용된다.

(1) 확률적 부하(SL:stochastic load):각 노드에서 메세지 도착은 도착율 λ인 포아송(Poisson) 프로세스에 따른다. 또 메세지 길이(단위:패킷)는 평균이 $1/(1-\delta)$인 기하(geometric) 분포에 따른다. 단 메세지의 최장 길이는 30으로 제한된다. 따라서 전체 시스템의 인가 부하 ρ는 $N\cdot\lambda/(1-\delta)$로 근사된다 (예컨데 δ가 0.9일때 메세지 길이가 30이상일 확률은 5% 이하이다).

(2) 비확률적 부하(DL:deterministic load):각 노드에서 메세지 길이는 1로 고정되며 매 슬롯마다 λ개의 메세지가 도착한다. 따라서 전체 시스템의 인가 부하 ρ는 $N\cdot\lambda$이다.

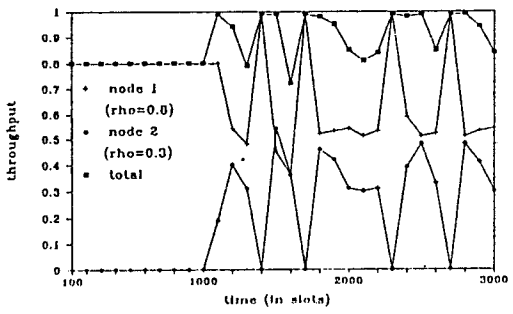
성능비교를 위해 IEEE 802.6표준안인 DQDB의 DQ 프로토콜도 시뮬레이션 하였다. DQ의 시뮬레이션 모델은 DIF의 모델과 같다. DQ의 시뮬레이션에서 BWB 메카니즘[4]이 사용되었으며 BWB (bandwidth balancing) 매개변수 값은 8로 하였다. 특별한 언급이 없는 한 시뮬레이션 모델의 매개변수 값은 표1과 같다.

표 1. 매개변수 값
Table 1. Values of simulation parameters.

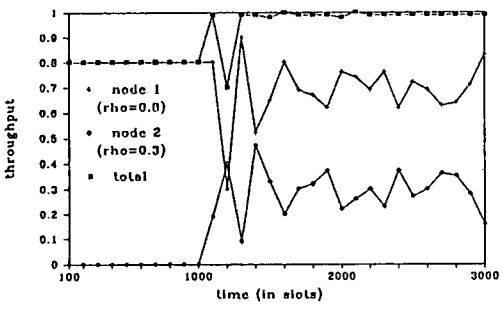
$\delta=0.9$
$D_{max}=100$ 슬롯시간
$W=100$ 슬롯시간
$N=50$
$R=99$슬롯시간
$T=2.83\mu$s
$B=500$패킷
각 버스의 전송속도=150Mbps
슬롯 크기=53octet
각 버스의 길이=100슬롯≈71Km
버스 전파시간(propagation delay time)=2.5×10^8m/s
버스 지연시간(latency)=100슬롯×2.83μs/슬롯=283μs

DIF를 구현할 때 값을 정해야 하는 프로토콜 매개변수로는 D_{max}, W 및 R이 있다. 이들 매개변수들이 프로토콜의 성능에 미치는 영향을 살펴보자.

W가 크면 많은 양의 과거정보를 가지고 있으므로 P_t는 빨리 변하지 못하며 따라서 급격한 트래픽의 변화에는 잘 적응하지 못하게 된다. 반면 W가 작으면, 트래픽의 변화에 P_t가 민감하게 변하므로 급격한 트래픽 변화에 잘 따르게 되나 안정성이 떨어진다. 즉 처리량의 진동이 심하게 되어 특정 환경에서는 부적당하게 된다. 과부하($\rho>1$)에서 W가 전체 이용도(utilization)에 미치는 영향이 그림8에 나타나 있다. 그림8을 위한 시뮬레이션에서는 먼저 노드1의 부하가 0.8로 인가된 후, $1000T$에서 노드2에 0.3의 부하를 인가한다. 이 경우 과부하가 가해지는 상태이므로 이론적인 최대 처리량은 0.99 이다. 그런데 그림8(a)에서는 W가 1인 경우로 전체 이용도가 이론값을 달성하지 못하고 진동함을 볼 수 있다. 각 노드의 처리량도 심한 진동을 하고 있다. 반면 그림8(b)에서는 W가 100인 경우로 전체 이용도가 항상 이론값이 되고 있으며 각 노드의 처리량도 안정함을 볼 수 있다.



