

TDX-10 ISDN 교환기의 패킷교환호 흐름제어 기법연구

백 장현, 류 병한, 국 광호, 이 재섭

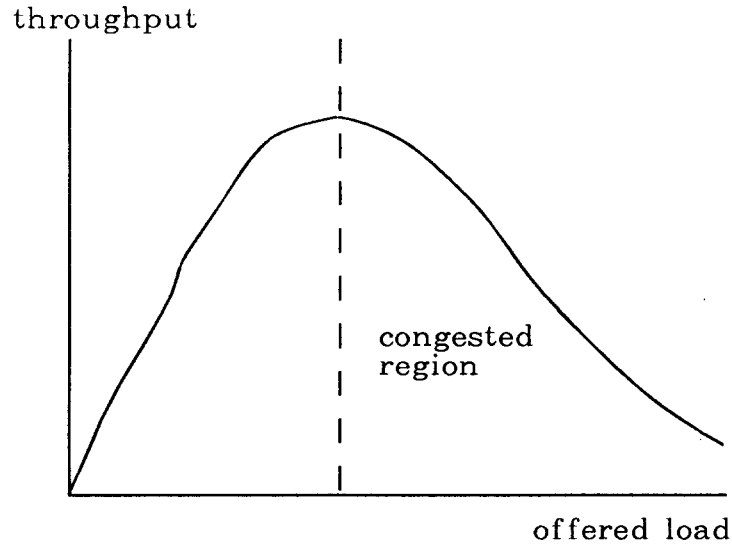
교환기술연구단
한국전자통신연구소
대전 대덕연구단지 사서함 8호
T.(042) 860-5313, FAX.(042) 861-1033

요 약

회선교환에서는 사용자 사이의 전송로가 통화가 지속되는 동안 계속 점유되는 것과는 달리 패킷교환에서는 망내의 자원을 여러 사용자들이 공유함으로써 자원을 효율적으로 사용하게 되는데 적절한 제어가 이루어지지 않으면 망내에 congestion이 발생하여 오히려 성능이 감소하게 된다. 본 연구에서는 기존의 흐름제어 기법들인 ack-at-end-of-window 기법과 pacing window flow control 기법을 토대로 새로운 흐름제어 기법을 제안한다. 새로운 흐름제어 기법의 성능을 시뮬레이션을 이용하여 TDX-10 패킷처리기에 적용하여 분석한 결과 기존의 방법보다는 새로운 흐름제어 기법의 성능이 다소 좋음을 보여준다.

I. 서 론

회선교환에서는 사용자 사이의 전송로가 통화가 지속되는 동안 계속 점유되는 반면 패킷교환에서는 여러 자원들(회선, 버퍼 등)을 사용자들이 공유하게 된다. 이러한 동적인 자원의 공유는 데이터 전송시 망의 자원을 효율적으로 사용할 수 있도록 해줌으로써 데이터의 처리능력을 향상시켜 준다. 그러나 패킷교환에서는 사용자가 전송하려는 데이터 양이 교환기의 처리능력을 초과하는 경우 적절한 제어가 이루어지지 않으면 congestion 현상이 발생한다. 통신망내에서 congestion 현상이 발생하여 데이터 전송 지연시간이 증가하면 송신측의 time-out에 의한 패킷 재전송에 의하여 트래픽의 양이 증가하게 되고 지연이 더욱 커지게 되는 상승작용에 의하여 <그림 1>과 같이 전체 성능이 감소하게 된다.



<그림 1> congestion에 의한 수율 감소 현상

Congestion 현상을 방지하기 위해서는 사용자의 데이터 전송속도를 조절할 수 있는 적절한 흐름제어 기법을 사용하는 것이 필요하다. 흐름제어의 주요 기능은 다음과 같다.

- . 과부하로 인한 성능 감소와 효율의 저하를 방지한다.
- . deadlock을 방지한다.
- . 사용자들에게 자원을 공평하게 할당한다.
- . 망과 사용자간의 속도를 조절한다.

