

# VBR 서비스상의 TCP 트래픽 성능 향상에 관한 연구

박봉주\*, 김명희\*, 박승섭\*

\*부경대학교 전자계산학과

e-mail:bjpark@woongbi.pknu.ac.kr

## A Study on Performance Improvement for TCP traffic over VBR service

Bong-Ju Park\*, Myong-Hee Kim\*, Seung-Seob Park\*

\*Dept of Computer Science, PuKyong University

### 요약

초고속통신의 실현과 다양한 멀티미디어 서비스를 위한 해결책으로서 ATM이 널리 사용되고 있다. TCP가 ATM망의 ABR/UBR 서비스를 사용할 때, ABR/UBR 제어방식과 TCP 흐름제어 사이의 서로 다른 동작 때문에 TCP 파라미터 값이 TCP층의 성능뿐만 아니라 ATM 층에도 영향을 미쳐 처리율과 공정성의 성능저하를 가져온다.

본 논문에서는 효과적인 처리율과 공정성의 관점에서 실제 통신망을 고려하여 VBR서비스 상에서의 TCP 트래픽의 성능향상을 위해 효과적인 파라미터 제어운용 방안을 제시하고, 시뮬레이션 결과를 통해 개선된 높은 처리율과 공정성을 나타내었다.

### 1. 서론

B-ISDN은 다양한 서비스를 하나의 망을 통해 제공하는 특징을 가진다. 기존의 통신망에서 제공하던 전화, 데이터 전송 서비스뿐만 아니라 고음질의 음향, HDTV, 동화상, 온라인 화상 회의등 높은 수준의 서비스 품질을 요구하는 서비스를 제공한다.

ATM(Asynchronous Transfer Mod)은 B-ISDN을 실현하기 위한 기본 방식으로 채택되어 탄생한 것이다. ATM Forum TM(Traffic Management) 4.0은 높은 처리율, 효율적인 자원활용, 사용자의 다양한 요구 만족을 위해 트래픽 특성과 서비스 품질에 따라 5개의 서비스 부류를 정의한다[1]. 실시간 서비스는 CBR(Constant Bit Rate), rt-VBR(real-time Variable Bit Rate)로 구성되며, 비실시간 서비스는 nrt-VBR(ont real-tim Variable Bit Rate), ABR(Availiable Bit Rate), UBR(Unspecified Bit Rate)로 구성된다.

ABR 서비스는 CBR/VBR의 실시간 서비스를 위한 대

역폭을 제외한 나머지 대역폭을 ABR 접속에 할당함으로써 전송 링크의 전체 대역폭 활용도를 높이는 것을 목적으로 한다. ABR 서비스는 긴 셀 지연이 문제되지 않으며 망의 상태에 따라 신속하게 적응할 수 있는 비실시간 데이터 전송에 가장 적합하다. 혼잡제어는 ABR 서비스에서 매우 중요한 부분을 차지한다. 또한 혼잡제어는 전송 링크를 공유하는 ABR 접속간에 공정하게 대역폭을 할당한다.

ATM포럼은 전송률 기반 방식을 혼잡제어를 위한 기본 방식으로 채택하였다. 이것은 구현 비용이 적고, 간단한 스위치 동작을 필요로 하며, 많은 유연성을 가지기 때문이다. 전송률 기반 방식은 공급원에서 생성된 RM(Resource Management)셀이 ATM 망과 도착지를 거쳐 다시 공급원으로 되돌아 오면서 망의 상태를 공급원에 전달하는 방식이다. 스위치에 RM셀이 도달하면 전송률 할당 알고리즘에 따라 접속간에 공정하게 대역폭을 할당하며, 공급원은 전달된 RM셀에 기반하여 망에 혼잡이 일어나지 않도록 폭주제어를 하고 제어 변수들을 적절하게 설정함으로써 셀

손실 없는 데이터 전송을 가능하도록 송신측의 전송 속도를 조절한다[1].

UBR 서비스는 상위 어떤 유용한 대역폭을 사용하고 셀 지연에 민감하지 않는 트래픽을 전송하는데 유리하며, 특히 ATM 망에서 제공하는 UBR 서비스는 명백한 혼잡 제어 메커니즘이 없다. 이러한 단점을 보완한 셀 폐기 정책 중 하나인 EPD(Early Packet Discard)를 이용해 망에서 불필요한 대역폭 낭비를 줄일 수 있다.

