

論文96-33A-2-1

서비스 가로채기가 있는 네트워크 접속장치내의 유한버퍼의 분석

(Analysis of a Finite Buffer with Service Interruption in a Network Interface Unit)

金 永 翰 *

(Young Han Kim)

요 약

본 논문에서는 네트워크 접속장치내의 유한 버퍼에서의 패킷 손실율을 분석하였다. 일반적인 네트워크 접속 장치는 망과 컴퓨터간의 접속을 담당하며 정합 프로토콜을 처리하기위한 마이크로프로세서와 프로토콜 프로세서등을 갖추고 있다. 이러한 네트워크 접속장치 설계에서 중요한 요소중의 하나는 패킷 수신버퍼의 용량결정으로서 마이크로프로세서와 프로토콜 프로세서의 영향을 고려한 분석이 필요하다. 본 논문에서는 패킷이 여러개의 블록으로 나누어져 수신버퍼에 저장된후 서비스되는 것을 고려하여 패킷서비스를 단속 시간 서비스 형태로 모델링하였으며 다시 단속적인 서비스시간을 소 슬롯으로 나누어진 모델로 근사시켜 패킷의 손실율을 분석하였다.

Abstract

In this paper, we analyzed the packet blocking probability of a finite buffer in a network interface unit. In general, a network interface unit which provides a means of interface between the network and computer has a microprocessor and a protocol processor for the network access protocols. It also has a receive buffer for the arriving packets from the network which is served by the microprocessor with service interruption by the protocol processor. In this paper, we modeled the receive buffer as a discrete time server with service interruption, and obtained the packet blocking probability using the mini-slot approximation.

I. 서 론

컴퓨터등 각종 정보기기를 근거리 통신망, ATM (asynchronous transfer mode) 네트워크, 패킷 네트워크등 각종 통신망에 접속시킬 때에 일반적으로 통신망과 단말기간의 접속프로토콜을 처리해주는 네트워크 접속 장치가 이용된다. 예를 들어 LAN (local area network) 어댑터 카드, ATM 어댑터 카드와 같이 컴퓨터의 내부 시스템 버스에 접속되어 컴퓨터를 통신망에 접속시켜주는 장치들이 여기에 속한다. 이러

한 네트워크 접속 장치의 구성에 있어서 가장 중요한 역할을 하는 것은 네트워크 접속시 대부분의 링크계층 프로토콜을 수행해 주는 VLSI 프로토콜 프로세서이다^{[1]-[5]}. VLSI 프로토콜 프로세서는 내부에 DMA (direct memory access) 제어기능을 갖추고 망으로부터 입력되는 패킷을 네트워크 접속장치내의 패킷 수신 버퍼에 직접 저장시킨다. 필요시에는 프로토콜 프로세서내에 FIFO (first in first out) 버퍼를 갖추고 DMA에 의해 수신버퍼로 패킷이 이동되기 전까지 망으로부터 입력되는 패킷을 저장하며 패킷이 손실되는 것을 방지하기도 한다^{[1], [3]}. 그밖에 프로토콜 프로세서가 처리하는 작업이외의 통신 프로토콜을 처리하는 부하를 호스트 컴퓨터에 주지않을 경우에는 네트워크 접속 장치내에 마이크로프로세서를 두고 필요한 프로

* 正會員, 崇實大學校 情報通信工學科

(Regular Member, Dept. of Information and Telecom., Soongsil Univ.)

接受日字: 1995年11月9日, 수정완료일: 1996年1月24日

토콜을 처리하게 한다. 이때 마이크로프로세서는 수신버퍼에 저장되어 있는 패킷을 읽어 해당되는 처리를 하는 수신버퍼의 서버가 된다.