본 논문에서는 X.25에서의 흐름제어 기법([1], [2] pp. 170-175)과 윈도우에 의한 흐름제어 기법([2] pp. 180-196), IBM SNA(System Network Architecture)에서 채택하고 있는 pacing window flow control 기법([2] pp. 197-209, [3], [4])에 대해서 살펴보고 이들을 토대로 새로운 흐름제어 기법을 제안한다. II절에서는 기존의 흐름제어 기법들에 대해서 간단히 살펴보고 III절에서는 본 논문에서 제안하는 흐름제어 기법에 대해서 기술한다. IV절에서는 기존의 흐름제어 기법과 새로 제안된 흐름제어 기법과의 성능을 TDX-10 패킷교환 구조에 적용하여 비교해 본다.

II. 흐름제어 기법

1. X.25에서의 흐름제어

X.25에서의 흐름제어는 각 가상회선에 대해 설정되는 윈도우(window)에 의해 수행된다. 윈도우 크기는 가상회선을 설정할 때 정해지는데 default 값은 2로 주어진다. 흐름제어는 DTE(Data Terminal Equipment)에서 DCE(Data Circuit-terminating Equipment) 방향, DCE에서 DTE 방향 모두 제공된다. 여기서 윈도우는 한방향(DTE에서 DCE 방향 또는

DCE에서 DTE 방향)의 가상회선상에 존재할 수 있는 최대 데이터 패킷수를 나타낸다. X.25에서는 데이터 패킷의 D 비트를 사용하여 DTE와 DCE 사이의 흐름제어를 수행하거나(D=0) 두 DTE 사이의 단-대-단(end-to-end) 흐름제어를 수행한다(D=1).

X.25에서 각 패킷은 순서 번호를 나타내는 P(S)와 P(R)을 포함하는데 P(S)는 데이터 패킷이 주어진 가상회선으로 전송될 때마다 1씩 증가한다. 데이터 패킷을 받은 수신측은 P(S)가 예상된 순서 번호와 같은지 비교하여, 같으면 그 패킷을 받은 후 P(R)을 송신측으로 전송하여 P(R)-1번까지의 패킷을 받았음을 알리고 다르면 REJ(REJECT) 패킷을 전송한다. 특정 가상회선에 대한 흐름제어는 수신측이 P(R)을 송신측으로 빨리 보내고 천천히 보냄에 의해 수행된다.

또한 3개의 제어 패킷, RR(Receive Ready), RNR(Receive Not Ready), REJ(REJECT)도 흐름제어를 목적으로 사용할 수 있다. 즉 P(R)에 의해서도 흐름을 조절할 수 있지만 더 빠른 방법은 RNR 패킷을 송신측에 보내 패킷 전송을 중단시키는 것이다. 이 경우 전송의 재개는 RR 패킷을 보냄으로써 알린다.

2. 윈도우에 의한 흐름제어

본절에서는 윈도우에 의한 흐름제어 기법중에서 두가지 극단적인 경우에 대해서 살펴본다. 첫번째 경우는 하나의 데이터 패킷을 수신할 때마다 데이터 패킷을 수신하였음을 송신측으로 알리는 방법(sliding window 기법)이고 두번째 경우는 윈도우 크기 만큼의 데이터 패킷을 수신한 다음에 데이터 패킷들을 수신하였음을 송신측으로 알리는 방법(ack-at-end-of-window 기법)이다. 앞에서 설명한 X.25에서의 흐름제어는 보통 위 두 극단적인 경우의 중간형태를 취한다.

윈도우에 의한 흐름제어 방법은 임의의 시점에 하나의 가상회선상에 윈도우의 크기(N) 이하의 패킷만이 전달될 수 있도록 제어함으로써 망내의 congestion이 발생하지 않도록 한다. 많이 사용되는 윈도우에 의한 흐름제어 방법중 sliding-window 기법과 ack-at-end-of-window 기법은 다음과 같다.