TCP는 현재 데이터를 전송하기 위한 전송 프로토콜로 가장 많이 사용된다. 기본적으로 데이터를 신뢰성 있게 전송하기 위하여 윈도우를 이용하며 Slow Start, 폭주회피, 그리고 재전송 단계로 구성된 폭주 제어와 오류제어를 수행한다[2]. TCP의 윈도우 기반 제어와 ABR 서비스와 UBR 서비스는 전송률 기반 제어가 독립적으로 동작한다.

TCP는 하위계층의 망 정보를 이용하지 않고 응답 메시지를 이용하여 망의 상태를 예측하기 때문에 갑작스런 트래픽 양의 증가는 망에서의 지연 변이를 급격히 증가시키고, TCP에서 잘못된 재전송을 유발할 가능성이 커진다. 이로 인해 대역폭의 낭비와 처리율의 저하를 가져온다. 이러한 문제는 ATM 하부구조하에서 효과적인 TCP 동작의 연구가 진행 중이다[3][4].

본 논문에서는 효과적인 처리율과 공정성의 관점에서 실제 통신망을 고려하여 VBR서비스 상에서의 TCP 트래픽의 성능향상을 위해 주요 파라미터인 timer granularity와 MTU의 제어 운용방안을 제시하고, ERICA, UBR, UBR+EPD, EFCI 각 알고리즘에 대해 시뮬레이션을 통한 성능 평가를 하였다. 본 논문의 구성은 1장 서론에 이어, 2장에서 ATM 계층에서의 혼잡제어 방식, 3장에서는 TCP over ABR/UBR 서비스를 기술하고, 4장에서는 시뮬레이션 환경 및 결과를 서술하였고, 5장에서는 개선방안을, 마지막으로 6장에서는 논문의 결론을 서술하였다.

## 2. ATM 계층에서의 혼잡 제어 방식

### 2.1 UBR과 UBR+EPD상에서의 동작

UBR 서비스는 CBR, VBR, ABR 서비스가 사용하고 남은 대역폭을 사용하는 지연에 민감하지 않는 트래픽을 전송하는데 유리한 서비스이다. 상위 계층인 TCP 제어방식에 의존한다. ATM 스위치의 버퍼에서

하나의 셀 손실은 TCP 층의 하나의 패킷 손실로 이어진다. 이는 망에서 낮은 처리율과 공정성으로 이어져 확실한 QoS의 보장이 어렵다. 이러한 단점을 보완한 셀 폐기 정책인 EPD(Early Packet Discard)나 PPD(Partial Packet Discard), RED(Random Early Detection)[5] 등을 사용해 망에 불필요한 대역폭 낭비를 줄일 수 있다. EPD는 버퍼에 임계치를 두어 이 임계치를 넘는 패킷의 첫 번째 셀이 있으면 그 첫 번째 셀과 같은 가상채널 식별자를 가지는 패킷



(그림1) EPD 동작

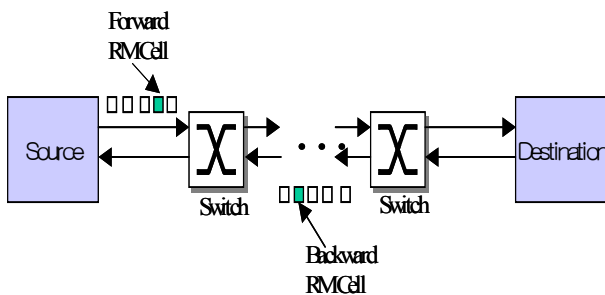
전부를 폐기하는 방법이다. 버퍼 오버플로우로 야기되는 패킷의 손실을 방지하여 불필요한 재전송과 대역폭 낭비를 막는다. (그림1)에서는 Packet1이 폐기된다.

### 2.2 ABR 상에서의 동작

ABR 트래픽에서는 망의 상태에 따라 송신측의 전송률을 조정할 수 있고, 이용 가능한 네트워크 자원을 공평하고 효율적으로 사용할 수 있다. 또한, 매우 버스티한 데이터 어플리케이션의 지원하기 위해 제안되었다. RM 셀은 제어정보에 대한 자세한 사항을 기술하여 소스로 전달하고 소스는 이 정보 셀을 이용하여 현재의 네트워크에서 적합한 셀 전송율을 조정하게 된다. 즉, 소스는 수신응답이 타임아웃 기간 동안 도달되지 않은 경우와 같은 암시적인 방법이나 RM 셀에 의한 명시적인 방법으로 네트워크의 폭주상태에 대한 정보를 얻는다. ATM 네트워크 프로토콜은 연결위주 접근법을 기초로 해 설계되었다.