이상과 같은 네트워크 접속 장치를 설계할 때 하드웨어적으로 중요한 결정요소는 마이크로프로세서, 프로토콜 프로세서의 선정과함께 수신버퍼의 적정 용량결정이다. 수신버퍼의 용량은 유한할 수 밖에 없으며 이 용량이 부족할 경우 망으로부터 입력되는 패킷의 지나친 손실을 초래하게 된다. 그러므로 마이크로프로세서와 프로토콜 프로세서의 상호 영향 및 네트워크로부터의 트래픽을 고려한 수신버퍼의 용량결정이 설계에 있어서 필수적이다.

지금까지의 네트워크 접속 장치에 대한 분석으로는 프로토콜 프로세서내의 FIFO 버퍼에 대한 분석이 DMA와 관련하여 수행되었고¹⁶⁾, 마이크로프로세서와 프로토콜 프로세서의 상호 동작을 고려한 분석은 없었다. 본 논문에서는 프로토콜 프로세서 내의 FIFO는 충분히 크므로 DMA 설정 요구시의 지연에 의한 패킷 손실은 무시될 수 있으므로 수신버퍼에서 DMA제어기와 마이크로프로세서간의 액세스 충돌에 의한 수신버퍼 서비스 제한때문에 생기는 패킷 손실등을 분석하였다. 이러한 분석은 각종 네트워크 접속 장치의 설계시 뿐만아니라 다른 통신시스템에서 공통자원의 액세스 충돌에의해 생기는 서버의 서비스 중단이 있는 시스템의 분석에 이용될 수 있을 것이다.

서론에 이어 2장에서는 분석 시스템의 동작과 모델을 살펴보고 3장에서 이에 대한 분석을 수행한다. 4장에서는 분석결과를 수치적 예를 들어 살펴보고 5장에서 결론을 맺는다.

II. 시스템 모델

일반적인 네트워크 접속기의 기능적 구조를 그림 1에 나타냈다. 망으로부터 데이터를 수신할 때 네트워크 프로토콜 프로세서는 데이터를 DMA 채널을 통하여 접속장치내의 수신버퍼로 이동시킨다. 이때 DMA 준비시간이 길거나 수신버퍼가 부족하게 되면 데이터는 손실된다. 그러므로 CPU는 가능한 빨리 네트워크 프로토콜 프로세서의 DMA 설정 요구에 응답해야 한다. 대개의 네트워크 프로토콜 프로세서는 내부에 FIFO (first in first out) 버퍼를 갖고서 망으로부터 입력되는 데이터를 일시 저장할 수 있게하여 손실을 줄이

고 있다. 또한 FIFO에 일시 저장된 데이터를 DMA에 의해 일시에 수신버퍼로 이동시킴으로써 DMA 설정요청을 매번하는 오버헤드를 감소시킬 수도 있다.

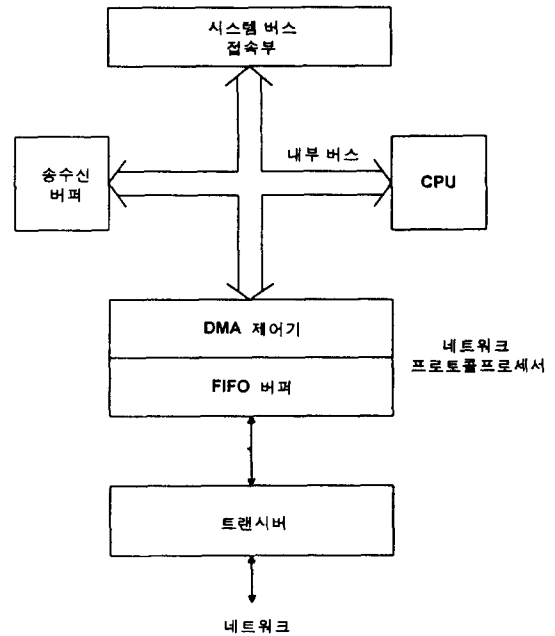


그림 1. 네트워크 접속 장치의 구조

Fig. 1. Structure of Network Interface Unit.

