

# X.25와 SDLC간 Protocol Converter의 성능 분석에 관한 연구

正會員 鄭 海\* 正會員 殷 鍾 官\*

## Performance Analysis of A Protocol Converter between X.25 and SDLC

Hae CHUNG\*, Chong Kwan UN\* *Regular Members*

**要 約** 본 논문에서는 CCITT에서 채택된 X.25 network과 IBM이 채택한 system network architecture(SNA)의 synchronous data link control(SDLC)간의 protocol converter의 성능 분석을 한다. 여기서 protocol converter는 link-level mapping method를 사용하여 구현 되었다고 가정한다. 성능 분석을 통하여 각종 parameter에 대한 throughput과 mean waiting time을 구하며, 다양한 parameter에 대하여 최적의 값을 조사한다. 특히 converter가 수신한 SDLC frame이 X.25 network에서 규정된 최대 data packet size 보다 더 큰 경우에 frame을 분할하게 되는데, 분할방법에 있어서 full and remainder packetizing과 equal packetizing 도입하여 서로간의 성능을 비교한다. message를 분할 할때 늘 손실으로 나누는 것이 frame의 오류율을 줄일 수 있기 때문에 전자보다 후자가 성능면에서 우수함을 알 수 있다.

**ABSTRACT** In this paper, we analyze the performance of a protocol converter between X.25 and Synchronous Data Link Control(SDLC) adopted by CCITT and IBM System Network Architecture(SNA). Here, we assume that the protocol converter is realized by using the link level mapping method. We obtain the throughput and the mean waiting time which are represented as a function of several parameters, and investigate the optimal values of various parameters. If the length of SDLC frame is longer than the maximum packet size defined in X.25, it must be divided to pass it through the X.25 network. Therefore, we introduce a scheme of full and remainder packetizing and a scheme of equal packetizing as the division methods, and compare the performances of the two methods. Because the equal packetizing scheme has decreasing effect of frame error probability, it turns out that the latter is better than the former in performance.

### I. 서 론

Computer의 보급이 확산되고 업무구조가 세분화되어짐에 따라, 서로 다른 network간의 정보교환 및 resource를 활용하기 위하여 그것들끼리 연결해 줄 수 있는 heterogenous network 구성의 필요성이 요구되고 있다. 이러한 서로 다른 구조를 갖는 network의 상호 접속을 위하여 protocol conversion이 필요하며, 그것은 computer communication의 발전과 융통성을 위하여 중요하지만 어려운 작업이다. 그리고 conversion에

있어서 간과할 수 없는 것이 성능 분석에 관한 연구이다. 그러나 그것에 대한 연구는 거의 존재한 상태이다[1]. 물론 homogeneous network에서 통신하는 것보다 conversion 과정에 있어서 과외의 processing time과 overhead의 증가로 인한 성능의 저하는 불가피하다[3]. 그래서 만약에 converter가 구현되었다 할지라도 같은 구조를 갖는 network상에서 통신할 때보다 지나치게 기능, 성능 면에서 열등하다면 converter가 도입될 수 밖에 없을 것이다.

본 논문에서는 protocol conversion에 대한 근간의 연구들[1],[2] 토대로 하여, CCITT에서 표준약으로 제시한 packet switched data network (PSDN)에 사용되는 X.25와 [4] [6] 전세

\*韓國科學技術院電氣및 電子工學科  
Department of Electrical and Electronic Engineering,  
KAIST  
論文番號 : 91-77 (接受1991. 3. 25)

계적으로 가장 널리 보급되어온 IBM SNA의 2 layer protocol인 synchronous data link control (SDLC) [7] [9]간을 연결하는 protocol converter의 성능 분석을 통하여 각종 parameter가 converter의 성능에 미치는 영향을 조사한다. 여기서 protocol converter는 link level mapping method를 이용하여 구현되었다고 가정한다[10]. 그것은 SNA의 2 layer를 X.25의 3 layer로 mapping 시키는 방법으로써, SNA node로 하여금 X.25 virtual circuit을 real circuit으로 간주하게 하여, X.25 network을 통해 SNA host인 두 end user간의 통신이 가능하게 하는 것이다.