1) Sliding-window 기법

Sliding-window 기법에서 송신측은 데이터 전송시작 시점에 데이터 패킷을 전송할 수 있는 양을 나타내는 counter가 N(윈도우 크기)으로 주어진다. 하나의 패킷이 전송될 때마다 counter는 1씩 감소된다. 수신측에서는 하나의 패킷이 수신될 때마다 송신측으로 ack(acknowledgement) 패킷을 전송한다. 송신측에서는 ack 패킷을 수신할 때마다 counter를 1씩 증가시킨다. 송신측에서는 counter 값이 0보다 클 경우에만 데이터 패킷을 전송함으로써 흐름제어가 수행되게 된다. 이때 모든 패킷에 대해서 ack 패킷이 전송됨으로서 ack 패킷을 처리하기 위한 overhead가 증가하는 단점이 있다.

2) Ack-at-end-of-window 기법

Ack-at-end-of-window 기법에서도 송신측은 데이터 전송시작 시점에 데이터 패킷을 전송할 수 있는 양을 나타내는 counter가 N 으로 주어지며 하나의 패킷이 전송될 때마다 counter가 1씩 감소된다. 수신측에서는 N 개의 패킷이 모두 수신된 경우에 송신측으로 ack 패킷을 전송한다. 송신측에서는 ack 패킷이 수신될 때마다 counter 값을 N 만큼 증가시킨다. Ack-at-end-of-window 기법에서는 ack 패킷이 N 개의 패킷을 수신한 경우에만 전송되므로 ack 패킷을 처리하기 위한 overhead는 감소하지만 N 개의 패킷을 송신한 후 ack 패킷을 수신할 때까지는 송신측이 데이터 패킷을 전송할 수 없으므로 수율(throughput)이 감소하는 현상이 발생한다.

3. Pacing window flow control 기법

IBM SNA에서 채택하고 있는 pacing window flow control 기법도 윈도우에 의해 각 가상회선에 대한 단대단 흐름제어를 수행한다. Pacing window flow control 기법에서 송신측은 데이터 전송시작 시점에 데이터 패킷을 전송할 수 있는 양을 나타내는 PC(pacing counter)가 N 으로 주어진다. 하나의 패킷이 전송될 때마다 PC는 1씩 감소한다. 수신측은 매 N 개의 패킷중 첫번째 패킷이 수신될 때마다 ack 패킷을 송신측으로 전송한다(Ack-at-end-of-window 기법에서는 매 N 개의 패킷중 마지막 패킷이 수신될 때마다 ack 패킷을 송신측으로 전송함). 송신측은 ack 패킷을 수신할 때마다 PC 값을 N 만큼 증가시킨다. 따라서 임의의 시점에 PC의 값은 0에서 $2N-1$ 의 값을 가질 수 있다. Pacing window flow control 기법은 ack-at-end-of-window 기법과 같이 매 N 개의 패킷이 전송될 때마다 1개의 ack 패킷을 송신측으로 전송함으로써 ack 패킷을 처리하는데 소요되는 overhead가 적은 반면 임의의 시점에 데이터 패킷을 전송할 수 있는 양을 나타내는 PC의 값이 N 보다 크기 때문에 ack-at-end-of-window 기법보다 수율이 증가되게 된다.

III. 새로운 흐름제어 기법

Ack-at-end-of-window 기법에서는 수신측이 매 N 개의 데이터 패킷중 마지막 패킷을 수신한 경우에 ack 패킷을 송신측으로 전송하고 pacing window flow control 기법에서는 매 N 개의 데이터 패킷중 첫번째 패킷을 수신한 경우에 ack 패킷을 전송한다. 이때 pacing window flow control 기법에서는 임의의 시점에 하나의 가상회선 상에 존재할 수 있는 데이터 패킷의 수가 N 보다 크게 되므로($0 < PC < 2N-1$) ack-at-end-of-window 기법보다 좋은 성능을 갖게된다.

