

DQDB(Distributed Queue Dual Bus) 프로토콜의 성능 분석†

이창훈* · 박광만** · 홍정완***

Performance Analysis of the DQDB Protocol

Chang Hoon Lie*, Gwang Man Park**, and Jung Wan Hong***

Abstract

In this paper, an analytical model of the message delay in the DQDB(Distributed Queue Dual Bus) network is investigated. The DQDB network has been adopted as a subnetwork for the IEEE 802.6 MAN(Metropolitan Area Network) standard. The DQDB network consists of two high speed unidirectional buses and a series of stations attached to both of the buses.

Messages arriving at each station consist of several packets according to its size. This system is approximated into " $B^{[X]}/G/1$ with exceptional first service queueing model" by defining the concept of service time on a packet. The service time for a packet is defined as the time from the instant the packet arrives at the transmission buffer until the time the packet is fully transmitted. By using the BASTA property and the average work in the system, the mean message delay time is obtained.

† 본 연구는 1993년도 한국학술진흥재단의 연구비 지원에 의하여 수행되었음.

* 서울대학교 산업공학과

** 한국전자통신연구소 교환방식연구실

*** 한국전자통신연구소 통신경영연구실

1. 서 론

컴퓨터 통신망의 발전으로 1980년대에 이르러 기존의 LAN과는 다른 개념의 MAN이 등장하게 되었다. MAN은 LAN보다 전송속도가 빨라지고 네트워크의 크기는 넓어지는데, 기존의 LAN 프로토콜을 사용하게 되면 전송매체의 최대 이용도가 급격히 떨어지게 된다고 알려져 있다.

1986년 오스트레일리아의 Telecom Australia에서 MAN 프로토콜로 QPSX (Queued Packet and Synchronous Exchange)를 제안하였다. 그리고 데이터 통신망에 대한 표준안 제안 기구인 IEEE 802위원회에서는 1989년에 QPSX에 바탕을 둔 DQDB를 MAN의 표준 프로토콜로 채택하였다.

DQDB가 IEEE 802.6 표준 프로토콜로 채택된 이후 DQDB에 대한 성능 평가가 활발히 이루어지고 있다. 그러나 성능 평가의 대부분이 시뮬레이션에 의한 연구였고[3,8,9,11] 분석적 방법에 의한 연구는 극히 적었다. 그리고 분석적 방법의 경우에 성능 평가 척도로 패킷의 평균지연시간을 제시하고 분석하였다[1, 2, 6, 7].

그러나 실제의 통신시스템에서는 패킷이 하나씩 도착하는 경우는 거의 없고, 대부분 다수의 패킷으로 이루어진 메시지가 도착하게 된다. 따라서 최근의 연구 동향은 다수의 패킷으로 구성된 메시지의 도착이 이루어질 때, 메시지의 지연시간을 분석하고자 하는 추세이다.

본 연구에서는 DQDB 프로토콜의 성능 평가 척도로 메시지의 평균지연시간을 제시하였고, 이를 분석적 방법에 의해 구하는 모형을 개발하였다. 이를 통해 시스템의 모수들이 시

스템의 성능에 미치는 영향을 파악할 수 있을 뿐만 아니라, MAN의 구축시 시스템의 설계에 반영할 수 있다. 그리고 사용자의 경우에는 이 통신 네트워크가 제공하는 서비스의 수준을 알아 볼 수 있을 것이다.

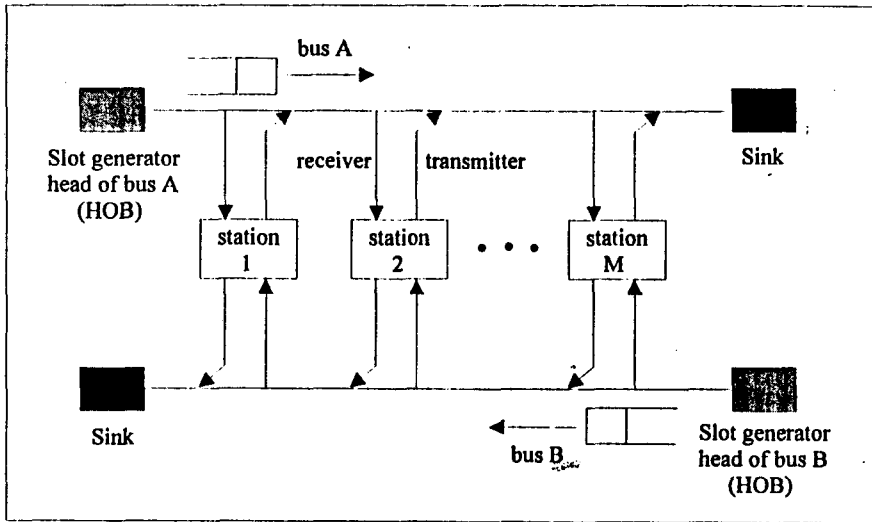
메시지의 도착을 고려한 Jing & Paterakis [5]의 분석에서도 한 메시지는 항상 일정한 n 개의 패킷으로 구성되어 있다고 가정하였고, 스테이션은 한 메시지의 대기만 허용한다고 가정하였다. 그러나 일반적으로 한 스테이션에서 다른 스테이션으로 전송하는 메시지의 길이는 다양하다.

본 연구에서는 Jing & Paterakis[5]의 분석에서 n 패킷으로 고정되어 있다고 가정하였던 메시지의 길이를 확률변수로 하여 분석을 일반화시키고자 한다. 그리고 대기용량도 하나의 메시지만을 수용하는 것이 아니라 무한하다고 가정하여 한 스테이션의 관점에서 메시지의 평균대기시간을 성능평가의 척도로 분석한다.