(a)



(b)

그림 8. 과부하에서 W가 처리량에 미치는 영향 ($D_{max}=100$, $N=2$, $L=1$, DL)
(a) $W=1$ (b) $W=100$

Fig. 8. Effect on throughput by W under overload condition. ($D_{max}=100$, $N=2$, $L=1$, DL)
(a) $W=1$, (b) $W=100$.

D_{max}가 작으면 지연정보 슬롯의 크기를 줄일 수 있으나 정확한 메시지 지연시간이 상위노드에 전달되지 않는다. 반면 D_{max}가 크면 지연정보 슬롯의 크기는 늘어나지만 메시지 지연시간이 잘 전달되므로 노드의 트래픽에 따른 전송확률의 결정이 정확해 진다. D_{max}는 부하가 많은 노드가 전체 용량을 독점하게 되는 상황을 방지하기도 한다. 과부하($\rho>1$)에서 D_{max}가 전체 이용도에 미치는 영향은 W의 경우와 비슷하다.

이상의 결과들을 보면 W나 D_{max}가 너무 작은 값이면 성능이 나빠짐을 볼 수 있다. 반대로 W의 증가는 트래픽 변화에 대한 적응성을 저하시키고 D_{max}의 증가는 지연정보 슬롯의 크기를 증가시켜서 최대 이용도를 떨어뜨리게 된다. 그러므로 W나 D_{max}는 환경에 따라 적당한 값을 선택해야 한다. 한편 정보주기 R도 설계시 결정해야 하는 값이다. R이 작을경우 메시지 지연시간 정보를 빨리 전달해줄 수는 있지만 최대 이용도가 떨어지는 단점이 있다.

DIF는 부하가 증가하더라도 항상 공평한 지연시간 특성을 나타낸다. 그림9에는 부하에 따라 DIF와 DQ의 각 노드 메세지 지연시간이 나타나 있다. 그림9(a)에서 두 프로토콜이 모두 공평한 지연시간을 보이고 있으며 평균 지연시간은 DIF가 작음을 알 수 있다. 그림9(b)에서는 부하가 1.0인 경우에 대해 두 프로토콜을 비교하고 있다. DIF는 역시 공평한 지연시간 특성을 나타내고 있지만 DQ는 하위노드들의 지연시간이 매우 커짐을 볼 수 있다. 그러나 평균 지연시간은 DIF가 큼을 볼 수 있다. 즉 DQ는 부하가 늘어날수록 불공평성이 커지지만 DIF는 모든 인가부하에 대해 좋은 공평성을 나타내고 있다.

그림10에서는 평균 메시지 지연시간이 인가부하에 따라 변하는 모양을 두 버스 길이에 대해 비교하고 있다. 버스 길이의 변화에 따른 평균 지연시간의 변화폭이 DQ는 크고, DIF는 작음을 볼 수 있다. 또한 버스 길이가 길어질수록 DIF의 평균지연시간 값이 DQ에 비해 작은 부하 구간이 증가함을 알 수 있다. 즉, 두 프로토콜의 평균 지연시간값이 교차하는 지점이 버스 길이가 길어질수록 커짐을 보인다. 바꾸어 말해 DIF는 버스 길이가 길어질수록 DQ에 비해 좋은 성능을 보인다. 이 결과는 전송 속도가 증가하는 경우에도 적용된다. 왜냐하면 L은 버스 길이나 전송 속도의 증가에 의해 커지는 값이기 때문이다.