성능 분석에 있어서, bit error probability, propagation delay, message length등 각종 parameter의 값에 따른 throughput과 mean waiting time을 구하여 비교하며, 특히 message 분할 방식에 있어서 full and remainder 방식보다 equal packetizing 방식이 훨씬 성능면에서 우월하다는 것을 보일 것이다.

## II. 시스템 모델링

X.25 link level에서 사용하는 protocol은 High Level Data Link Control(HDLC)로서 그 성능 분석은 이미 잘 알려져 있다[11][12]. 그러나 본 논문에서 구현한 converter의 성능 분석은 그것과는 크게 두가지 차이점을 들 수 있다. 첫째, SDLC의 host에서 송출하는 data packet의 길이가 X.25 link level에 규정된 길이보다 길때는 packet을 둘 이상의 packet으로 나누어야 한다는 점에서 arrival process가 batch arrival로 간주되어야 한다는 점이다. 둘째, converter는 SDLC host의 departure process를 arrival process로 받아들이고 있어서 엄밀하게 tandem queue로 해석해야 한다는 것이다.

본 논문에서는 전자의 조건은 충족시키고 후자의 조건은 근사시키는 방향으로 성능 분석을 하기로 한다. 그림 1에서 보여지는 바와 같이 link 1은 SDLC protocol로 제어되고 link 2는

HDLC protocol로 제어되며 두 link 모두가 full duplex circuit로 구성된다. Converter나 X.25 node 모두가 무한 buffer라고 가정하며 first come / first serverd (FCFS)에 따라 전송된다. 또한 message의 길이는 일정한 길이  $l$ (bits)이라 가정하며, 전송 채널의 특성으로써 전송율  $v$ , bit error probability  $P_b$ (단 bit error는 상호 독립적), 전송지연(propagation delay)  $D$ , 또한 수신된 frame의 처리시간(processing time)을 일정한 상수  $P$ 로 규정한다. 그래서 해석을 용이하게 하기 위하여 새로운 선택적인 상수

$$t_p = D + P \tag{1}$$

를 도입한다. 그리고 한 frame이 error가 일어난 확률을  $P_f$ 라고 할때,

$$P_f = 1 - (1 - P_b)^{l+48} \tag{2}$$

이 된다. 여기서  $l+48$ 은 한 frame이 overhead를 포함한 전체 bit 수이며 48은 overhead를 의미한다.  $t$ 를 Information frame(I frame)의 전송시간(transmission time)이라 할때

$$t_I = (l + 48)/v \tag{3}$$

가 되며,  $t_s$ 를 supervisory frame(S frame)의 전송시간이라 할때

$$t_s = 48/v \tag{4}$$

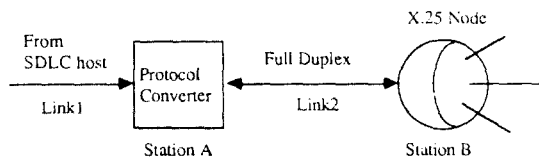


그림 1. Data-Link 모델의 구조  
Fig. 1. Structure of the data link model.

가 된다. 단 여기서 모듈러스(modulus) M을 8로 가정한다.

I-frame을 error없이 송신한 직후 acknowledgement 가 수신될 때까지 걸리는 시간을 acknowledgement time  $t_{ack}$ 라 정의하며 S frame에 error가 발생하는 상황을 무시할 때 다음과 같이 표기할 수 있다.

$$t_{ack} = 2t_p + t_s \quad (5)$$

### III. 성능 해석

#### 1. 최대 throughput의 계산

여기에서는 주어질 link 2의 channel 특성상에서 정보 frame(information frame, I frame)에 대한 최대 throughput을 구하는 것이 목적이다. 최대 throughput이라 함은 traffic의 upper bound를 의미하며 여기서는 station A가 항상 보낸 정보를 가지고 있는 반면 B에서 A로 향한 channel은 idle한 상태로 본다. 우선 converter가 HDLC protocol로 송신할 때 최대 throughput을 구하고 그 다음 protocol conversion했을 경우의 그것을 구하며 서로 비교한다.