2. DQDB 프로토콜

DQDB는 분산 대기(distributed queue)라는 개념에 바탕을 둔 매체 접속 제어 프로토콜(media access control protocol)이다. 즉, 스테이션 각각에 분산되어 있는 대기를 통하여 네트워크 전체를 포괄하는 하나의 총괄적 대기행렬을 유지시킴으로써 개개의 스테이션에 도착하는 메시지가 FCFS로 전송매체에 접속하도록 제어하는 프로토콜이다.

DQDB 네트워크의 구조는 [그림 1]과 같이 두개의 단방향 버스와 이에 부착된 스테이션들로 구성되어 있다. [그림 1]에서 볼 수 있는



[그림 1] DQDB 네트워크의 토폴로지

것과 같이 버스들은 서로 반대 방향으로 데이터를 전달한다. 스테이션은 도착하는 데이터의 목적지에 따라 적절한 버스를 선택하여 목적지 스테이션으로 데이터를 전송하고, 다른 스테이션으로부터 전송되는 데이터를 수신할 수 있다. 각 버스의 가장 위쪽에는 HOB(Head Of Bus) 스테이션이 존재한다. HOB는 슬롯을 계속적으로 발생시키며, 이 슬롯은 데이터를 전송하는데 사용된다. 그리고 각 버스의 가장 아래쪽에는 Sink 스테이션이 존재하며 이 스테이션은 전송된 정보의 흐름을 종결한다. 한 슬롯은 53 octets으로 구성되어 있는데 이 중에서 1 octet은 접속제어부(access control field)이다. 접속제어부 중에서 프로토콜의 작동과정에 중요한 역할을 하는 것으로는 busy 비트와 request 비트가 있다. busy 비트는 슬롯이 패킷을 전송하고 있는지의 여부를 나타낸다. busy 비트가 1인 슬롯은 현재 패킷을 전송하고 있다는 것을 나타내며, 0인 슬롯은 비어있어 패킷

의 전송에 사용될 수 있다는 것을 나타낸다. 설명의 편의상 busy 비트가 1인 슬롯을 busy 슬롯, 0인 슬롯을 empty 슬롯이라 부르기로 한다. request 비트는 패킷 전송을 위하여 슬롯을 예약하는데 사용된다. 만약 스테이션이 버스 A를 통하여 전송할 패킷이 있는 경우에는 버스 B를 통해 운송중인 슬롯의 request 비트를 0에서 1로 바꾸어 주게 된다. request 비트가 1인 슬롯을 marked 슬롯, 0인 슬롯을 unmarked 슬롯이라 부르기로 한다.

패킷이 버스 A를 접속하고자 할 때의 프로토콜의 제어 과정은 버스 B를 접속하고자 할 때의 제어 과정과 동일하기 때문에, 버스 A를 접속하고자 할 때의 제어 과정에 대해서만 살펴보도록 하자. 편의상 스테이션의 상대적 위치에 따라 그 스테이션의 위쪽에 존재하는 스테이션은 상류(upstream) 스테이션, 그리고 아래쪽에 존재하는 스테이션은 하류(downstream) 스테이션이라고 한다.

한 스테이션에 도착한 메시지는 전송의 기본 단위인 패킷들로 이루어져 있고, 패킷들은 그 스테이션의 버퍼에서 전송되기까지 대기하게 된다. 버스 A를 통하여 하류 스테이션에 전달할 패킷이 있는 스테이션은 버스 B를 통하여 전송되고 있는 슬롯의 request 비트를 0에서 1로 바꾸어 상류 스테이션에 전송을 원하는 패킷이 존재함을 알리게 된다. 그러면 이 request는 상류 스테이션들에 등록된다.

각 스테이션은 request 카운터(이하 RQ_CTR)와 countdown 카운터(이하 CD_CTR)를 유지하며, 이것들은 총괄적 분산 대기를 형성하는데 이용된다. 모든 스테이션은 버스 B를 통하여 하류 스테이션이 요청한 request의 수를 RQ_CTR에 유지하는데, 이는 현재 버스 A에 접속을 요청한 하류 스테이션의 수를 알려주게 된다. 만약, 한 스테이션에 전송할 패킷이 도착하게 되면, 스테이션은 그 패킷을 empty 슬롯을 이용하여 전송하기 전에, 그때까지 RQ_CTR에 축적되어 있던 request의 갯수만큼 empty 슬롯을 그냥 보내게 된다. CD_CTR는 현재 그 스테이션이 전송할 패킷이 있음에도 하류 스테이션에 그 패킷보다 먼저 도착한 패킷들을 위하여 그대로 보내 주어야 할 empty 슬롯의 수를 유지한다.

각 스테이션은 분산대기를 위하여 하나의 버퍼를 유지하고 있고, 이 버퍼는 그 스테이션의 대기행렬의 가장 앞쪽에 위치하고 있다. 이 버퍼를 전송버퍼(transmission buffer : 이하 TB)라 부른다.

한 스테이션에서 전송할 패킷이 있는 경우와 없는 경우의 프로토콜 작동 과정은 다르다.

한 스테이션은 버스 A를 통하여 전송할 패킷이 없는 경우에 하류 스테이션으로부터 버스 B를 통하여 전달되는 매 request마다 RQ_

CTR을 1씩 증가시킨다. 그리고 만약 RQ_CTR가 0보다 크다면, 버스 A에 empty 슬롯이 지나갈 때마다 RQ_CTR를 1씩 감소시킨다.

한 스테이션에 버스 A를 통하여 전송할 패킷이 도착하면, 스테이션은 RQ_CTR의 수를 CD_CTR에 복사하고 RQ_CTR를 0으로 조정한다. 그리고 버스 A로 empty 슬롯이 지나갈 때마다 CD_CTR를 1씩 감소시키며, CD_CTR가 0이 된 다음 도착하는 empty 슬롯을 통하여 패킷을 전송하게 된다. 이 동안에도 스테이션은 버스 B의 상태를 계속 모니터하고 만약 request가 도착하면 RQ_CTR를 1씩 증가시킨다.