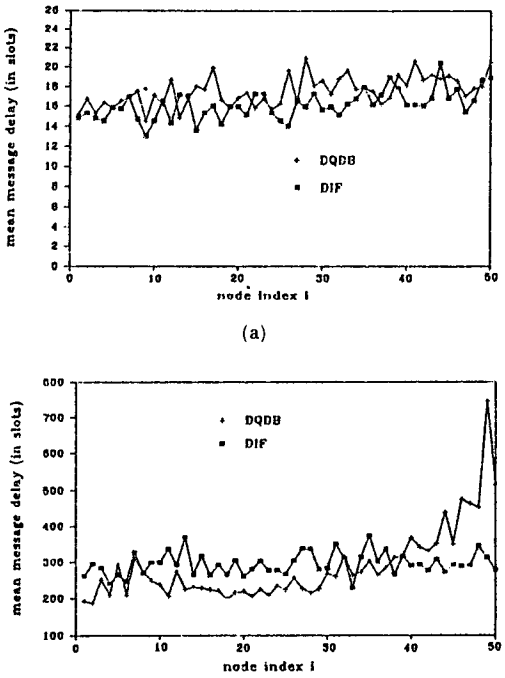


그림 9. 두 부하에 대한 공평성($N=50$, $L=2$, SL)
(a) $\rho=0.3$ (b) $\rho=1.0$

Fig. 9. Fairness under two different load conditions. ($N=50$, $L=2$, SL)
(a) $\rho=0.3$ (b) $\rho=1.0$

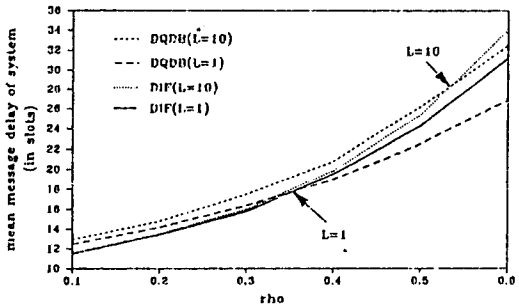


그림10. DIF와 DQ의 처리량-메세지 지연 특성 비교($N=20$, SL)

Fig. 10. Comparison of throughput-delay characteristics between DIF and DQ. ($N=20$, SL)

전체 트래픽이 과부하(over-load)가 되면 각 노드가 할당받은 전송용량을 적절히 조절하여 전송기회를 전혀 가지지 못하게 되는 노드가 없게해야 한다. 이와같은 문제를 해결하기 위해 비율제어(rate control)[5]가 많이 사용된다. 그림11은 DIF에 비율

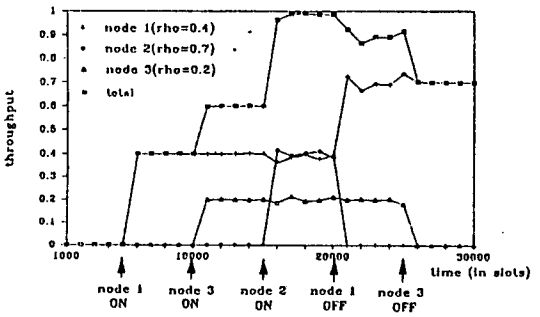


그림11. 비율제어의 예 (L=1, B=50, N=3, DL)
Fig. 11. Example of rate control. (L=1, B=50, N=3, DL)

제어 기능이 내재함을 보여준다. 그림11에는 시간이 흐르면서 각 노드의 처리량이 변하는 모양이 그려져 있다. 먼저 노드 1이 5000T에 0.4의 부하로 동작을 시작했다. 이후 10000T에 노드 3이 부하 0.2로 동작하고 있다. 전체 이용도는 0.6이며 각 노드의 처리량은 부하와 같다. 15000T에 노드 2가 부하 0.7로 동작을 시작했다. 전체 부하가 1을 넘어 비율제어가 시작되었다. 노드 1과 3은 부하만큼의 처리량을 보이지만 노드 2는 비율제어가 되어 부하가 0.7이지만 처리량은 0.4가 된다. 전체 이용도는 최대값을 나타내고 있다. 20000T에 노드1이 작동을 중지했다. 이때부터는 비율제어가 필요없어 진다. 노드 3은 계속 같은 처리량을 나타내고 노드 2는 그동안 비율제어에 의해 0.4의 처리량을 보이다가 부하만큼인 0.7의 처리량을 나타낸다. 이후 노드 3이 작동을 중지하면 노드 2는 인가부하 0.7을 계속 처리한다.