CPU는 수신버퍼내의 데이터를 읽어서 추가적인 프로토콜 처리를 수행한후 이를 컴퓨터의 시스템 버스등을통해 컴퓨터로 전달하게 된다. CPU와 네트워크 프로토콜 프로세서내의 DMA제어기는 네트워크 접속장치내의 내부버스를 공유하고 있다. 그결과 수신버퍼를 CPU가 액세스하기 위해서는 망으로부터 FIFO에 입력된 데이터가 없어서 내부버스에대한 DMA제어기로부터의 요청이 없을때만 가능하다. 이에 대한 모델을 그림 2에 나타냈다. 여기서 C_x (Mbits/s)은 망으로부터의 입력속도, C_D (Mbits/s)는 DMA전송속도, C_C (Mbits/s)는 CPU에 의한 데이터 블럭이동속도를 의미한다. FIFO 버퍼로 입력되는 데이터 패킷은 포아손 분포에 따르는 것으로 가정하며 패킷의 길이는 지수분포를 갖는다고 가정한다. 그러면 Burke의 정리¹⁷⁾에 의해 FIFO 버퍼에서의 출력 프로세스, 즉 수신버퍼의 입력 프로세스는 역시 포아손 프로세스라 할 수 있다. 또한 FIFO 버퍼에 대기되는 데이터는 수신버퍼의 데이터보다 우선순위로 서비스를 받기 때문에 FIFO버퍼에서의 오버플로우가 없다는 가정아래 FIFO의 영향을

무시하고 수신버퍼만을 독립적으로 분석할 수가 있다. 단 수신버퍼의 서버에 해당하는 CPU는 FIFO버퍼의 서버인 DMA 제어가 동작할 확률 P_{OFF} 로 랜덤한 서비스 인터럽트를 받으며 P_{ON} 확률로 서비스를 수행할 수 있다고 가정한다. 이러한 가정은 수신버퍼의 패킷 도착과정과 서비스과정이 독립적인 것으로 가정한 것으로 이를 이용한 분석은 실제 시스템의 시뮬레이션을 통한 검증이 필요하다. 하지만 P_{OFF} 가 단순히 FIFO버퍼로부터 수신버퍼로의 DMA에 의한 패킷 이동 이외에 CPU의 다른 처리를 통한 수신버퍼 서비스 중단 등에 의하여 증가될 수 있는 점 등을 고려하면 수신버퍼만을 독립적으로 분석하기 위하여 가정할 수 있다.

위의 시스템 모델과 함께 다음과 같은 가정과 변수를 설정한다.

- (1) 수신버퍼에 입력되는 데이터는 λ (패킷/s)를 변수로 하는 포아송 프로세스이다.
- (2) 도착하는 패킷의 길이는 지수분포를 가지며 도착한 패킷이 일정한 크기의 블록으로 나누어져 있는 수신버퍼의 i 블록을 취할 확률은

$$g_i = p q^{i-1}$$

$$p = \frac{1}{n}, p+q = 1 \quad (1)$$

이 되며, \bar{n} (블록/패킷)은 패킷당 필요한 평균 수신버퍼 블록수이다. 또한 각 패킷의 길이는 i.i.d.(independent identical distribution)를 취한다.

- (3) CPU가 수신버퍼의 한 블록을 이동시키는데 소요되는 시간은 T 슬롯 시간이라 정의한다.
- (4) 수신버퍼내의 블록은 FIFO 서비스를 받으며 서버는 각 타임슬롯동안 하나의 블록을 서비스하며 수신버퍼에서 제거한다.
- (5) 수신버퍼는 유한하며 N 블록의 크기를 갖는다고 가정한다.
- (6) 입력되는 패킷에 대해 수신버퍼내의 비어있는 블록수가 모자를 경우 패킷 전체 블록중 일부만이 받아들여지지 않고 전체가 거부된다. 즉 batch acceptance strategy^[8]가 적용된다.