3. 메시지 지연시간의 분석

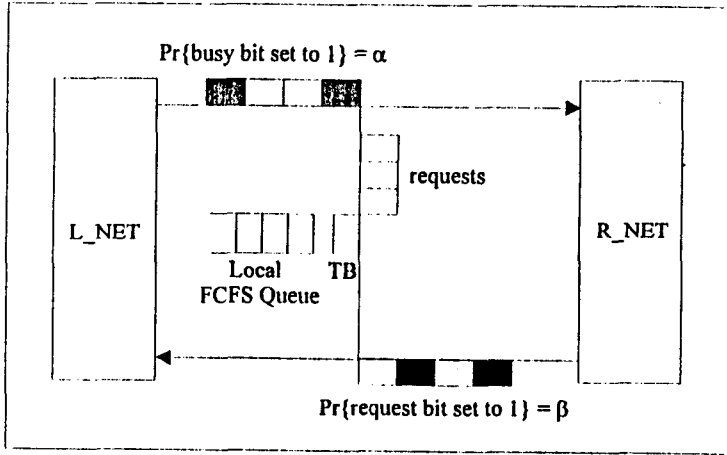
3. 1. 모형화

특정한 스테이션의 관점에서 살펴 보면 그 스테이션의 상류스테이션들은 L_NET, 하류스테이션들은 R_NET이라는 총괄적 스테이션으로 간주할 수 있다. 실제로 특정한 스테이션은 패킷을 전송하여 슬롯의 상태를 busy하게 한 상류스테이션을 구분할 수 없다. 마찬가지로 request를 보낸 하류스테이션도 구분할 수 없다. 따라서 L_NET와 R_NET 스테이션으로의 총괄화(aggregation)는 그 스테이션의 작동 과정에 영향을 미치지 않는다. 이와 같은 특정한 스테이션의 관점에서 상류스테이션들과 하류스테이션들로 총괄화하여 분석하는 방법은 Bisdikian이 제안한 이후 많이 쓰이고 있으며 [1,2,7], 본 연구에서도 이를 사용하였다.

L_NET은 버스A를 사용하여 메시지를 전송하기 때문에 특정한 스테이션에 도착하는 슬롯의 상태는 busy일 수도 empty일 수도 있다. 또한 R_NET는 버스B를 사용하여 request를 보내므로 특정한 스테이션에 도착하는 슬롯은 request를 포함할 수도 있고, 포함하지 않을 수도 있다.

따라서, 본 연구에서는 다음과 같이 DQDB 프로토콜을 모형화하였다.

- 1) L_NET는 α 의 확률로 busy한 슬롯을 생성한다.
 - 2) R_NET는 β 의 확률로 request를 보낸다.
 - 3) 메시지의 도착은 확률 λ 인 베르누이 과정 (Bernoulli process)을 따르며, 메시지의 크기는 일반적인 확률분포를 따른다.
 - 4) 스테이션의 버퍼 용량은 무한대이다.
- 다음 [그림 2]는 본 연구에서 분석하고자 하



[그림 2] 한 스테이션 관점에서의 분석모형

는 시스템을 보여주고 있다. 위의 [그림 2]를 살펴보면 한 스테이션이 서비스해야 하는 고객으로는, 그 스테이션에 도착하는 패킷과 하류 스테이션이 요청한 request의 두 가지임을 알 수 있다.

스테이션의 버퍼에 도착한 패킷의 서비스 시간을 그 패킷이 전송버퍼에 들어온 순간부터 매체 접속에 성공할 때까지의 시간으로 정의한다. 그러면 패킷은 스테이션의 버퍼에 도착하는 순간 전송버퍼가 빈 것을 발견하는 패킷(이하, type e 패킷)과 찬 것을 발견하는 패킷(이하, type b 패킷)의 두 가지 종류로 구분할 수

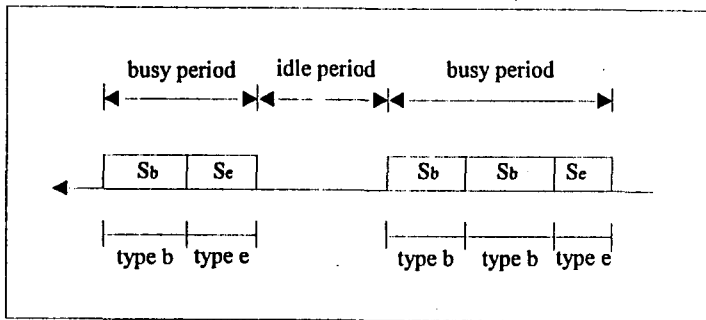
있다. Noh[7]는 이와 같이 전송버퍼의 상태에 따라 패킷을 구분하여 분석하였으나 성능 평가 척도로 패킷의 평균지연시간을 제시하였다. 본 연구에서는 Noh[7]에서 제시된 전송버퍼의 상태에 따라 패킷을 구분하여 분석하는 방법을 사용하였으나 성능 평가 척도로 메시지의 평균 지연시간을 제시하고 이를 분석하고자 한다.