소스가 셀을 전송하기 전에 사용자는 통신량의 특성과 서비스의 요구품질을 명시하고 연결이 설정되었을 때 네트워크는 피드백 정보를 활용하여 셀손실을 최소로 하면서도 해당 소스에 대한 원하는 서비스 품질을 보장하도록 한다.

ABR 트래픽 관리 모델을 전송률 기반 종단간 폐쇄 루프라 부른다. (그림2)와 같이 전송률 기반이라 함은 TCP처럼 크레딧 기반이 아니라 송신측이 정해진 전송률로 데이터를 보낸다는 의미이다. 또한 네트워크와 송신측간의 끊임없는 제어 정보 피드백으로 전송률을 관리하며, 송신측으로부터 수신측으로 제어 정보를 보내고 다시 수신측으로부터 송신측으로 보내는 종단간 관리기법이다. 지금까지 제안된 대표적인 전송률 기반의 ABR 스위치로는 크게 EFCI(Explicit Forward Congestion Indication)스위치와 ER(Explicit Rate) 스위치로 나눌 수 있다.



(그림2) RM 셀의 경로

ER 스위치는 VC간의 링크 전송률의 공평한 몫인 공정성을 계산하여 역방향 RM 셀의 ER 필드에 실어 송신원에 알리게 된다. 이렇게 함으로써 송신측은 경로상의 모든 스위치의 허가된 MCR(Minimum Cell Rate) 값을 받게 되고 병목현상을 고려할 수 있게 된다. 이러한 ER 스위치의 구조는 EPRCA(Enhanced Proportional Rate Congestion Avoidance), ERICA(Explicit Rate Indication for congestion Avoidance), ERICA+, CAPC등의 구조가 있다[8]. 이 중에서 ERICA 알고리즘은 *load factor(z)*를 사용하여 VCshare와 fair share를 계산한다. 그 정의는 다음과 같다.

$$z = \frac{\text{InputRate}}{\text{TargetRate}}$$

스위치의 목표 용량은 전체 대역폭의 90~95%정도로 높게 설정한다. 스위치가 계산하는 faire share는 다음 식과 같다.

$$\text{Fairshare} = \frac{\text{Targetcapacity}}{\text{NumberofactiveVCs}}$$

송신원이 사용할 수 있는 여분의 용량은 다음 식으로 표현한다.

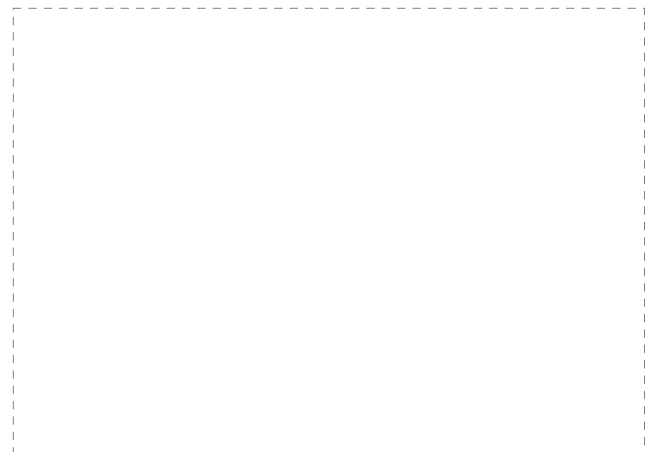
$$\text{VCshare} = \frac{\text{CCR}}{z}$$

이상과 같이, ABR 서비스의 트래픽 제어를 위한 표준으로 ATM 포럼 TM 4.0에서 정의한 EFCI 방식과 ERICA 방식을 사용하여 성능 평가를 하였다.

### 3. TCP트래픽에서의 ABR 서비스

#### 3.1 TCP트래픽에서 ABR

TCP는 연결지향형 프로토콜로서 현재 데이터를 전송하기 위한 전송 프로토콜로 가장 많이 사용되고 있으며 데이터를 신뢰성있게 전송하기 위하여 윈도우를 이용하여 폭주 제어와 오류제어를 수행한다. 또 TCP는 점대점 통신만 가능하며 전이중 데이터 전송 기능을 제공한다. 오류제어와 흐름제어를 위해 go-back-N ARQ(automatic repeat request)와 sliding-window 방식을 이용한다.