본 연구에서는 ack-at-end-of-window 기법과 pacing window flow control 기법의 절충 형태로서 임의의 시점에 가상회선상에 존재할 수 있는 데이터 패킷의 수는 최대 $2N$ 이 되며 매 N 개의 패킷중 마지막 패킷을 수신한 경우에 ack 패킷을 전송하는 방법을 제안한다

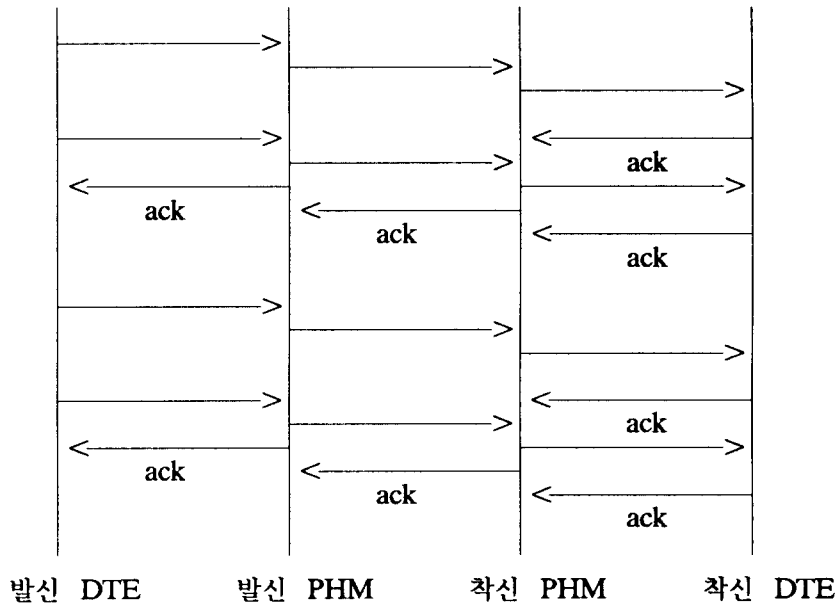
(시뮬레이션 수행결과 임의의 시점에 가상회선상에 존재할 수 있는 최대 데이터 패킷의 수가 윈도우 크기의 2배 정도가 되는 경우에 성능이 좋음). 즉, 송신측은 데이터 전송시작 시점에 데이터 패킷을 전송할 수 있는 양을 나타내는 counter가 $2N$ 으로 주어지며 데이터 패킷이 전송될 때마다 counter는 1씩 감소된다. 수신측에서는 N 개의 패킷이 모두 수신된 경우에 송신측으로 ack 패킷을 보내게 된다. 송신측에서는 ack 패킷이 수신될 때마다 counter 값을 N 만큼 증가시킨다. 이 흐름제어 기법은 매 N 개의 패킷이 전송될 때마다 1개의 ack 패킷이 송신측으로 전송됨으로서 ack 패킷을 처리하는데 소요되는 overhead가 적은 반면 임의의 시점에 데이터 패킷을 전송할 수 있는 양을 나타내는 counter의 값($2N$)이 ack-at-end-of-window 기법이나 pacing window flow control 기법보다 크므로 이들보다 수율이 좋으리라 예측된다.

IV. 흐름제어 기법 비교 분석

본 절에서는 앞서 제시한 방법중 ack-at-end-of-window 기법, pacing window flow control 기법, 그리고 새로운 흐름제어 기법을 TDX-10의 패킷교환 구조인 패킷처리에 적용함으로써 각 기법의 성능을 비교하고자 시뮬레이션 언어인 SLAM II를 이용하여 시뮬레이션을 수행하였다.

1. TDX-10 패킷처리기의 흐름제어 구조

TDX-10 패킷처리기의 흐름제어 구조를 간단히 그림으로 나타내면 <그림 2>와 같다.