S_e 를 type e 패킷의 서비스 시간, S_b 를 type b 패킷의 서비스 시간이라 하자. 그러면 S_e 의 분포는 type e 패킷이 시스템에 도착하는 순간의 RQ_CTR의 값에 따라 결정되는데, 이 때의 RQ_CTR의 값은 또한 이전 패킷이 시스

팀을 떠난 시점의 RQ_CTR의 값과 idle period 동안 RQ_CTR의 변화 상태에 의존한다. 그리고 S_b 의 분포는 type b 패킷이 전송버퍼에 도착하는 순간의 RQ_CTR의 값에 따라 결정된다. 이와 같이 패킷의 서비스 시간을 그 패킷이 전송버퍼에 들어온 순간부터 매체 접속에 성공할 때까지의 시간으로 정의하면, 우리가 분석하고자 하는 시스템은 메시지의 도착은 배치(batch)를 가진 베르누이 프로세스를 따르고, 각 패킷의 서비스 시간은 패킷이 시스템에 도착한 시점의 전송버퍼의 상태에 따라 서비스 분포가 달라지는 시스템이다.

그런데 type e 패킷은 도착한 순간에 전송버퍼가 빈 것을 발견하는 패킷이므로, 임의의 패킷이 type e 패킷이 되려면, 그 패킷이 속해 있는 메시지의 도착시점에 스테이션의 전송버퍼가 비어 있어야 하고, 그 패킷이 메시지의 첫번째 패킷이어야 한다.

그러면 이 시스템은 일반적인 $B^{[X]}/G/1$ 대기 모형과는 다른 “ $B^{[X]}/G/1$ with exceptional first service” 대기 모형으로 모델링할 수 있다. 이 대기 모형의 상태를 시간축에 도시하면 다음 [그림 3]과 같다.



[그림 3] $B^{[X]}/G/1$ Queue with exceptional first service

3. 2. 메시지의 평균지연시간의 분석

메시지의 지연시간을 메시지의 첫번째 패킷이 버퍼에 도착한 순간부터 마지막 패킷이 매체 접속에 성공할 때까지의 시간으로 정의하면 임의의 메시지의 지연시간은 다음의 관계를 만족한다.

$$\begin{aligned} & \text{임의의 메시지의 지연시간} \\ &= \text{메시지의 도착시점에 시스템내에 존재하는 작업량(work)} \\ & \quad + \text{메시지의 서비스 시간} \end{aligned}$$

그런데, 메시지의 도착은 베르누이 프로세스를 따르므로 BASTA(Bernoulli Arrival See

Time Average)가 성립한다. 따라서 D를 메시지의 지연시간을 나타내는 확률변수, V를 시스템내의 평균 작업량(average work) 그리고 S_b 를 임의의 메시지의 전체 서비스 시간을 나타내는 확률변수라고 정의하고, 위의 관계식에 평균을 취하면 다음과 같다.

$$E[D] = V + E[S_b] \tag{1}$$

앞의 관계식에서 살펴 볼 수 있듯이 메시지의 평균지연시간을 구하려면 시스템내의 평균 작업량 V와 임의의 메시지의 서비스 시간에 대한 평균치를 구해야 한다.

3. 2. 1. 평균 작업량 V의 계산

W : 임의의 패킷의 대기에서의 지연시간

N : 메시지 길이를 나타내는 확률변수

먼저 다음과 같이 용어를 정의하자.

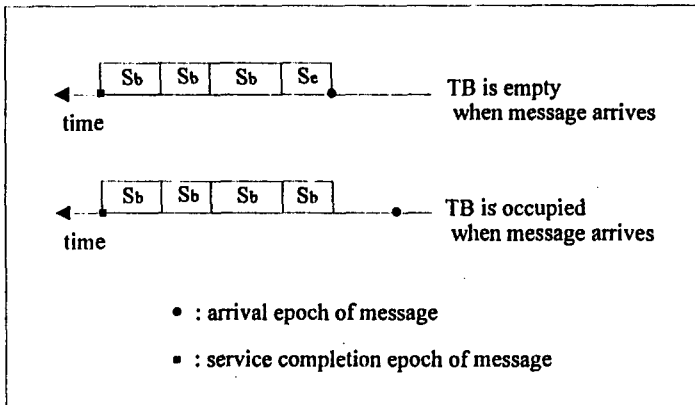
Pr[type e] : 임의의 패킷이 type e일 확률

Pr[type b] : 임의의 패킷이 type b일 확률

W_b : type b 패킷의 대기에서의 지연시간
(queue waiting time)

본 연구에서 분석하고자 하는 “B^[X]/G/1 with exceptional first service” 대기 모형의 시스템내의 평균 작업량 V는 다음과 같다[4, 10].

$$\begin{aligned}
 V &= \lambda E[N] Pr[type e] \frac{E[S_e^2] - E[S_e]}{2} \\
 &\quad + \lambda E[N] Pr[type b] (E[S_b W_b] + \frac{E[S_b^2] - E[S_b]}{2}) \\
 &= \lambda E[N] Pr[type e] \frac{E[S_e^2] - E[S_e]}{2} \\
 &\quad + \lambda E[N] Pr[type b] \frac{E[S_b^2] - E[S_b]}{2} + \lambda E[N] E[S_b] E[W]
 \end{aligned} \tag{2}$$



[그림 4] 임의의 메시지의 도착시 시스템 상황

앞의 식(2)를 살펴 보면 시스템내의 평균 작업량 V는 type e 패킷의 서비스 시간과 type b 패킷의 서비스 시간의 1, 2차 모멘트와 Pr[type e], Pr[type b], 그리고 E[W]의 함수임을 알 수 있다.

먼저 E[W]를 구해 보자. 임의의 패킷의 대기 지연시간은 다음 관계를 만족한다.

임의의 패킷의 대기 지연시간

= 패킷이 속한 메시지의 도착시점에 존재하는 시스템내의 작업량 + 메시지 내에서 패킷의 위치에 의한 지연시간

위의 관계식에 평균을 취하면 다음과 같다.

$$E[W] = V + E[W_b] \tag{3}$$

위의 관계식에서 W_b 는 임의의 패킷이 동일한 메시지안에서 자신의 위치에 따라 대기하게 되는 시간을 나타내는 확률변수이다.