#### (1) Balanced HDLC의 최대 Throughput $T_H$

Throughput을 구하기 위해서는 송신단에서 하나의 I frame에 대하여 error 발생상황을 고려한 service time을 계산해야 하는데, 그것을 구하기 위해 virtual service time의 개념을 도입한다. Virtual transmission time은  $N(s)=i-1$ 이 전송에러없이 순서대로 수신되었다는 가정하에  $N(s)=i$  frame을 전송하는 순간부터 I frame이 성공적으로 송신되었을 경우 송신단에서 I frame ( $N(s)=i$ ) 전송이 끝나는 순간까지를 의미한다 [11][12]. 그래서 송신단에서는 inter departure time으로 보이고 수신단에서는 inter arrival time으로 보인다. 만약  $M-1$ 개의 unacknowledged I frame들에 의해  $N(s)=i$ 가 전송될 수 없다면 virtual transmission time은 window가 다시 열린

때까지 지연되어 길어질 것이다. 이 virtual transmission time의 평균을  $t_v$ 라고 할 때 maximum information의 throughput  $T_H$ 는

$$T_H = 1/t_v \quad (6)$$

로 주어진다. 결국 saturated된 경우의 평균 virtual transmission time을 구하면 maximum throughput이 구해진다. 여기서 평균 virtual transmission time을 구하기 위해 virtual transmission time  $T$ 가 window width  $w$ 에 종속적인 경우와 비종속적인 경우 두가지를 고려하기로 한다. Window width와 virtual transmission time이 시작되는 순간 acknowledgement frame을 받지 않고 전송할 수 있는 I frame의 최대수를 말한다.

첫째, window width에 종속적인 경우는 일반적으로 packet의 round trip delay가 full window packet의 수( $M-2$ )보다 클 경우로써 flow control은 window width에 종속적인 것으로 알려져 있다.[13]. 그래서  $t_{ack}$ 가  $(M-2)t$ 보다 클 경우 virtual transmission time은 window width에 종속적이다. 이 경우에 mean virtual transmission time은

$$t_v = \sum_{w=0}^{M-2} \phi(w)t_v(w) \quad (7)$$

로 주어진다[11]. 여기서

$$\phi(w) = \frac{P_F(1-P_F)^{M-2-w}}{1-(1-P)^{M-1}} \quad \text{for } w=0,1,\dots,M-2 \quad (8)$$

$$t_v(w) = E[T_v(w)]$$

$$= E[T_o(w)] + P_F E[T_1(w)] + \frac{P_B^2}{1-P_B} t_2 \quad (9)$$

이다. 이때 제 일항의 의미는 I frame에 error가 없을 때의 mean virtual transmission time이고, 제 이항은 I frame에 단 한번의 error가 발생했을 때 mean virtual transmission time이며, 최종항은 두번이상 error가 났을 경우의 mean virtual transmission time을 의미한다.

둘째, virtual transmission time이 window width에 비종속적인 경우는  $t_v$ 가  $(M-2)t$  보다 작거나 같을 경우로써, virtual transmission time은 window width에 비종속적이며[13]

$$t_v = E[T_v] = t_f + P_B E[T_1] + \frac{P_b^2}{1-P_b} t_2 \quad (10)$$

로 주어진[11][12], 이 때에도 각 항의 의미는 이미 설명한 바와같다.

### (2) Converter의 Maximum Throughput

Converter는 그림 1에서 보는 바와 같이 link 2에서 HDLC protocol을 사용하고 있으므로 virtual transmission time의 개념은 그대로 적용된다. 그러므로 앞 절에서 구한  $t$ 를 이용하여 maximum throughput을 구해보자. 우선 분석하기 전에 (7)식에서 (10)식에 나타나 있는 parameter들을 관찰하면  $t$ 는 message length  $l$ 의 함수임을 알 수 있다. 그래서 해석의 편의상  $t$ 를  $l$ 의 함수, 즉  $t(l)$ 로 표기하기로 한다.

그림 2에서 정상적으로 송·수신할 때 SDLC와 converter의 I frame의 구조를 보여주고 있는데, converter에서 송출하는 I-frame의 overhead는 SDLC host에서 송출하는 SDLC I frame의 overhead보다 증가한다. 이 그림에서 보듯이 SDLC I-frame에서 address field와 I field만이 converter의 data packet에 mapping 된다. 그래

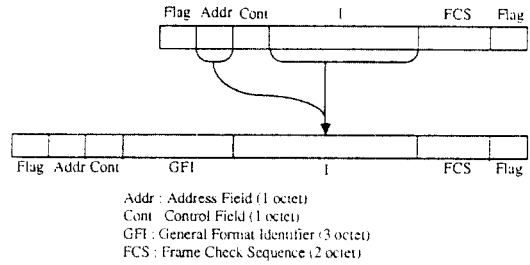


그림 2. I frame 구조의 비교  
Fig. 2. Comparison of I frame structure.