<그림 2> TDX-10 흐름제어 구조

<그림 2>는 발신 DTE와 발신 PHM, 발신 PHM과 착신 PHM은 ack-at-end-of-window 기법으로, 착신 PHM과 착신 DTE는 sliding window 기법으로 흐름제어를 수행하는 경우를 나타낸다. 즉, <그림 2>에서 발신 DTE는 발신 PHM에서의 ack 패킷이 없으면 N개까지만 데이터 패킷을 보낼 수 있다. 발신 PHM에서는 발신 DTE에서의 데이터 패킷이 N개 수신될 때마다 하나의 ack 패킷을 발신 DTE로 보낸다. 이때 발신 PHM은 가상회선당 버퍼가 데이터 패킷 B개까지로 한정되어 있으며 버퍼에 여유가 있어야 발신 DTE로 ack 패킷을 보내게 된다. 해당 가상회선의 버퍼가 차게 되면, 버퍼에 여유가 생길 때까지 발신 DTE로 ack 패킷을 보낼 수 없다. 이 상태에서 발신 PHM이 착신 PHM으로부터 ack 패킷을 받으면, 이는 현재 버퍼에 있는 데이터 패킷중 가장 먼저 송신한 N개의 데이터 패킷이 전달되었음을 의미하므로 해당되는 N개의 데이터 패킷을 버퍼에서 제거한다. 그러면 버퍼에 여유가 생기므로 발신 DTE로 ack 패킷을 보낼 수 있다.

발신 PHM과 착신 PHM간의 흐름제어도 윈도우 크기 N인 ack-at-end-of-window 방법을 따른다. 즉 착신 PHM은 발신 PHM으로부터 매 N개의 데이터 패킷이 수신될 때마다 하나의 ack 패킷을 발신 PHM으로 전달한다. 착신 PHM도 가상회선당 버퍼가 B개까지로 한정되어 있으며, 버퍼에 데이터 패킷 N개 이상의 여유가 있어야 발신 PHM으로 ack 패킷을 보내게 된다.

착신 PHM과 착신 DTE간의 흐름제어는 윈도우 크기 N인 슬라이딩 윈도우 방법을 따른다고 가정하였으나 DTE에 따라서 다양한 흐름제어 방식을 가질 수 있다.

2. 시뮬레이션을 위한 가정사항

시뮬레이션 수행을 위한 가정사항은 다음과 같다.

- o 패킷호의 발생간격은 지수분포를 따른다.
- o 하나의 패킷호에 대하여 PHM을 거치는 패킷을 다음과 같이 가정한다(B채널 패킷호 시나리오를 근거로 하여 산출하였음).

| 종 류 | 패킷크기 (byte) | 패킷 수 | |
|---------|----------------|--------|--------|
| | | 발신 PHM | 착신 PHM |
| 흐제어 패킷 | 50 | 25 | 24 |
| Data 패킷 | 145 | 30 | 30 |
| ack 패킷 | 30 | 15 | 15 |

- o 패킷호 설정후 30개의 데이터 패킷이 한꺼번에(bursty) 발생하며 데이터 패킷의 발생간격은 145 byte 크기의 데이터 패킷이 64kbps 링크로 전달된다고 가정하여 18.125 msec로 한다.
- o 분석에 필요한 파라미터 값들을 다음과 같이 가정한다.
 윈도우의 크기 $N = 2$
 PHM 버퍼의 크기 $B = 8$

DTE에서의 데이터/ack 처리시간 = 1.0 msec
 PHM에서의 데이터/ack 처리시간 = 2.211 msec
 PHM에서의 흐름제어 패킷 처리시간 = 0.444653 msec
 DTE-PHM간 데이터 전송시간 = 18.125 msec
 DTE-PHM간 ack 전송시간 = 3.0 msec

3. 성능척도

흐름제어 기법의 비교에 있어서 성능척도는 수율(throughput)과 지연시간(delay time)을 들 수 있다. 일반적으로는 동일한 수율하에서 지연시간을 비교하거나 동일한 지연시간하에서 수율을 비교함으로써 흐름제어 기법의 좋고 나쁨을 판정하게 된다. TDX-10에서 권고하고 있는 데이터 패킷 전송지연의 내용은 다음과 같다.