위의 가정중 (1)과 (2)는 많은 시스템에서 관측 결과를 토대로 모델링하고 있으며^[9], 이러한 가정아래 시

스템의 시간축은 일정크기의 타임슬롯으로 나누어진 형태가 되며 각 타임슬롯의 끝점에서 수신버퍼내의 한 블록이 CPU에 의해 제거되는 것으로 모델링할 수 있다. 이상의 시스템 모델에 대한 분석을 다음절에서 행한다.

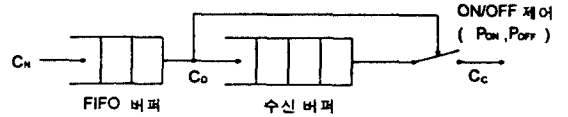


그림 2. FIFO 와 수신버퍼 모델
Fig. 2 Model of FIFO and Receive Buffer.

III. 성능분석

블록의 수신버퍼에서 떠나는 시점인 각 타임슬롯의 끝점은 다음 슬롯의 시작점과 같으므로 π_i 를 슬롯의 시작점에서의 버퍼내의 블록갯수가 i 일 확률, $P_{i,j}$ 를 슬롯 시작점에 버퍼내의 블록 갯수가 j 일때 타임슬롯동안 i 개의 블록을 받아들일 확률이라 정의하면 시스템의 평형상태에서의 상태 방정식은 다음과 같이 된다.

$$\pi_0 = P_{0,0}\pi_0P_{OFF} + P_{0,0}\pi_1P_{ON} + P_{0,0}\pi_0P_{ON}$$

$$\pi_1 = P_{1,0}\pi_0P_{ON} + P_{1,1}\pi_1P_{ON} + P_{0,2}\pi_2P_{ON} + P_{1,0}\pi_0P_{OFF} + P_{0,1}\pi_1P_{OFF}$$

$$\pi_i = P_{i,0}\pi_0P_{ON} + \sum_{k=0}^{i-1} P_{i-k,k+1}\pi_{k+1}P_{ON} + \sum_{k=0}^i P_{i-k,k}\pi_kP_{OFF} \quad (2)$$

$$i = 0, 1, 2, \dots, N.$$

이와 함께

$$\sum_{i=0}^N \pi_i = 1 \quad (3)$$

의 조건을 갖는다. 이때 P_{OFF} 는 그림 2에서 FIFO서버가 동작중일 확률로서 트래픽의 실효율을 U 라면 망으로부터의 실효 입력속도는

$$C_N^* = uC_N \quad (4)$$

이 된다. 또한 DMA속도가 C_D 이므로

$$P_{OFF} = C_N^*/C_D \quad (5)$$

$$P_{ON} + P_{OFF} = 1$$

이 된다. 그러므로 (2)식에서 $P_{i,j}$ 를 구하면 π_i 를 구할 수 있다. 그러나 임의의 갯수와 크기의 패킷이 한 타임

슬롯동안에 도착할 수 있고 수신버퍼는 batch acceptance strategy를 사용하므로 실제 모든 경우를 고려하여 $P_{i,j}$ 를 구하기는 어렵다. 그러므로 본 논문에서는 각 타임슬롯을 M개의 작은 소 슬롯으로 분해하여 각 소슬롯에 도착하는 패킷을 구하여 $P_{i,j}$ 를 구하는 방법을 이용한다^[10].

M개로 분해된 소슬롯에서 임의의 소슬롯에 i개의 패킷이 도착할 확률을 f_i 라 하면

$$f_i = \frac{(\lambda T/M)^i}{i!} e^{-\lambda T/M} \quad (6)$$

이 된다. 이때 M이 충분히 크다고 하면 $i \geq 2$ 에 대한 f_i 는 무시될 수 있다. 즉 각 소슬롯에 도착할 패킷수는 한개를 넘지 않는다고 가정할 수 있다. 이상의 가정아래 다음을 정의한다.