서 SDLC의 overhead는  $48/(l+48)$ 인 반면 I frame이 converter의 frame으로 mapping될 때, SDLC host의 관점에서 보면  $80/(l+80)$ 으로 증가한다. 그러나 SDLC의 I-frame이 X.25 network에 규정된 data packet의 길이를 초과할 경우 2개 이상의 I frame으로 나누어 주어야 하므로, 실제의 overhead는 더욱 증가하게 된다. SDLC의 message length를  $l$ , X.25 network에 규정된 data packet의 length를  $l_n$ , SDLC의 address field의 length를  $l_a$ , X.25의 General Identifier Format(GFI) field의 length를  $l_{GFI}$ 라 할때 임의의  $l$ 에 대하여 converter가 분할하는 I frame의 수를  $k$ 라고 놓으면,

$$k = \left\lceil \frac{l_s + l_a}{l_H - l_{GFI}} \right\rceil + 1 \quad (11)$$

이 된다. 그래서 SDLC의 I-frame의 관점에서 converter가 송출할 때 덧붙는 head는 총  $(48 + l_{GFI})k + l_a$ 가 된다. 그러므로

$$overhead = \frac{(48 + l_{GFI})k + l_a}{l_s + (48 + l_{GFI})k + l_a} \quad (12)$$

가 된다. 그러면 이제  $k$ 개의 I-frame으로 분할하는 방식에 대해 고려해 보기로 하자. 첫번째 방식은  $l + l_a$ 가  $l_n - l_{GFI}$ 를 초과하여  $k$ 개의 frame

으로 분할될 때  $k-1$ 개의 frame은  $(l_H - l_{GF})$ 의 length를 갖게하고 나머지 bit들은 마지막 frame에 집어넣는 방식이다. 두번째는  $k+l$  bit를 모두  $k$ 등분하여 각각을  $k$ 개의 frame을 만들어 주는 방식이다. 전자를 full and remainder packetizing이라하고 후자를 equal packetizing이라 부르게 하자.

Full-and remainder packetizing 방식으로 message를 분할할 때 converter가 취하는 mean virtual transmission time을  $t_{CFR}$ 이라 할 때

$$t_{CFR} = (k-1)t_v(l_H) + t_v(l_s + l_a - (k-1)(l_H - l_{GF}) + 24) \quad (13)$$

가 될 것이다.

$$T_{CFR} = \frac{l_s}{t_{CFR}} \quad (14)$$

Equal packetizing 방식으로 message를 분할할 때 converter가 취하는 mean virtual transmission time을  $t_{CE}$ 라 하고 maximum throughput을  $T_{CE}$ 라 하고 할 때

$$t_{CE} = k \cdot t_v \left( \frac{l_x + l_a}{k} + 24 \right) \quad (15)$$

$$T_{CE} = \frac{l_s}{t_{CE}} \quad (16)$$

### 2. Mean Waiting Time의 해석

앞서 언급했던 마와 같이 SDLC host에서 송출하는 frame이 X.25 link level에 규정된 길이보다 길 때는 둘 이상의 frame으로 나누어야

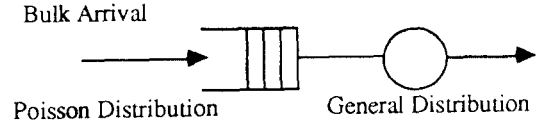


그림 3. Converter의 queuing 모델  
Fig. 3. Queuing model of converter.

하므로 converter에 들어오는 arrival process는 batch로 간주하고 Poisson 분포를 갖는다고 가정하자.