데이터 패킷 전송지연(data packet transfer delay)

데이터 패킷 전송지연은 "교환기가 발신측 DTE로부터 데이터 패킷을 수신한 순간부터 착신측 DTE로 그 데이터 패킷을 송신하기까지의 지연시간"을 의미하며 권고값은 <표 1>과 같다.

<표 1> 데이터 패킷 전송지연시간 권고값

| 지연시간 항목 | 기준부하 A | | 기준부하 B | |
|-------------------|--------|-----|--------|-----|
| | 평균 | 95% | 평균 | 95% |
| 데이터 패킷 전송지연(msec) | 200 | 400 | 350 | 700 |

4. 흐름제어 기법 비교 결과

흐름제어 기법의 성능을 비교하기 위해 TDX-10 패킷처리기의 발신 PHM과 착신 PHM 사이는 ack-at-end-of-window 기법을 적용하고 착신 PHM과 착신 DTE 사이는 sliding window 기법을 적용한 상태에서, 발신 DTE와 발신 PHM 사이의 흐름제어 기법으로서 ack-at-end-of-window 기법, pacing window flow control 기법, 새로운 흐름제어 기법을 적용한 경우의 시뮬레이션 수행 결과는 다음과 같다.

1) Ack-at-end-of-window 기법 적용시

발신 DTE와 발신 PHM 사이에 ack-at-end-of-window 기법을 적용한 경우에 PHM에 부가되는 트래픽에 따른 데이터 패킷 전송지연시간은 <표 2>와 같다.

<표 2> Ack-at-end-of-window 기법 적용시의 데이터 패킷 전송지연시간

| PHM당 트래픽 (데이터/초) | PHM 점유율(%) | 평균값 (msec) | 400msec 초과율(%) |
|------------------------|---------------|---------------|-------------------|
| 44.4 | 39 | 88.4 | 1.0 |
| 50.0 | 42 | 94.3 | 2.6 |
| 60.0 | 49 | 117 | 4.8 |
| 66.7 | 56 | 131 | 5.6 |
| 72.7 | 59 | 155 | 11.8 |
| 80.0 | 65 | 181 | 13.7 |
| 85.0 | 70 | 223 | 14.5 |
| 100.0 | 86 | 517 | 36.5 |

<표 2>로부터 PHM의 데이터 패킷 처리능력은 400 msec 초과율이 5% 이하가 되는 약 60 데이터 패킷/초 정도가 됨을 알 수 있다.

2) Pacing window flow control 기법 적용시

발신 DTE와 발신 PHM사이에 pacing window flow control 기법을 적용한 경우에 PHM에 부가되는 트래픽에 따른 데이터 패킷 전송 지연시간은 <표 3>과 같다.

<표 3> Pacing window flow control 기법 적용시의 데이터 패킷 전송지연시간

| PHM당 트래픽 (데이터/초) | SNA PC | | |
|------------------------|---------------|---------------|-------------------|
| | PHM 점유율(%) | 평균값 (msec) | 400msec 초과율(%) |
| 44.4 | 39 | 78.3 | 0.8 |
| 50.0 | 42 | 84.3 | 2.0 |
| 60.0 | 49 | 104 | 3.2 |
| 66.7 | 56 | 117 | 3.6 |
| 72.7 | 59 | 142 | 10.5 |
| 80.0 | 65 | 166 | 11.9 |
| 85.0 | 70 | 177 | 13.6 |
| 100.0 | 85 | 357 | 34.8 |

<표 3>으로부터 PHM의 데이터 패킷 처리능력은 400 msec 초과율이 5% 이하가 되는 약 67 데이터 패킷/초 정도가 됨을 알 수 있다.

3) 새로운 흐름제어 기법 적용시

발신 DTE와 발신 PHM사이에 새로운 흐름제어 기법을 적용한 경우에 PHM에 부가되는 트래픽에 따른 데이터 패킷 전송 지연시간은 <표 4>와 같다.