(그림3) 서비스 처리 범주와 프로토콜 계층

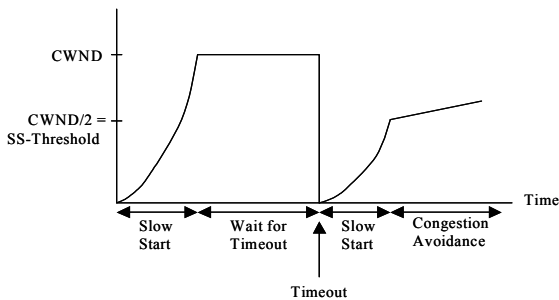
TCP가 ABR서비스 상에서 동작하게 될 경우 TCP의 윈도우 기반 흐름제어는 ABR 서비스의 전송률 기반폭주 제어 위에서 동작하게 되는데 이는 성능의 저하를 가져올 수 있다[8]. 이러한 이유는 (그림 3)과 같이 TCP의 윈도우 기반 제어와 ABR 서비스의 전송률 기반 제어가 독립적으로 동작하기 때문이다. TCP는 하위계층의 망 정보를 이용하지 않고 응답 메시지를 이용하여 망의 상태를 예측하기 때문에 갑작스런 트래픽 양의 증가는 망에서의 지연 변이를 급격히 증가시켜 TCP에서 잘못된 재전송을 일으킬 가능성이 있다. 이런 잘못된 재전송은 정상적으로 전송된 셀을 다시 전송 하게 되므로 대역폭의 낭비와 처리율의 저하를 초래한다. TCP와 ABR의 상호 작용을 방지하기 위해서 coarse timer granularity 와

큰 윈도우 사이즈가 요구된다. 더 나아가 ABR 기반의 TCP 성능의 개선을 위해서는 ACR과 같은 ECN이 잘 정의되어 ATM-API를 통해 TCP 흐름제어 메커니즘이 원활하게 운용되어야 한다[10].

TCP 프로토콜은 슬라이딩 윈도우 흐름제어 기법과 재전송 기법을 사용하여 TCP 사용자 사이에서 에러 없이 종단간 데이터를 보증한다. 윈도우 크기는 버퍼 오버플로우로 인한 대역폭 낭비를 제한하는 몇몇 TCP 세션들 사이에서 대역폭을 나누도록 보증하기 때문에 적응적이다. TCP 윈도우 조절 알고리즘은 거대한 연구의 주제가 되고 있고 TCP의 성능을 보증하기 위해 계속해서 발전하고 있다. 일반적으로 송신측은 세그먼트를 송신한 후에 수신측으로부터 ACK 없이 연속적인 세그먼트 송신이 가능하다. 그러나 수신측 버퍼용량이 부족할 때는 송신측에서의 세그먼트 흐름을 제어하는 흐름제어가 필요하다. 가장 많이 쓰이는 방법은 sliding-window 제어이다. TCP에서는 타이머가 만료될 때까지 ACK (positive acknowledgement)가 수신되지 않으면 전송한 세그먼트를 다시 전송하는 재전송방법을 사용하여 TCP 송신측은 최대로 윈도우 크기만큼의 세그먼트를 ACK 없이 송신할 수 있다. 송신된 세그먼트에는 데이터 세그먼트의 순서번호를 가지고 있다고 가정한다. ACK를 수신한 송신측은 할당받은 크레딧에 따라서 ACK 없이 송신할 수 있는 자신의 윈도우를 증가시킨다[7].

### 3.2 TCP 혼잡제어

TCP의 혼잡제어로는 slow start, congestion avoidance, re transmission, fast recovery 등의 4가지 알고리즘을 사용한다 [6].



(그림 4) Slow start와 폭주회피 단계의 폭주윈도우의 크기 변화

(그림4)와 같이 Slow start는 송신원에서 수신원에 연결 요청 수락 후, 초기 데이터를 전송하고 수신측 TCP 컨넥션은 송신원에 ACK를 보내서 수신에 성공했음을 알린다. 이때 송신원은 RTT(Round Trip Time)를 계산하고, RTT를 기본으로 RTO (Retransmission Timeout)를 계산한다. 타이머가 만료될 때까지 ACK 신호를 받지 않으면 세그먼트 손실로 간주하고 전송한 세그먼트부터 다시 전송하는 재전