그림 3에서 보는 마와 같이 converter로 arrival하는 process는 arrival rate가  $\lambda$ 인 Poisson bulk로 간주할 수 있고 service distribution이 일반적인  $B(x)$ 가 될 것이다. 이때 mean queue length와 mean waiting time은 다음과 같다[14].

$$\bar{q} = \rho + \frac{\rho^2(1+C_b^2)}{2(1-\rho)} + \frac{\bar{g}-1}{2(1-\rho)} \quad (17)$$

$$W = \frac{\rho \bar{g} \bar{x}}{2(1-\rho)} \left( 1 + C_g^2 + \frac{C_b^2}{g} \right) \quad (18)$$

여기서  $\bar{q}$ 는 mean queue length,  $W$ 는 mean waiting time,  $\bar{x}$ 는 mean service time  $\bar{g}$ 는 mean batch number,  $C_g^2$ 과  $C_b^2$ 은 각각 squared coefficient of variation으로서 다음과 같이 주어진다.

$$C_g^2 = \frac{\sigma_g^2}{\bar{g}^2}, \quad \sigma_g^2 \text{은 } g \text{의 분산} \quad (19)$$

$$C_b^2 = \frac{\sigma_b^2}{\bar{x}^2}, \quad \sigma_b^2 \text{은 } x \text{의 분산} \quad (20)$$

또한  $\rho$ (utilization factor)는 system이 busy할 확률로써

$$\rho = \lambda \bar{g} \bar{x} \quad (21)$$

로 주어진다. 앞에서 가정했던 것처럼 SDLC host에서 송출하는 message의 길이를 일정한 length  $l$ 라고 한다면 converter에서 분할하는 frame의 수  $k$  역시 일정한 상수가 될 것이다. 그러므로

$$\bar{g} = k \quad (22)$$

$$\bar{g}^2 = k^2 \quad (23)$$

$$C_g^2 = 0 \quad (24)$$

$$\rho = k \lambda \bar{x} \quad (25)$$

이 된다.  $k$ 는 (11)식에 주어졌듯이이다. 결국

$$\bar{q} = \rho + \frac{\rho^2(1+C_b^2)}{2(1-\rho)} + \frac{k-1}{2(1-\rho)} \quad (26)$$

$$W = \frac{k \bar{x} \rho}{2(1-\rho)} \left(1 + \frac{C_b^2}{k}\right) \quad (27)$$

이 된다. 그래서 converter로 들어오는 arrival process가 Poisson bulk일 경우 service time의 1,2차 moment만 구한다면 mean queue length와 mean waiting time을 구할 수 있다.

우리는 service time의 moment를 구하기 위하여 virtual transmission time의 개념을 이용할 수 있다. Service time은 protocol의 선지에서 볼 때 I-frame의 transmission time 뿐만 아니라 S-frame sequence error에 의한 error recovery 과정, 그리고 time out에 의한 error recovery 과정등 모든것을 포함하고 있는데, 이 virtual transmission time은 모든 과정을 다 수용하고 있기 때문이다. 그러므로 우리는 다음 등식이

성립함을 알 수 있다.

$$\bar{x} = E[T_v]$$

$$\bar{x}^2 = E[T_v^2]$$

여기서  $E[T_v]$ 와  $E[T_v^2]$ 은 virtual transmission time의 1,2차 moment로서 [11]에 주어진 결과를 이용하기로 한다.

Virtual transmission time의 1,2차 moment는 message length  $l$ 의 함수로 나타낼 수 있는데 그것을 각각  $m_1(l)$ ,  $m_2(l)$ 이라고 하자.

Full-and-remainder packetizing에서 service time의 1,2차 moment를 각각  $m_{CF1}$ ,  $m_{CF2}$ 라고 한다면

$$m_{CF1} = \frac{(k-1)m_1(l_H) + m_1(l_s + l_a - (k-1)(l_H - l_{GFI}) + 24)}{k}$$

$$m_{CF2} = \frac{(k-1)m_2(l_H) + m_2(l_s + l_a - (k-1)(l_H - l_{GFI}) + 24)}{k}$$

가 될 것이다. 그래서 이 방식으로 packet을 분할할 때 mean queue length와 mean waiting time을 각각  $q_{CF}$ ,  $W_{CF}$ 라고 한다면

$$\bar{q}_{CF} = \rho_{CF} + \frac{\rho_{CF}^2(1+C_{CF}^2)}{2(1-\rho_{CF})} + \frac{k-1}{2(1-\rho_{CF})}$$

$$W = \frac{k m_{CF1} \rho_{CF}}{2(1-\rho_{CF})} \left[1 + \frac{C_{CF}^2}{k}\right]$$