<표 4> 새로운 흐름제어 기법 적용시의 데이터 패킷 전송 지연시간

| PHM당 트래픽 (데이터/초) | PHM 점유율(%) | 평균값 (msec) | 400msec 초과율(%) |
|------------------------|---------------|---------------|-------------------|
| 44.4 | 39 | 71.7 | 0.4 |
| 50.0 | 42 | 78.0 | 2.0 |
| 60.0 | 49 | 93.7 | 2.7 |
| 66.7 | 56 | 106 | 2.9 |
| 72.7 | 59 | 131 | 9.1 |
| 80.0 | 65 | 156 | 9.6 |
| 85.0 | 70 | 166 | 12.2 |
| 100.0 | 85 | 352 | 34.0 |

<표 4>로부터 PHM의 데이터 패킷 처리능력은 400 msec 초과율이 5% 이하가 되는 약 67 데이터 패킷/초 정도가 됨을 알 수 있다.

위 결과로부터 pacing window flow control 기법이나 새로운 흐름제어 기법을 적용할 때 TDX-10 PHM의 데이터 패킷 처리능력이 약 10% 정도 향상됨을 알 수 있다(발신 PHM과 착신 PHM, 착신 PHM과 착신 DTE사이에도 pacing window flow control 기법이나 새로운 흐름제어 기법을 사용할 경우 TDX-10 PHM의 데이터 패킷 처리능력은 보다 향상되리라 예측됨). 이때 pacing window flow control 기법과 새로운 흐름제어 기법 적용시 데이터 패킷 처리능력에는 차이가 없는 것으로 보이나 동일한 트래픽이 PHM에 가해졌을 경우 새로운 흐름제어 기법의 지연시간이 pacing window flow control 기법의 지연시간보다 적음을 알 수 있다(예, PHM에 60 데이터 패킷/초의 트래픽이 가해질 경우, pacing window flow control 기법의 데이터 패킷 전송 지연시간은 평균 104 msec 정도이나 새로운 흐름제어 기법 적용시 약 93.7 msec 정도임). 이는 새로운 흐름제어 기법에서 하나의 가상회선 상에 존재할 수 있는 최대 데이터 패킷의 수가 pacing window flow control 기법보다 1만큼 크기 때문이다.

V. 결 론

패킷교환에서는 망의 자원을 여러 사용자들이 공유함으로써 자원을 효율적으로 사용하게 되는데 적절한 제어가 이루어지지 않으면 망내에 congestion이 발생하여 오히려 성능이 감소하게 된다. 본 연구에서는 기존의 흐름제어 기법들을 고찰하였고 ack-at-end-of-window 기법과 pacing window flow control 기법을 토대로 새로운 흐름제어 기법을 제안하였다. 이들 기법들의 성능을 시뮬레이션에 의해 TDX-10 패킷처리기에 적용하여 분석한 결과 ack-at-end-of-window 기법보다는 pacing window flow control 기법이나 새로운 흐름제어 기법이 더 좋은 성능을 가짐을 보였다. 또한 새로운 흐름제어 기법이 pacing window flow control 기법보다 동일한 수율하에서 더 적은 지연시간을 가짐을 보였다.

[참고문헌]

- [1] CCITT Recommendations X.25, Blue Book, 1988.
- [2] M. Schwartz, "Telecommunication Networks : Protocols, Modeling and Analysis," Addison-Wesley publishing Company, 1987.
- [3] J. B. Suk and C. G. Cassandras, "Analysis and Optimization of Pacing Window Flow Control with Admission Delay," *IEEE INFOCOM '88*, 4C.1.1.-4C.1.8., 1988.
- [4] M. Schwartz, "Performance Analysis of the SNA Virtual Route Pacing Control," *IEEE Trans. Comm.*, Vol. COM-30, 172-184, Jan, 1982.
- [5] A. Alan and B. Pritsker, "Introduction to Simulation and SLAM II," System Publishing Corporation, 1986.
- [6] 전 경표, 류 병한, "TDX-10 Packet Handler의 성능분석," *대한산업공학회 한국경영과학회 '92 춘계공동학술대회 발표논문 및 초록집*, 466-475, 1992.