$Z_m \equiv$ m번째 소슬롯의 종점에서의 버퍼내의 블럭수
($1 \leq m \leq M$)

$Z_0 \equiv$ 타임슬롯의 시작점에서의 버퍼내의 블럭수

$a_{i,j}(m, n) \equiv$ Probability [$Z_n = j | Z_m = i$] (7)

$$a_{i,j} \equiv a_{i,j}(n, n+1) \quad (8)$$

여기서

$$a_{i,j} = \begin{cases} f_0 + f_1 \bar{g}_{N-i}, & i=j \\ f_1 g_{j-i}, & j>i \\ 0, & \text{그밖의 경우} \end{cases} \quad (9)$$

이 된다. 이때 \bar{g}_i 는 패킷의 크기가 i개의 블럭보다 클 확률로서

$$\bar{g}_i = \begin{cases} 1 - (g_1 + g_2 + \dots + g_i), & i \geq 1 \\ 1, & i = 0 \end{cases} \quad (10)$$

이 된다. $a_{i,j}$ 를 이용하여 $P_{i,j}$ 는 다음과 같이 구해진다.

$$P_{i,j} = a_{j,i+j}(0, M) \quad (11)$$

$a_{i,j}$ 의 행렬 [$a_{i,j}$]를 A , [$a_{i,j}(0, M)$]를 $A(0, M)$ 라 하면

$$A(0, M) = A^M \quad (12)$$

이 되고 (9)식과 (12)식에 의해 (11)식의 $P_{i,j}$ 가 구해진다. 또한 $P_{i,j}$ 가 구해지면 π_i 가 구해진다.

이상에서 구한 π_i 를 이용하여 패킷의 손실률을 다음에서 구한다. 먼저 PB' 를 타임슬롯 시작후 t초후에 도

착한 패킷이 손실될 확률, π_i' 를 타임슬롯 시작후 t초후에 시스템의 블럭수가 i일 확률, PB 를 임의의 시간에 도착한 패킷이 손실될 확률이라 하면

$$PB' = \sum_{i=0}^N \pi_i' \bar{g}_{N-i} \quad (13)$$

$$PB = \frac{1}{T} \int_0^T PB' dt = \sum_{i=0}^N \frac{1}{T} \left(\int_0^T \pi_i' dt \right) \bar{g}_{N-i} \quad (14)$$

가 된다. 한편 $0 \leq t < T$ 시간에는 블럭처리가 끝나지 않으므로 식(2)에 $P_{ON}=0, P_{OFF}=1$ 을 대입하여 π_i' 의 평형상태 방정식(15)를 얻게된다.

$$\pi_i' = \sum_{k=0}^i P'_{i-k, k} \pi_k \quad ; \quad i=0, 1, \dots, N, \quad (15)$$

$$\sum_{i=0}^N \pi_i' = 1.$$

여기서 $P'_{i,j}$ 는 슬롯시작시 j블럭이었던 수신버퍼가 t초후에 i블럭을 받아들일 확률을 의미한다.

$$\pi_i^* = \frac{1}{T} \int_0^T \pi_i' dt \quad (16)$$

$$P'_{i,j} = \frac{1}{T} \int_0^T P'_{i,j} dt$$

이라 정의하면 식 (15)로부터

$$\pi_i^* = \sum_{k=0}^i P'_{i-k, k} \pi_k, \quad i=0, \dots, N \quad (17)$$

이 되고 식(14)의 PB 는

$$PB = \sum_{i=0}^N \pi_i^* \bar{g}_{N-i} \quad (18)$$

가 된다.

한편 M이 충분히 크다고 가정하면

$$P'_{i,j} = \frac{1}{T} \int_0^T P'_{i,j} dt \cong \frac{1}{T} \sum_{k=0}^M a_{j,i+k}(0, k) \quad (19)$$

이 되므로 (19)식을 (17)식에 대입하여 π_i^* 를 구하고 이로부터 다시 (18)식을 이용하여 패킷의 손실률 PB 를 구하게 된다.