송 방법을 사용한다. 또한 에러가 발생되지 않을 경우에는 최대 윈도우 크기만큼의 세그먼트를 ACK 없이 송신할 수 있다. Slow start 동안에는 매 ACK에 대해서 폭주 윈도우(congestion window: cwnd)의 크기를 하나씩 증가시키며, 폭주 윈도우의 크기 증가는 ss-threshold (slow start threshold)에 이르거나 세그먼트의 손실이 감지될 때까지 계속된다. 이러한 방식으로 망에서 윈도우의 크기를 동적으로 조정하여 혼잡을 회피하는데, 혼잡 시나 전송한 세그먼트에 대한 타이머가 만료되면 현재 윈도우 크기의 반(cwnd/2)을 ss-threshold로 설정하고 폭주 윈도우(cwnd)의 크기를 1로 설정하고 slow start 과정을 cwnd가 ss-threshold이거나 세그먼트의 손실이 감지될 때까지 계속 증가하게 된다. 또한 폭주윈도우의 크기가 ss-threshold와 같아지거나 커지면 혼잡회피 단계로 들어가는데, 현재의 폭주 윈도우의 크기를  $a$ 로 가정하면 매 ACK마다  $1/a$ 만큼의 폭주 윈도우가 증가하게 된다. 이는 폭주윈도우크기에 도달하거나 세그먼트의 손실이 감지될 때까지 진행시킨다. 빠른 재전송은 수신원에 순차번호에 맞지 않는 세그먼트가 수신되면 잘못 수신된 세그먼트 순차번호를 ACK에 실어 송신하여 올바른 순차번호를 갖는 세그먼트가 수신될 때까지 계속 보낸다. 그러면 송신원은 중복된 ACK가  $n$ 개 이상 수신되면 손실로 간주하여 재전송 한다. 이것은 재전송 타이머가 만료될 때까지 기다려야 하는 시간을 줄이는 방법이다.

### 4. 시뮬레이션 환경

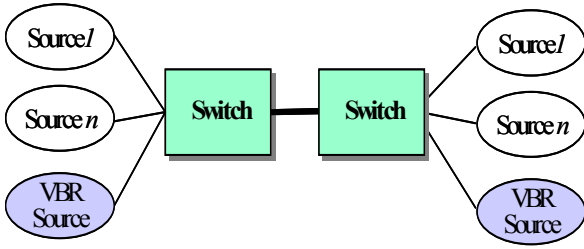
본 장은 시뮬레이션 망 모델과 파라미터에 대해 설명하고 결과를 분석한다. C++로 작성된 YATS ATM 시뮬레이션 도구[9]를 사용하여 시뮬레이션 하였다.

#### 4.1 시뮬레이션 모델

UBR과 ABR 서비스 상에서 TCP의 성능 분석을 위해 (그림 5)와 같은 하나의 병목 구간을 가지는 피어 투 피어 모델을 설정하였다. 전송방향은 단방향 트래픽으로만 데이터를 전송하게 한 모델이다. 링크 속도는 150Mbps이고, 스위치간 거리는 56Km, TCP 컨넥션 수는 55개로 제한한다.

스위치 방식으로는 EFCI, ERICA를 적용해서 ABR 상에서의 TCP 성능을 분석하였고, UBR 상에서의 TCP 성능 분석을 위해서는 UBR 그 자체와 UBR+EPD 알고리즘을 적용하였다.

그리고 송신원의 TCP 애플리케이션은 고정 프레임 길이를 가지면서 지속적으로 프레임을 전송하고, Bursty 트래픽을 위해 ON-구간과 OFF-구간을 가지는 VBR 소스를 삽입 시켰다.



(그림 5) 망 구성도

## 5.2 시뮬레이션 파라미터 설정 및 결과

각 ABR, UBR, VBR, TCP 컨넥션, 스위치, Link의 기본 파라미터 설정은 아래<표1>과 같이 나타내었다. 정의된 파라미터 값으로써 성능 개선에 대해서 다를 때는 timer granularity와 MTU크기를 변경하였다. 그리고 ATM 계층에서는 TCP 최대 세그먼트 크기(MSS)를 2000 바이트에 TCP 헤더 20 바이트와 IP 헤더 20 바이트, LLC헤더 8 바이트, AAL트레이일러 8 바이트에 패딩과 ATM 층의 헤더 5 바이트가 합쳐져서 총 43 cells이 된다. 그래서 최대 이용 가능한 처리율은  $2000 / 2279 = 0.87$ 로 130.5Mbps가 된다.

성능 평가 요소로서는 timer granularity와 MTU에 따른 처리율과 공정성을 기준으로 하였다. timer granularity와 윈도우 사이즈 ABR상의 TCP에서 상호작용을 피하기 위해서 필요한 파라미터들이다[10]. 공정성은 서로 다른 VC 간 대역폭 할당에 있어서 공평하게 할당받았는가를 평가하는 기준으로 다음 식과 같이 주어진다.