그러면 $E[W_b]$ 를 구해 보자. 먼저 시스템의 트래픽(offered traffic)을 ρ 라고 하면 ρ 는 다음과 같다.

$$\rho = \lambda E[N] Pr[type e] E[S_e] + \lambda E[N] Pr[type b] E[S_b] \quad (4)$$

$$E[W_b] = \frac{(1-\rho)}{E[N]} (E[S_e](E[N]-1) + \frac{E[S_b](E[N^2] - 3E[N] + 2)}{2}) + \frac{\rho E[S_b](E[N^2] - E[N])}{2E[N]} \quad (5)$$

식 (1),(2) 그리고 (3)으로부터 시스템내의 평균 작업량 V 는 구할 수 있다.

그러면 임의의 패킷이 type e일 확률 $Pr[type e]$ 과 임의의 패킷이 type b일 확률 $Pr[type b]$ 을 구해 보자.

3. 2. 2. $Pr[type e]$ 와 $Pr[type b]$ 의 계산

임의의 패킷이 type e일 확률은 메시지의 도착 순간 시스템의 상태와 메시지의 크기에 조건을 두어 구해 보면 다음과 같이 구할 수 있다.

$$Pr[type e] = \frac{1-\rho}{E[N]} \quad (6)$$

따라서 $Pr[type b]$ 는 다음과 같이 표현할 수 있다.

$$Pr[type b] = 1 - \frac{1-\rho}{E[N]} \quad (7)$$

[그림 4]에서 알 수 있듯이 임의의 메시지가 도착했을 때, 그 메시지는 $(1-\rho)$ 의 확률로 시스템이 빈 것을 발견하고, ρ 의 확률로는 시스템이 찬 것을 발견하게 된다. 따라서 $E[W_b]$ 는 메시지의 도착시 시스템 상황과 메시지의 크기 즉, 배치의 크기와 패킷의 메시지내에서의 위치에 조건을 두어 다음과 같이 구할 수 있다 [4,10].

3. 2. 3. $E[S_b]$, $E[S_b^2]$ 과 $E[S_e]$, $E[S_e^2]$ 의 계산

type b 패킷의 서비스 시간의 1, 2차 모멘트는 [그림 3]에서 살펴 본 것과 같이 type b 패킷이 전송버퍼에 도착하는 순간의 RQ-CTR의 값에 따라 결정된다. 마찬가지로 type e 패킷의 서비스 시간의 1, 2차 모멘트는 type e 패킷이 시스템에 도착하는 순간의 RQ-CTR의 값에 따라 결정되는데 이것은 또한 이전 패킷이 시스템을 떠난 시점의 RQ-CTR의 값과 idle period 동안 RQ-CTR의 변화 상태에 의존한다.

type b 패킷과 type e 패킷의 1, 2차 모멘트는 이전 패킷의 type과 이전 패킷이 떠난 시점의 RQ-CTR의 값에 조건을 두어 구할 수 있다. 먼저 다음과 같이 용어를 정의하자.

$S_b(i)(S_e(i))$: 이전 패킷이 시스템을 떠난 시점의 RQ-CTR가 i 였을 때 type b(또는 type e)패킷의 조건부 서비스 시간

$\pi_b(i)(\pi_e(i))$: type b(또는 type e)패킷이 시스템을 떠난 시점의 RQ_CTR가 i일 극한 확률(limiting probability)

때 이전 패킷이 type e일 확률
Pr[pretype e|type e] : 현 패킷이 type e일 때 이전 패킷이 type e일 확률
그러면 $E[S_b]$ 과 $E[S_e]$ 은 다음과 같이 표현된

다.
Pr[pretype e|type b] : 현 패킷이 type b일

$$E[S_b] = \Pr[\text{pretype e}|\text{pretype b}] \sum_{i=0}^{\infty} \pi_e(i)E[S_b(i)] + (1-\Pr[\text{pretype e}|\text{type b}]) \sum_{i=0}^{\infty} \pi_b(i)E[S_b(i)] \quad (8)$$

$$E[S_e] = \Pr[\text{pretype e}|\text{type e}] \sum_{i=0}^{\infty} \pi_e(i)E[S_e(i)] + (1-\Pr[\text{pretype e}|\text{type e}]) \sum_{i=0}^{\infty} \pi_b(i)E[S_e(i)] \quad (9)$$

그리고 $E[S_b^2]$ 과 $E[S_e^2]$ 은 $\text{Var}[S_b]$, $\text{Var}[S_e]$ 와 앞에서 표현된 $E[S_b]$, $E[S_e]$ 를 이용하여 구

할 수 있다. $\text{Var}[S_b]$ 와 $\text{Var}[S_e]$ 는 다음과 같이 표현된다.

$$\text{Var}[S_b] = \Pr[\text{pretype e}|\text{type b}] \sum_{i=0}^{\infty} \pi_e(i)\text{Var}[S_b(i)] + (1-\Pr[\text{pretype e}|\text{type b}]) \sum_{i=0}^{\infty} \pi_b(i)\text{Var}[S_b(i)] \quad (10)$$

$$\text{Var}[S_e] = \Pr[\text{pretype e}|\text{type b}] \sum_{i=0}^{\infty} \pi_e(i)\text{Var}[S_e(i)] + (1-\Pr[\text{pretype e}|\text{type e}]) \sum_{i=0}^{\infty} \pi_b(i)\text{Var}[S_e(i)] \quad (11)$$

위의 식에서 Pr[pretype e|type b]은 다음과 같이 type e 패킷과 type b 패킷이 같은

메시지내의 패킷인지 아닌지에 따라 조건을 두어 구할 수 있다.

$$\begin{aligned} & \Pr[\text{pretype e}|\text{type b}] \\ &= \Pr[\text{pretype e}|\text{type b, 동일 메시지}]\Pr[\text{동일 메시지}|\text{type b}] \\ & \quad + \Pr[\text{pretype e}|\text{type b, 다른 메시지}]\Pr[\text{다른 메시지}|\text{type b}] \\ &= \frac{(1-\rho)r_1}{E[N]} + \left(1 - \frac{1}{E[N]}\right) \frac{(1-\rho)(1-r_1)}{E[N]} \end{aligned} \quad (12)$$

그리고 Pr[pretype e|type e]은 다음과 같이 구할 수 있다.