로 표현된다. 여기서

$$\rho_{CF} = k \lambda m_{CF1}$$

$$C_{CF}^2 = \frac{m_{CF2}}{m_{CF1}^2}$$

Equal packetizing 방식에서 service time의 1,2차 moment를 각각  $m_{CE1}$ ,  $m_{CE2}$ 라고 한다면

$$m_{CE1} = m_1 \left( \frac{l_s + l_a}{k} + 24 \right)$$

$$m_{CE2} = m_2 \left( \frac{l_s + l_a}{k} + 24 \right)$$

로 주어지며 mean queue length와 mean waiting time은 (34)식과 (35)식으로 주어지며 이때

$$\rho = k \lambda m_{CE1}$$

$$C_b^2 = \frac{m_{CE2}}{m_{CE1}^2}$$

$$\bar{x} = m_{CE1}$$

이다.

#### IV. 수치해석의 결과

여기서는 주어지는 각종 parameter, message length  $l$ , processing 및 propagation delay  $t$ , transmission rate  $\nu$ , bit error probability  $P_b$

등과 message length의 변화에 따른 throughput과 mean waiting time의 결과를 보여주고 있다.

##### 1. Maximum Throughput

수치 해석을 위해서 X.25가 규정된 전형적인 modulus  $M$ 을 8로 놓는다. 원래의 HDLC protocol에 대한 throughput이 그림 4에 나타나 있다. 전체적인 특징은 short message에 대해서는 message length에 비해 frame head가 차지하는 비율이 높은 편이어서 throughput이 낮고, long message에 대해서는 수신선  $l$ 에 대해 frame error probability가 증가하여 throughput이 저하되는 특징을 보여 적절한 message length가 존재함을 보여준다. Transmission rate  $\nu$ 가 증가함에 따라 더 짧은 길이의 message에 대해 저하되어 낮은 throughput을 보이게 하는 것은 message를 빠른 속도로 송출하는 반면 Ack를 기다리는 시간이 상대적으로 길기 때문이다.

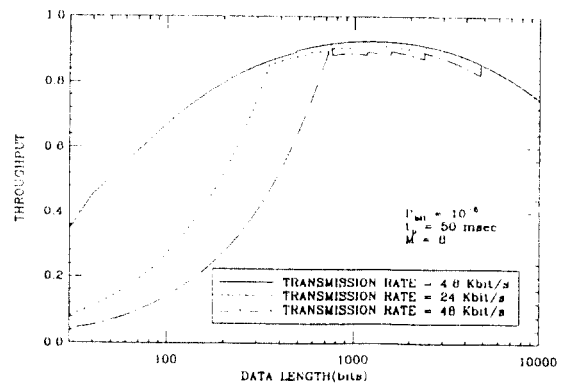


그림 4. Transmission rate에 따른 HDLC의 전형적인 throughput  
Fig. 4. Typical throughput of HDLC versus transmission rate.

그림 5는  $\nu$ 를 4.8K bits/sec로 고정시키고 HDLC가 지니는 throughput과 converter가 가지는 throughput을 비교하고 있다. 이때 X.25가 제공하는 최대 information field의 길이는 HDLC

에서 최대의 throughput을 산출하는 message length에서 산출하였다. 이때의 message length를  $l_m$ 라고 할 때, 이 그림에서 짧은 message에 대해서는 conversion을 통하여 부피되는 head로 말미암아 throughput이 감소하고, 긴 message에 대해서는 frame을 분할하여 전송함으로써 frame-error probability를 감소시킴으로써 HDLC의 throughput보다 오히려 향상되는 것을 관찰할 수 있다. 또한  $l_m$ 의 배수 간격으로 불연속점이 발견되는데, 이것은 converter가 message를 두개의 frame 이상으로 분할함으로써 link 2로 전송한 후 두개 이상의 frame에 대한 Ack를 다 수신해야만 완전한 Ack로 간주하므로 virtual transmission time이 지연되어 발생하는 현상이다. 그리고 full-and-remainder packetizing보다 equal packetizing이 보다 나은 throughput을 산출하는 이유는 하나의 message를 분할할 때 등간격으로 나누는 것이 전체 frame-error probability를 감소시킬 수 있다는 것을 보여준다. 그래서 구현에 있어서 equal packetizing 방법이 유리하다는 결론이다.