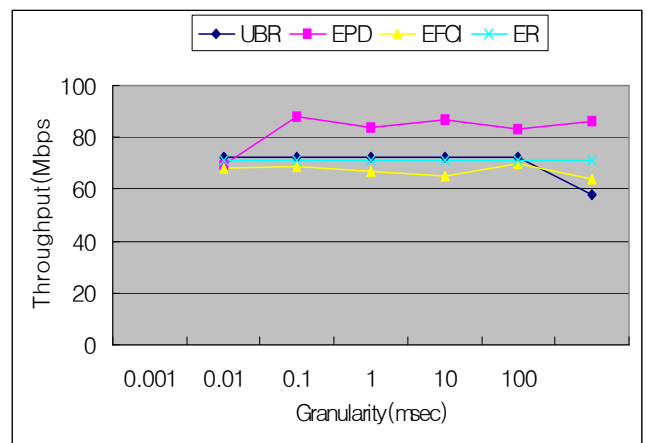
$$Fairness = \frac{(\sum x_i)^2}{n \times (\sum x_i^2)}$$

여기서,  $x_i = T_i / O_i$ 로  $Y_i$  ( $i=1 \dots n$ )는 각 VC마다 성공적으로 도착된 패킷의 측정된 처리 값으로 단위 시간 당 전송 받은 양을 바이트로 계산하였고,  $O_i$  ( $i=1 \dots n$ )는 최대-최소의 최적 처리율이고,  $n$ 은 전체 VC의 수이다. 여기에서 공정성의 기준값으로 1을 설정하였는데, 공정성의 의미는 각 TCP 컨넥션 간의 대역폭을 공평하게 할당받았는가를 평가한다. 실험 결과의 (그림6)과 (그림7)에서 VBR 서비스상의 TCP 트래픽의 성능을 timer granularity 변화에 따른 처리율과 공정성을 보여 주고 있다. UBR+EPD에서 TCP가 가장 높은 처리율을 보여주고

<표 1> 시뮬레이션 파라미터

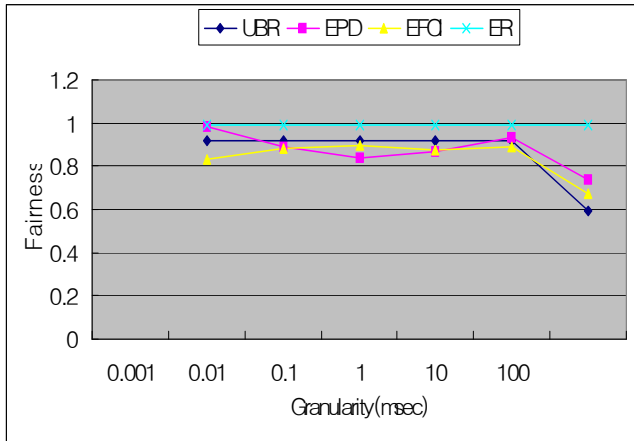
<b>TCP Application</b>	
Data Frame	= 4 Kbytes
<b>VBR</b>	
MMBP source(ON/OFF)	
EB(ON 상태의 평균 duration)	= 3000
ED(OFF 상태의 평균 duration)	= 3000
ED(분산된 셀의 평균 간격)	= 1.5 $\mu$ s
<b>TCP</b>	
Mean Packet Processing	= 300 $\mu$ s
Send/Receiver Buffer Size	= 64 Kbytes
Minimum Receiver Window Size	= 64 Kbytes
Minimum RTO	= 100 msec
Fast Retransmission and Recovery	= off
<b>Link</b>	
speed	= 150Mbps
between Switch Distance	= 56 Km
<b>ABR end system</b>	
Packet Processing Delay	= 1 $\mu$ s
Buffer Size	= 100 Kbytes
PCR(Peak Cell rate)	= 150 Mbps
Nrm	= 32, MCR = 0
FRTT	= 10 msec
<b>ABR Switch</b>	
Output Buffer Size	= 4096 cells
High threshold	= 300 cells
Low Threshold	= 200 cells
<b>UBR Switch</b>	
Output Buffer Size	= 4096 cells
EPD Threshold	= 3500 cells

UBR은 스위치의 버퍼 크기와 TCP 혼잡 제어방식에 의존하기 때문에 가 10msec에서 최고의 처리율을 보이다가 100msec를 넘어서면서 처리율이 급격히 떨어