여기서 r_j 는 메시지의 크기가 j일 확률이다. 그러면 type b 패킷이 시스템을 떠난 시점의 RQ_CTR가 i일 극한 확률 $\pi_b(i)$ 와 type e 패킷이 시스템을 떠난 시점의 RQ_CTR가 i일

$$\Pr[\text{pretype e}|\text{type e}] = (1-\rho)r_1 \quad (13)$$

극한 확률 $\pi_e(i)$ 를 구하기 전에 먼저 다음을 정의하자.

$P_b(i,j)$ ($P_e(i,j)$) : 이전 패킷이 시스템을 떠난
시점의 RQ_CTR의 값이 i
라는 조건 하에서 type b

(또는 type e) 패킷이 시스템
을 떠날 때의 RQ_CTR
가 j 일 확률

그러면 $\pi_b(i)$ 와 $\pi_e(j)$ 는 다음과 같이 표현된다.

$$\pi_b(i) = \sum_{j=0}^{\infty} \Pr[\text{pretype } e | \text{type } b] \pi_e(j) P_b(j, i) + \sum_{j=0}^{\infty} (1 - \Pr[\text{pretype } e | \text{type } b]) \pi_b(j) P_b(j, i) \tag{14}$$

$$\pi_e(i) = \sum_{j=0}^{\infty} \Pr[\text{pretype } e | \text{type } e] \pi_e(j) P_e(j, i) + \sum_{j=0}^{\infty} (1 - \Pr[\text{pretype } e | \text{type } e]) \pi_b(j) P_e(j, i) \tag{15}$$

여기서 $P_b(i,j)$, $E[S_b(i)]$, 그리고 $\text{Var}[S_b(i)]$ 를 구해 보자.

$P_b(i,j)$ 는 type b 패킷의 서비스 시간에 조건을 두어 구해 보면 다음과 같다.

$$P_b(i, j) = \begin{cases} \sum_{k=i+1}^{\infty} \binom{k}{i} \beta^{k-i} \binom{k-1}{i} (1-\alpha)^{i+1} \alpha^{k-i-1}, & i+1 \geq j \\ \sum_{k=j}^{\infty} \binom{k}{j} \beta^{k-j} \binom{k-1}{j} (1-\alpha)^{j+1} \alpha^{k-j-1}, & i+1 < j \end{cases} \tag{16}$$

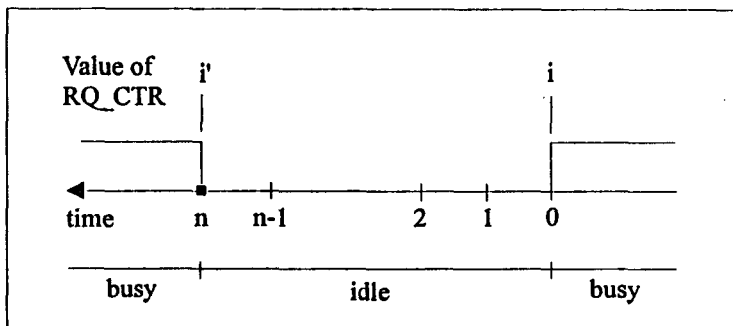
그리고 $E[S_b(i)]$, $\text{Var}[S_b(i)]$ 는 확률 $1-\alpha$ 를 갖는 기하분포의 $(i+1)$ -convolution이므로 다음과 같다.

$$E[S_b(i)] = \frac{i+1}{1-\alpha} \tag{17}$$

$$\text{Var}[S_b(i)] = \frac{(i+1)\alpha}{(1-\alpha)^2} \tag{18}$$

$P_e(i,j)$, $E[S_e(i)]$ 그리고 $\text{Var}[S_e(i)]$ 는 [그림 5]에서 볼 수 있듯이 type e 패킷의 도착시점과 그 시점에서의 RQ_CTR의 수에 의해 결정된다.

$Y(n)$ 을 idle period의 시작시점을 0으로 잡았을때 n 시점에서의 RQ_CTR의 수라고 정의하면 $Y(n)$ 은 다음과 같은 random walk 프로세스를 따른다.



[그림 5] type e 패킷 도착시 시스템 상태

$$Y(n) = \max[0, Y(n-1) + \zeta] \quad (19)$$

$$\text{여기서, } \zeta = \begin{cases} 1, & \text{w.p. } \alpha\beta \\ 0, & \text{w.p. } 1-\alpha\beta-(1-\alpha)(1-\beta) \\ -1, & \text{w.p. } (1-\alpha)(1-\beta) \end{cases}$$

다음과 같이 위 random walk 프로세스의 n-단계 전이 확률(n-step transition probabi-

$$P_e(i, j) = \sum_{n=1}^{\infty} (1-\lambda)^{n-1} \lambda \sum_r q^{(n)}(i, i') P_e(i', j) \quad (21)$$

$$E[S_e(i)] = \sum_{n=1}^{\infty} (1-\lambda)^{n-1} \lambda \sum_r q^{(n)}(i, i') \frac{(i'+1)}{(1-\alpha)} \quad (22)$$

$$\text{Var}[S_e(i)] = \sum_{n=1}^{\infty} (1-\lambda)^{n-1} \lambda \sum_r q^{(n)}(i, i') \frac{\alpha(i'+1)}{(1-\alpha)^2} \quad (23)$$

3. 2. 4. E[S_B]의 계산

패킷으로 이루어진 임의의 메시지의 전체 서비스 시간에 대한 평균값 E[S_B]는 앞절에서

lity)을 정의하자.

$$q^{(n)}(i, i') = \Pr[Y(n) = i' | Y(0) = i], n \geq 1 \quad (20)$$

그러면 P_e(i,j), E[S_e(i)] 그리고 Var[S_e(i)]는 type e 패킷의 도착시점과 그 시점에서의 RQ_CTR의 수에 조건을 두어 구하면 다음과 같다.