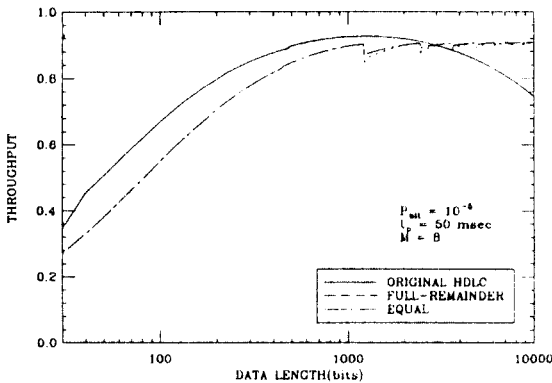


그림 5. 분할 방식에 따른 throughput의 비교( $v=4.8Kbits/s$ )  
Fig. 5. Comparison of throughput by division methods.

그림 6은  $v$ 를 24Kbits/sec로 놓고 서로의 throughput을 비교한 것인데 다른 모든 효과는 이미 설명한 바와 같고 다만 그림 5보다  $l_m$ 의 간격으로 더욱 현저한 throughput의 증감을 보이

는데 그것은 transmission rate가 클 때 short message에 대해 ack를 기다리는 시간이 상대적으로 길어짐으로써 throughput이 저하하는 특성과 관련이 있다. 즉 message가 분할될 때는 원래의 message보다 더욱 짧은 message가 두개이상 생성되기 때문이다.

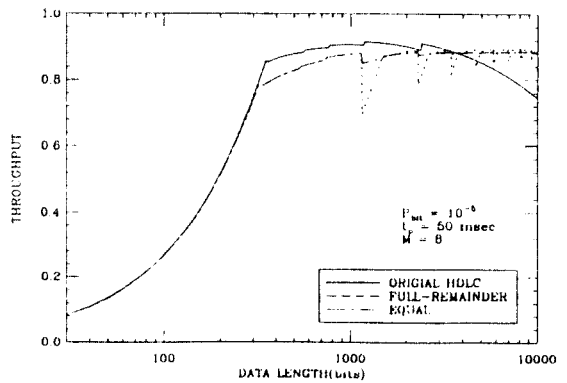


그림 6. 분할 방식에 따른 throughput의 비교( $v=24Kbits/s$ )  
Fig. 6. Comparison of throughput by division methods.

## 2. Mean Waiting Time

여기서는 channel의 full capacity 이하로 load를 걸어야 하므로 다음과 같이 useful channel load를 정의한다.

$$Y_u = \lambda l_s / v$$

$\lambda$ 는 converter로 유입되는 arrival rate를 의미한다. 그리고 SDLC host가 송출하는 transmission rate와 converter가 송출하는 transmission rate를 같다고 보는 것이 full duplex communication에서는 타당할 것이다.

그림 7은  $Y_u$ 를 일정하게 높음으로써 짧은 message에 대해서는 arrival rate가 높아서 queue에서 기다리는 시간이 길어지고, 긴 message에 대해서는 상대적으로 transmission time이 증가함과 동시에 frame error probability의 증가로 인한



error recovery 과정이 빈번히 발생하여 전체적인 waiting time이 증가함을 보여준다. 그리고 짧은 message에서 HDLC보다 converter의 waiting time이 길어지는 것은 conversion 과정에서 head의 부착으로 overhead가 증가했기 때문이고 긴 message에서 HDLC보다 성능이 향상되는 이유는 1절에서 설명한 것과 같은 효과이다. 또한 throughput curve에서 보여준 것과 같은 불연속 점도  $l_p$ 의 간격으로 관찰되는데 waiting time에서도 full and remainder packetizing보다 equal packetizing 방식이 우월하다는 것이 두드러지게 드러난다.

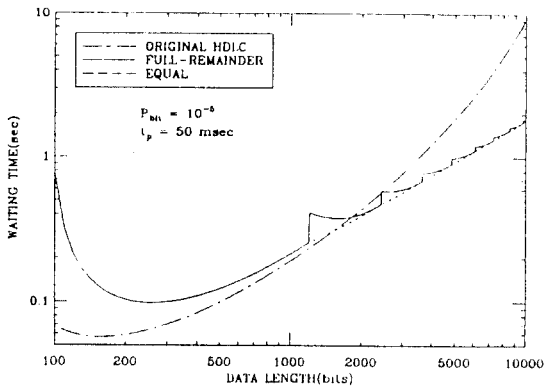


그림 7. 분할 방식에 따른 mean waiting time의 비교( $1.8Kbits/s$ )  
Fig. 7. Comparison of mean waiting time by division methods.