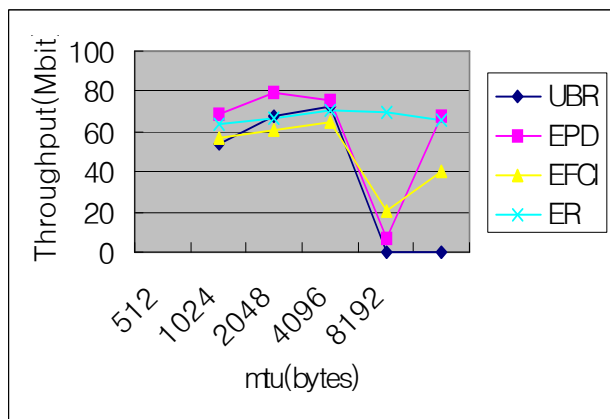


(그림6) Timer Granularity 에 따른 처리율 짐을 보이고 있다. 이는TCP의 timer granularity 가 지나치게 커도 ATM층의 성능을 악화시킬 수 있음을 보여주고 있다. 실험 결과 ABR 기반의 TCP에서 더 좋은 성능과 공정성을 보임을 알수 있다.

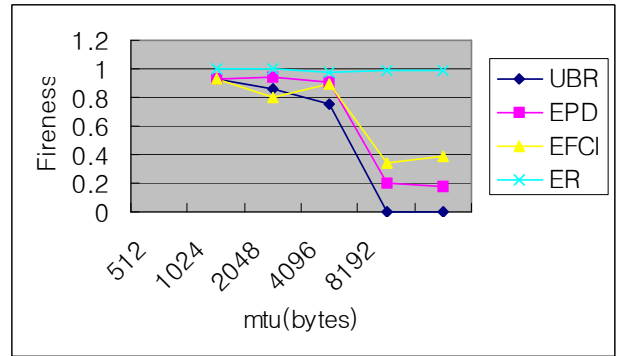


(그림7) Timer Granularity에 따른 공정성

(그림8)에서 볼 수 있듯이 MTU크기가 UBR과 ABR에 미치는 영향을 실험한 결과 처리율에 있어서 UBR이 영향을 많이 받고 있음을 보였다. 특히 MTU크기가 2048bytes를 넘어섰을 때 급격한 성능저하를 보였다. (그림9)에서는 나타나듯이 공정성측면에서도 ER을 제외한 나머지가 모두 많은 영향을 받았으며, 역시 MTU 크기가 2048bytes이상에서 급격한 성능저하나 나타났다.



(그림8) MTU 에 따른 처리율



(그림9) MTU 에 따른 공정성

## 5. 개선방안

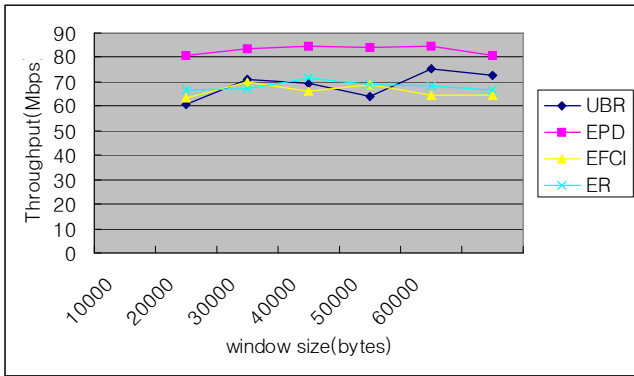
본 장에서는 4장의 결과 분석에 나타난 문제점과 비교 분석 결과를 가지고 파라미터의 변화에 따른 TCP 성능 개선 방안에 대해서 기술하였다.

### 1) UBR 상에서의 개선 방안

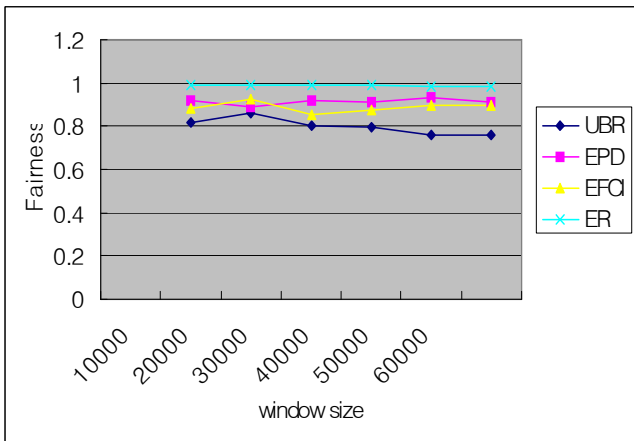
UBR이 TCP 윈도우크기나 세그먼트의 변화 등 TCP의 여러 파라미터에 영향을 받는다. (그림10)과(그림11)은 실험에서 가장 높은 성능을 보여준 timer granularity와 mtu값인 10msec, 2048bytes를 가지고 TCP window 크기 변화를 주었을 때, 처리율과 공정성의 변화에 관해 나타낸 것이다. 그 결과 UBR과 UBR+EPD의 처리율과 공정성 면에서 급격한 감소가 없이 좋은 성능을 보임을 알 수 있다.