M은 식(6)에서 $\sum_{i=2}^{\infty} f_i \leq \epsilon$ 되게 결정할 수 있으며 등가적으로 $1 - f_0 - f_1 \geq 1 - \epsilon$ 로부터 결정할 수 있다. 이상의 결과식을 이용하여 임의의 실효 입력속도에 대한 수신버퍼에서의 패킷 손실율을 다음절에서 구해 본다.

IV. 수치적 결과

본 절에서는 수신버퍼 블럭이 기본 크기가 100 바이트, DMA속도 C_D 와 CPU의 블럭 이동 속도 C_C 가

각각 40 Mbits/s인 네트워크 접속 어댑터에 대해 망으로부터의 실효 데이터 입력 속도 C_N^* 와 패킷의 평균 길이 $1/p$ 를 달리할때 수신버퍼에서의 패킷 손실율을 살펴 보았다. 먼저 평균 패킷 길이가 400바이트, 즉 $1/p$ 이 4인 경우 실효 입력 속도 C_N^* (Mbits/s)가 각각 16,10,1일때 수신버퍼크기 N (블럭)에따른 패킷 손실율을 그림 3에 나타냈다. 이때 소슬롯의 수 M 의 크기는 ϵ 이 10^{-6} 보다 작도록 실효 입력속도에 따라 32와 64로 정했다. 마찬가지로 평균패킷의 길이가 200바이트, 즉 $1/p$ 이 2일 때의 결과를 그림 4에 보였

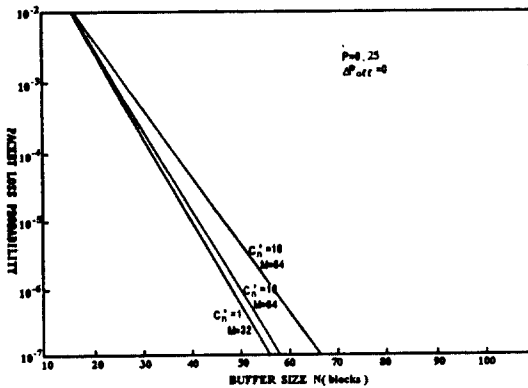


그림 3. 각 트래픽 밀도 C_N^* (Mbits/s) 에서 버퍼크기에 대한 패킷 손실율 (CPU에 다른 일이 없고 $p = 0.25$ 일때)
 Fig. 3. Packet loss probability versus buffer size with various traffic intensity C_N^* (Mbits/s) (without other jobs for CPU, $p = 0.25$).

이상에서 각 C_N^* 에 대한 수신버퍼의 서버중단 확률 P_{OFF} 는 식(5)에 의해 각각 0.4, 0.25, 0.025가 되며

$$\lambda = C_N^*/\text{평균프레임길이} = pC_N^*$$

$$T = \text{한블럭길이}/C_c = 100/C_c$$

로부터 $\lambda T = 100pC_N^*/C_c$ 가 된다.

그림에서 보듯이 패킷 손실율은 $\log N$ 에 반비례하고 있으며 그림을 비교하여 볼때 같은 트래픽 밀도에 대해 평균 패킷길이 $1/p$ 이 클수록 손실율이 증가함을 볼 수 있다. 이것은 평균 길이가 커질수록 패킷당 필요한 수신버퍼의 블럭수가 증가하여 그만큼 패킷 전체가 받아들여질 확률이 감소되리라는 점에서 이해될 수 있다.