V. 결 론

DIF는 인가부하가 증가하더라도 노드간의 메세지 지연시간이 공평한 특성을 보였다. 반면 비교대상이었던 DQDB의 DQ 프로토콜은 인가부하가 증가할수록 공평성이 손상받는 특성을 보였다. 인가부하가 작은 상태에서 DIF는 DQ에 비해 작은 평균 메세지 지연시간을 나타내었다. 인가부하가 커질수록 DQ의 평균 메세지 지연시간이 작았다. 버스 길이가 늘어나거나 전송 속도가 빨라지면 DIF가 DQ에 비해 작은 메세지 지연시간을 보이는 부하 구간이 커진다. 이는 DIF가 버스 길이가 늘어나거나 전송 속도가 빨라지면 DQ에 비해 성능이 좋아짐을 의미한다. 또한 DIF는 버스 길이나 전송 속도의 변화에 따른 지연시간의 변화폭이 작다. 이 점은 P_i-퍼시스턴트 종류의 프로토콜에서 볼 수 있는 장점중의 하나이다. DIF는 윈도우내의 지연시간을 고려하므로 동적으로 변하는 트래픽 상황에도 잘 적응하는 성질을 보이는 반면 계산량이 많다는 단점도 있다.

이러한 결과들을 볼 때 DIF는 MAN뿐 아니라 고속 LAN에서도 유용하게 쓰일 수 있을 것이다.

參 考 文 獻

[1] M. Conti, E. Gregori, and L. Lenzin, "DCP: A distributed control polling MAC protocol: specifications and comparison with DQDB," *IEEE J. Select. Areas Commun.*, vol. 9, no. 2, pp. 241-247, Feb. 1991.

[2] M. M. Nassehi, "CRMA: An access scheme for high-speed LANs and MANs," in *Proc. SUPERCOMM/ICC'90,* 1990, pp. 1697-1702,

[3] I. Rubin and J.E. Baker, "Media access control for high-speed local area and metropolitan area communications networks," *Proc. IEEE*, vol. 78, no. 1, pp. 168-203, Jan. 1990.

[4] IEEE, "Distributed queue dual bus(DQDB)subnetwork of a metropolitan area netwoek(MAN)," IEEE Standard 802.6-1990, July 1991.

[5] E.L. Hahne, A.K. Choudhury, and N.F. Maxemchuk, "Improving the fairness of distributed queue dual bus network," in *Proc. INFOCOM'90,* June 1990, pp. 175-184.

[6] D.G. Jeong, C.H. Choi, and W.S. Jeon, "Fairness improvement in DQ protocol with multiple priority classes," in *Proc. ICC'91,* June 1991, pp. 1340-1344.

[7] B. Mukherjee and J.S. Meditch, "The P_i - Persistent protocol for unidirectional broadcast bus netweork," *IEEE Trans. Commun.*, vol. 36, no. 12, pp. 1277-1286, Dec. 1988.

[8] B. Mukherjee, A.C. Lantz, N.S. Matloff, and S. Banerjee, "Dynamic control and accuracy of the P_i - persistent kprotocol using channel feedback, " *IEEE Trans. Commun.*, vol. 39, no. 6, pp. 887-897, June 1991.

[9] 김성원, "새로운 고속 MAN 프로토콜의 설계," 서울대학교 공학석사학위논문, 1992.

著 者 紹 介

金 星 元(準會員)

1967年 11月 22日生. 1990年 서울대학교 공과대학 제어계측공학과 학사. 1992年 서울대학교 대학원 제어계측공학과 석사. 1992年~현재 금성사 정보기기연구소 연구원. 주관심분야는 프로토콜 설계 및 성능평가, G4 faximile 등임.

●

崔 悰 鎬 (正會員) 第28卷 B編 第8號 參照
현재 서울대학교 제어계측공학과 교수

鄭 東 根(正會員)

1959年 3月 16日生. 1983年 서울대학교 공과대학 제어계측공학과 학사. 1985年 서울대학교 제어계측공학과 석사. 1986年~1990年 한국데이타통신(주) 정보통신연구소 주임연구원. 1990年~ 현재 서울대학교 대학원 제어계측공학과 박사과정. 주관심분야는 통신망 프로토콜 설계, 통신망 성능평가, LAN/MAN 등임.