### 2) ABR 상에서의 개선 방안

TCP의 주요 파라메타인 MTU와 timer granularity를 적절하게 선택함으로써 처리율과 공정성을 향상시킬 수 있다. ABR은 패킷 손실을 제거할 수 있기 때문에 ABR 기반 TCP는 좋은 성능을 보여 주었다. TCP 흐름제어가 ABR 전송률 제어가 서로 다르게 동작하므로 상호 작용을 막기 위해서는 MTU와 타이머 granularity와 윈도우 크기가 영향을 미치므로 적절한 파라미터 설정이 있어야 한다. (그림10)과 (그림11)에서는 UBR 동일한 timer granularity을 선택하고 TCP window 크기에 변화를 주었을 때, EFCI의 처리율과 공정성이 많이 개선되었음을 보였다.



(그림10) Window size 에 따른 처리율



(그림 11) window size 에 따른 공정성

## 6. 결론

본 논문에서는 시뮬레이션의 결과를 통해 다음과 같은 결론을 얻었다.

1) ABR은 패킷 손실을 제거할 수 있기 때문에 ABR 기반의 TCP가 가장 좋은 성능을 보임을 알 수 있다. TCP 흐름제어가 ABR 전송률 제어와 상호 작용하지 않도록 하기 위해 timer granularity와 MTU 크기, window 크기와 같은 TCP 파라미터가 중요한 요소임을 확인하였다.

2) UBR서비스상의 TCP traffic은 세그먼트가 커질 때, 성능이 급격히 악화된다. EPD는 처리율이 좋은 반면 공정성이 떨어지는 단점이 있음을 확인하였다.

3) VBR서비스상의 TCP 트래픽의 처리율이 순수 ABR/UBR서비스상의 처리율의 70%수준을 보여주었고, 공정성에 있어서는 매우 유사한 결과를 가져왔다.

## 참고문헌

[1] ATM Forum "ATM Traffic Management

Specification Version 4.0", Apr. 1996.

[2] Janey C. Hoe : Improving the startup behavior of congestion control scheme for TCP, proceedings of the ACM SIGCOMM, 1996

[3] S. Kalyanarman, R. Jain S. Fahmy, R. Goyal, ABR on ATM Backbone and with various VBR Traffic Patterns" submitted to ICC'97, pp8-12 Jun 1997.

[4] Sufian YOUSEF and Caroline STRANGE "TCP/IP over Challenges in Enterprise Network intergration ", IEEE International Conference on ATM, ICATM'98 1998, pp 447-453.

[5] O. Elloumi, H. Afifi, "RED Algorithm in ATM Networks", IEEE ATM '97 Workshop, pp.312-319, July 1997.

[6] S. Kalyanaraman, R. Jain, R. Goyal, S. Fahmy, and Vandalore, "The ERICA Switch Algorithm for ABR Traffic Management in ATM Networks", Submitted to IEEE/ACM Transactions on Networking, November 1997

[7] V. Jacobson, "Congestion Avoidance and Control", Proceedings of the SIGCOMM'98 Symposium , pp.314-332, August. 1998.

[8] Per Janhansson, Elin Wedlund, Johan M. Karlsson, "Interaction Between TCP Flow Control and ABR Rate Control", Proceedings of the IEEE ATM'97 workshop, May 1997.

[9] YATS simulation for ATM Networks, Dresden University Technology, 1997.

[10] Masatoshi Kawarasaki, Mika Ishizuka and Arata Koike "Dynamics of TCP flow control over High-Speed ATM Networks" , 0-7803-4788-9/98 , IEEE 1998.

[11] 박승섭, 육동철 “ABR 과 UBR 서비스상에서의 인터넷 프로토콜: 문제점, 해결방안, 그리고 성능평가”, 정보처리 논문지, 제6권 제 11S 호 pp. 3260-3268, 1999년 11월.