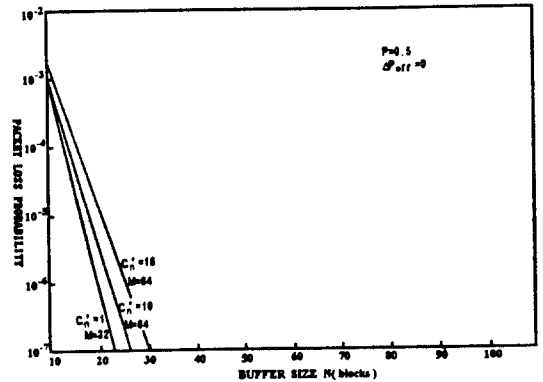


그림 4. 각 트래픽 밀도 C_N^* (Mbits/s) 에서 버퍼크기에 대한 패킷 손실율. (CPU에 다른 일이 없고 $p = 0.5$ 일때)
 Fig. 4. Packet loss probability versus buffer size with various traffic intensity C_N^* (Mbits/s). (without other jobs for CPU, $p = 0.5$)

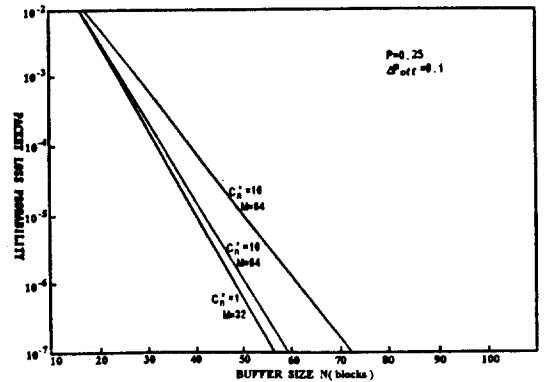


그림 5. 각 트래픽 밀도 C_N^* (Mbits/s) 에서 버퍼크기에 대한 패킷 손실율. (CPU에 다른 일이 있고 $p = 0.25$ 일때)
 Fig. 5. Packet loss probability versus buffer size with various traffic intensity C_N^* (Mbits/s). (with other jobs for CPU, $p = 0.25$)

한편 수신버퍼의 서버인 CPU의 서비스 중단 확률 P_{OFF} 가 CPU의 다른 처리로 인해 증가되는 경우를 생각할 수 있다. 즉 식(5)에서 P_{OFF} 는 망으로부터의 패킷 입력속도에 의해서만 결정되었으나 네트워크 접속장치에 다른 처리할 일이 있을경우 이는 CPU의 패킷 블럭 이동 일 이외의 오버헤드가 되어 C_N^* 에 관계없

이 P_{OFF} 가 증가될 것이다. 이때의 패킷 손실율의 변화를 그림 5, 그림 6에 $p=0.25, 0.5$ 일때 각각 나타냈다. 그림에서 P_{OFF} 는 원래의 값보다 0.1만큼 증가된 경우이며 C_M^* 와 다른 변수들의 값은 그림 3.4와 같다. 각 그림을 비교하여 볼때 C_M^* 가 클경우 P_{OFF} 증가에 의한 패킷 손실률 증가가 현저하며 C_M^* 가 작을 경우는 거의 영향이 없음을 볼 수 있다. 즉 CPU의 다른 처리할 일이 추가될 경우 허용된 패킷 손실율을 만족시키기 위해서는 버퍼의 용량을 증가시키거나 CPU의 속도를 증가시켜야함을 알 수 있다. 이상의 결과를 가지고 주어진 망 트래픽 및 CPU 속도에 대해 필요한 버퍼의 크기를 선택할 수 있을 것이다.

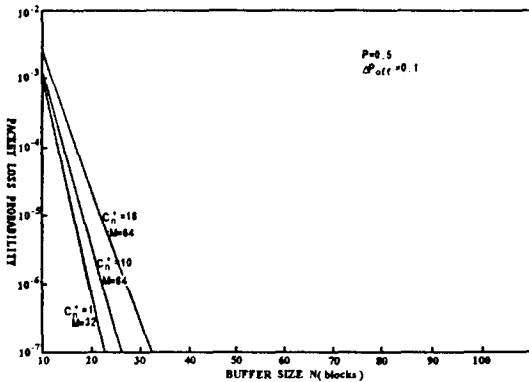


그림 6. 각 트래픽 밀도 $C_M^*(Mbits/s)$ 에서 버퍼크기에 대한 패킷 손실율 (CPU에 다른 일이 있고 $p = 0.5$ 일때)