E[W_B]를 구했던 절차에 따라서 구할 수 있다. 즉, 메시지의 도착시 시스템 상황과 메시지의 크기에 조건을 두어 E[S_B]를 구하면 다음과 같다[4,10].

$$E[S_B] = (1-\rho) \left(E[S_e] + E[S_b] \frac{E[N^2] - E[N]}{E[N]} \right) + \frac{E[S_b] E[N^2]}{E[N]}$$

3. 3. 메시지의 평균지연시간 계산 절차

지금까지 DQDB 프로토콜의 성능 평가 척도로 본 연구에서 제시된 메시지의 평균지연시간을 구하기 위한 분석을 수행하였다.

그런데 부하 트래픽 ρ는 식(4)에서 볼 수 있듯이 E[S_e]와 E[S_b]의 함수로 표현된다. 마찬가지로 E[S_e]와 E[S_b]을 구하는 과정을 살펴보면 E[S_e]와 E[S_b]은 부하 트래픽 ρ의 함수로 나타나 있다. 따라서 E[S_e]와 E[S_b]은 다음과 같은 순환적인 절차(iterative procedure)에 따라 구할 수 있다.

Step 1 P_b(i,j), q⁽ⁿ⁾(i,j), P_e(i,j), E[S_b(i)], E[S_e(i)]를 계산한다.

Step 2 (initialization step)

n을 0으로 둔다. ρ₀를 r로 둔다. 단, r은 0과 1사이의 값이다.

Step 3 ρ_n 값을 이용하여 π_b(j), π_e(j)를 계산한다.

Step 4 Step 1에서 구한 E[S_b(i)], E[S_e(i)]와 Step 3에서 구한 π_b(j), π_e(j)를 이용하여 E[S_b], E[S_e]를 계산한다.

Step 5 n = n+1로 둔다.

Step 4에서 구한 E[S_b], E[S_e]를 이용하여 ρ_n값을 계산한다.

|ρ_n - ρ_{n-1}| ≤ ε이면 Step 6으로 가고 아니면 Step 3으로 간다.

Step 6 $\rho_n = \rho$ 로 둔다.

ρ 값을 이용하여 $E[S_a^2]$, $E[S_b^2]$ 을 계산한다.

V 및 $E[S_b]$ 값을 계산하여 메시지 지연시간 $E[D]$ 를 계산한다.

을 다양한 α, β, λ 에 대하여 구해 보았다. 스테이션에 부과되는 총부하량 ρ_{total} 은 다음과 같이 정의된다.

$$\rho_{total} = \lambda E[N] + \alpha + \beta$$

식(25)에서 살펴 본 바와 같이 시스템이 안정상태에 있기 위하여 총부하량은 1보다 작아야 한다.

본 연구에서 제시한 분석방법을 이용하여 스테이션에 도착하는 메시지의 길이가 r인 기하분포를 따르는 경우의 메시지의 평균지연시간을 다양한 $\alpha, \beta, \lambda, r$ 값의 범위에서 구하였고, 이를 시뮬레이션 결과와 비교하였다.

시뮬레이션은 시뮬레이션 전용 패키지인 SIMAN으로 코딩하여 IBM486 PC에서 수행하였다. 한번의 실행(run)에서는 모두 10000개의 메시지를 발생시켜 메시지의 평균지연시간을 구하였고, 이와 같은 실험을 모두 30번 수행하여 산술평균한 값을 메시지의 평균지연시간으로 예측하였다.

본 연구에서 제시한 분석방법으로 구한 메시지의 평균지연시간과 시뮬레이션을 통하여 구한 결과와의 비교를 그림으로 나타내면 다음 [그림 6]-[그림 8]과 같다.

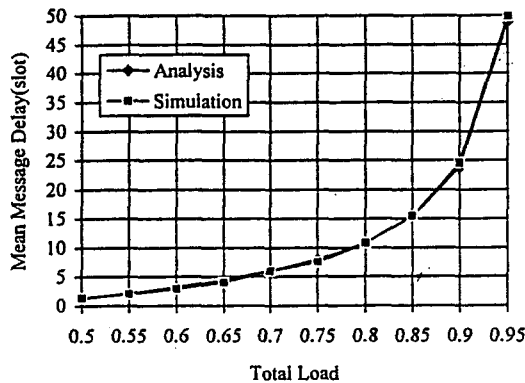
4. 결과 및 예제

지금까지 DQDB 프로토콜의 성능평가척도로 본 연구에서 제시된 메시지의 평균지연시간을 구하는 과정을 보였다.