### V. 결 론

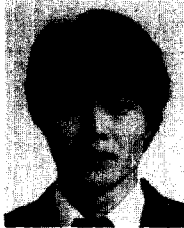
본 논문에서는 packet network와 protocol인 X.25와 SNA 2 layer protocol인 SDLC간을 접속하는 protocol converter의 성능분석을 하였다. Converter는 link level mapping method로 구현하였다고 가정하였다.

성능 분석을 통하여 각종 parameter에 대한 throughput과 mean queue length, 그리고 mean waiting time을 고찰할 수 있었다. 또한 packet을 분할하는 방식에서 full and remainder

packetizing보다 equal packetizing 방식이 성능면에서 우월하다는 것이 드러났다. 그 이유는 하나의 message를 분할 할때 동간격으로 나누는 것이 전체 frame error probability를 감소 시키는데 있어서 가장 유리하게 작용하기 때문이다.

### 참 고 문 헌

1. P.E. Green, "Protocol Conversion", *IEEE Trans. Commun.*, Vol. COM 34, No.3, pp.257-268, Mar. 1986.
2. K. Okumura, "A Formal Protocol Conversion Method", *Jour.*, 0-89791-201-2, pp.30-37, 1986.
3. P.Francois and A. Potocki, "Some Methods for Providing OSI Transport in SNA", *IBM J. Res. Dev.*, Vol.27, No.5, pp.452-463, Sep. 1983.
4. B. Lindgren, *Painless Packet Switching: a Protocol Primer*, Infotrans AB, Sweden, 1987.
5. B. Lindgren, *The X.25 Handbook*, Infotrans AB, Sweden, 1987.
6. CCITT Recommendation X.25, Red Book, Vol. VIII, 3, 1981.
7. "System Network Architecture", GC30-3073-2, IBM Cor., Sep. 1986.
8. "IBM Synchronous Data Link Control General Information", GA27-3093-1, IBM Corporation, May 1975.
9. "The X.25 Interface for Attatching SNA Nodes to PSDN's General Information Manual", GA27-3345-1, IBM Corporation, Feb. 1983.
10. F. P. Cori and D. H. Neal, "SNA and emerging international standards", *IBM Sys. Journal*, Vol. 18, No.2, pp.244-262, 1979.
11. W. Bux, K. Kummerle, and Hong Linh Truong, "Balanced HDLC Procedure: A Performance Analysis", *IEEE Trans. Commun.*, Vol. COM 28, No.11, pp.1889-1900, Nov. 1980.
12. M. Schwartz, *Telecommunication Networks: Protocols, Modeling and Analysis*, Addison Wesley Pub. Co, 1987.
13. D.Bertsekas and R. Gallager, *Data Network*, Prentice Hall, Inc., 1987.
14. L.Kleinrock, *Queueing Systems Vol. 1*, John Wiley & Sons Inc., 1975.



鄭 海(Hae CHUNG) 正會員  
1962年 1月 1日生  
1987年 2月: 漢陽大學校 電子通信工學科  
卒業  
1991年 2月: 韓國科學技術院 電氣 및  
電子工學科 碩士課程 卒業  
1991年 9月: 韓國科學技術院 電氣 및  
電子工學科 博士課程



殷 鍾 官(Chong Kwan UN) 正會員  
1940年 8月 25日生  
1964年 6月: 美國 University of Dela-  
ware 電子工學科卒業, 電子  
工學學士學位  
1966年 6月: 同大學院卒業, 電子工學碩  
士學位  
1969年 6月: 同大學院卒業, 電子工學博  
士學位  
1969年 9月 ~ 1973年 5月: 美國 University of Maine 電子工學  
助教授  
1973年 5月 ~ 1977年 6月: 美國 全理互換研究所(SRI) 責任研究  
員  
1977年 6月 ~ 現在: 韓國科學技術院 電氣 및 電子工學科 教  
授, 本學會 理事  
1983年 7月 ~ 1989年 6月: 韓國科學技術院 通信工學研究室長