Fig. 6. Packet loss probability versus buffer size with various traffic intensity $C_M^*(Mbits/s)$. (with other jobs for CPU, $p = 0.5$)

V. 결 론

본 논문에서는 컴퓨터 통신망에서 컴퓨터를 네트워크에 접속시켜 주는 네트워크 접속장치내의 버퍼에 대한 용량분석을 수행하였다. 네트워크 접속장치는 컴퓨터의 부담을 줄여주기 위해 마이크로프로세서와 네트워크 접속용 각종 통신 프로토콜을 처리하는 프로토콜 프로세서가 내장된 형태의 지능형 어댑터를 다루었다.

CPU와 프로토콜 프로세서내의 DMA제어기는 어댑터내의 내부버스를 서로 공유하며 작업함으로써 패킷의 수신버퍼의 제한된 용량에 따른 패킷 손실율에 영향을

주게된다. 이때 분석을 위한 모델로는 좀더 실제 시스템에 접근하기 위하여 버퍼의 구성이 블럭화 되어있음을 주목하여 CPU에 의한 수신버퍼에서의 블럭이동 및 처리를 서비스시간이 일정 시간간격(타임슬롯)으로 나누어진 것으로 모델링하여 분석하였다. 즉 버퍼의 서비스는 각 타임슬롯의 종점에서 이루어지는 단속 서비스 시간모델로 부터 평형상태의 버퍼 점유상태를 구하여 패킷의 손실율을 구하였다.

이때 각 타임슬롯동안에 도착하는 패킷의 수를 구하기 위해 각 슬롯을 여러개의 소슬롯으로 나누어 패킷의 도착 프로세스를 단순화하였고 그 결과로 각 트래픽 밀도에 대한 패킷 손실율을 구할 수 있었다.

참 고 문 헌

- [1] A. S. Krishnakumar, K. Sabnani, "VLSI implementations of communication protocols - a survey," IEEE JSAC, Vol. 7, No. 7, Sept. 1989.
- [2] L. Svobodova, "Implementing OSI systems," IEEE JSAC, Vol. 7, No. 7, Sept. 1989.
- [3] The AM7990 Family IEEE-802.3, Advanced Micro Devices, 1984.
- [4] H. Ichikawa et al., "Protocol control VLSI for broadband packet communications," in Proc. Globcom, Nov. 1988.
- [5] ATM Adapter: Design and Architecture, Fore Systems, 1994.
- [6] J. W. Wong, W. Bux, "Analytic modelling of an adapter to local area networks," IEEE Trans. on Commun. Vol. COM-32, No. 10, Oct. 1984.
- [7] L. Kleinrock, Queueing systems Vol. I: Theory, John Wiley & Sons, 1975.
- [8] D. R. Manfield, D. Tran-Gia, "Analysis of a finite storage system with batch input arising out of message packetization," IEEE Trans. Commun. Vol. COM-30, Mar. 1982.
- [9] E. Fuchs, P. E. Jackson, "Estimates of distributions of random variables for certain computer-communications traffic models," Commun. Ass. Computer Vol.

13. Dec. 1970.

[10] H. Bruneel, B. G. Kim, Discret-Time
Models for Communication Systems

Including ATM, Kluwer Academic Pub.,
1993.

— 저 자 소 개 —



金 永 翰(正會員)

1962년 2월 12일생. 1984년 2월 서울대학교 전자공학과 졸업(공학사). 1986년 2월 한국과학기술원 전기 및 전자공학과 졸업(공학석사). 1990년 8월 한국과학기술원 전기 및 전자공학과 졸업(공학박사). 1987년 1월 ~ 1994년 8월 디지콤정보통신연구소 데이터통신연구부장. 1994년 9월 ~ 현재 숭실대 정보통신공학과 조교수. 주관심분야는 ATM 네트워크 및 멀티미디어 통신 프로토콜 등임