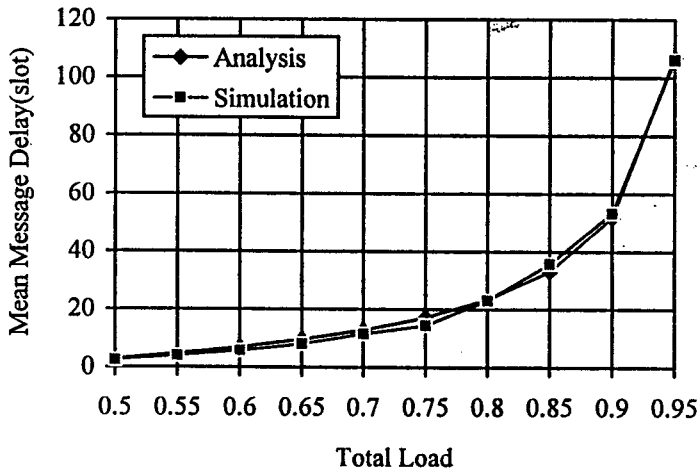
메시지의 평균지연시간은 시스템내의 평균작업량 V와 임의의 메시지의 전체 서비스 시간에 대한 평균의 합으로 구할 수 있다. 이 대기 시스템이 안정되기 위해서는 스테이션에 도착하는 패킷과 request의 트래픽 비율보다 그 스테이션에 도착하는 empty 슬롯의 비율이 높아야 한다. 따라서 다음과 같은 안정조건(stability condition)이 성립하여야 한다.

$$\lambda E[N] + \beta < 1 - \alpha \tag{25}$$

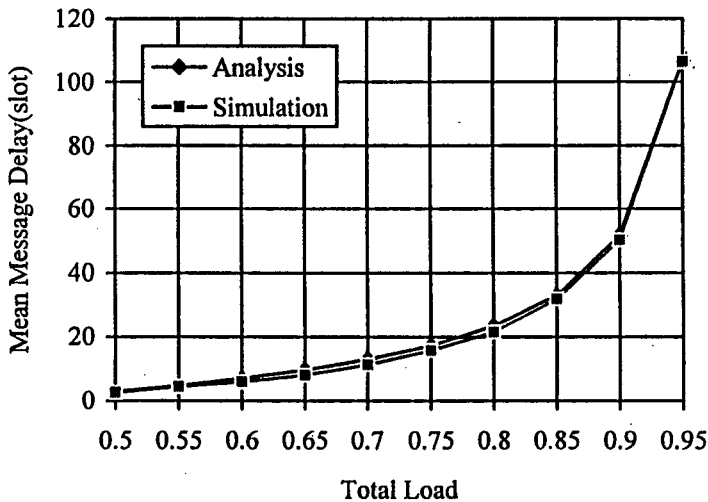
본 연구에서 분석한 결과를 이용하여 스테이션에 도착하는 메시지의 길이가 확률 r인 기하분포를 따르는 경우의 메시지의 평균지연시간



[그림 6] 메시지의 평균지연시간 비교($\alpha=0.1, \beta=0.1, \lambda=0.2$ 의 경우)



[그림 7] 메시지의 평균지연시간 비교(α=0.1, β=0.2, λ=0.1의 경우)



[그림 8] 메시지의 평균지연시간 비교(α=0.2, β=0.1, λ=0.1의 경우)

5. 결론

본 연구에서는 MAN의 매체접속제어 프로토콜의 표준으로 채택된 DQDB 프로토콜의 성능 평가 척도를 제시하고 그것을 분석하는 절차를 보였다.

기존의 대부분의 분석에서는 프로토콜의 성능 평가 척도로 패킷의 평균지연시간을 제시하고 이를 구하였으나, 본 연구에서는 메시지의 길이를 확률변수로 하여 메시지의 평균지연시간을 구하였다.

본 연구에서 성능 평가 척도로 제시된 메시지의 평균지연시간은 이 시스템을 “ $B^{[X]}/G/1$ ”

with exceptional first service" 대기모형으로 근사화하여 구하였다.

앞의 그림에서 살펴 볼 수 있는 바와 같이 본 연구에서 제시한 분석 방법에 의해 계산된 메시지의 평균지연시간과 시뮬레이션을 수행하여 얻은 결과가 매우 근접함을 알 수 있다.

참고 문헌

- [1] Bisdikian, C., "Waiting Time Analysis in a Single Buffer DQDB(802.6) Network," IEEE J. on Selected Areas in Commun., Vol. 8, No. 8(1990), pp. 1565-1573.
- [2] Bisdikian, C., "A Queueing Model for a Data Station within the IEEE 802.6 MAN," IBM Research Division, RC 15587, Mar., 1990
- [3] Conti, M., E. Gregori and L. Lenzini, "A Methodological Approach to an Extensive Analysis of DQDB Performance and Fairness," IEEE J. on Selected Areas in Commun., Vol. 9, No. 1(1991), pp. 76-87.
- [4] Cooper, R. B., Introduction to Queueing Theory, North Holland, 1981
- [5] Jing, W. and M. Paterakis, "Message Delay Analysis of the DQDB(IEEE 802.6) Network," IEEE INFOCOM'92 (1992), pp. 527-535.
- [6] Mukherjee, B. and S. Banerjee, "Alternative Strategies for Improving the Fairness in and an Analytical Model of DQDB Networks," IEEE INFOCOM'91(1991), pp. 879-888.
- [7] Noh, S. J., "Performance Evaluation of the Distributed Queue Dual Bus Metropolitan Area Network," Ph. D. Dissertation, University of Michigan, 1992
- [8] Stavrakakis, I. and R. Landry, "Delay Analysis of the DQDB MAN Based on a Simple Model," IEEE ICC'92(1992), pp. 154-158.
- [9] Tran-Gia, P. and T. Stock, "Approximate Performance Analysis of the DQDB Access Protocol," Computer Networks and ISDN Systems, Vol. 20 (1990), pp. 231-240.
- [10] Wolff, R. W., Stochastic Modeling and the Theory of Queues, Prentice-Hall International Editions, 1989
- [11] Wong, P. C., "A Frame-based DQDB protocol for Consecutive Slots Transmission in Gbit/s Metropolitan Area Networks," IEEE ICC'91(1991), pp. 1049-1055.