

# 전송률 변화 횟수를 줄이기 위한 개선된 e-PCRTT 알고리즘

Improved e-PCRTT Algorithm for Reducing the Number  
of Transmission Rate Changes

김종훈\*, 이면재\*\*, 김종진\*\*

Jong-Hoon Kim\*, Myoun-Jae Lee\*\*, Jong-Jin Kim\*\*

## 목 차

1. 서 론
  2. 관련 연구
  3. 제안 알고리즘
  4. 실험 결과
  5. 결론 및 추후 연구 방향
- \* 참고문헌

## 1. 서 론

최근에 정보화 시대와 더불어 정보의 중요성은 배가되고 있으며, 정보의 이동이 이루어지는 매개체 중의 하나인 통신 네트워크의 중요성 또한 새롭게 인식되고 있다. 단독(Stand-alone) 시스템으로밖에 실현할 수 없었던 멀티미디어 환경도 두 개 이상의 시스템이 통신망에 연결되어 멀티미디어 정보의 수신, 송신, 그리고 공유가 가능

\*제주교육대학교 컴퓨터교육과 조교수

\*\*홍익대학교 컴퓨터공학과 박사과정 수료

하게 되었다. CD-ROM 타이틀로 배포되던 멀티미디어 콘텐츠도 대용량의 서버와 멀티미디어 데이터 베이스의 출현, 고속 네트워크의 발전과 보급에 따라 통신망을 경유한 배포가 활발히 이루어지고 있다. 그 배경에는 광대역 네트워크를 기반으로 하는 전송망 기술의 빠른 발전을 들 수 있다. 광케이블로 대표되는 통신 매체의 고속화와 FDDI(Fiber Distributed Data Interface), ADSL(Asymmetric Digital Subscriber Line), Cable 망, 기가비트 이더넷(GigaBit Ethernet), ATM(Asynchronous Transfer Mode) 등 통신 네트워크의 고속화를 지원하는 기술의 발전에 따라 멀티미디어 데이터를 네트워크를 통해서 송수신하는 환경이 마련되고 있다. ISDN(Integrated Service Digital Network)등의 고속 네트워크 서비스의 출현으로 LAN(Local Area Network)보다 넓은 범위에서 멀티미디어 데이터의 교환이 자유로이 이루어질 수 있는 기반도 조성되고 있다. 또한 고성능의 마이크로 프로세서 개발과 보급에 의해 멀티미디어 데이터와 같은 방대한 양의 데이터를 실시간으로 전송, 서비스를 할 수 있게 되었다. 이와 같은 프로세서의 처리 능력 향상과 더불어, 버스와 입출력 장치 등의 외부장치 시스템 또한 고속화되었다. 이러한 기술적 발전 위에서 멀티미디어 정보를 저장하고 재생하는 기능을 가진 컴퓨터 시스템은 지리정보 시스템(GIS : Geographical Information System), 인터넷 방송국, 전자 도서관(Digital Libraries), 주문형 멀티미디어 정보 서비스, 사이버 대학(Cyber University), 전자상거래(EC : Electronic Commerce), 인터넷 쇼핑 등 다양하게 활용되고 있다. 그러나 많은 응용 프로그램과 멀티미디어 정보들을 안정적으로 공급하기 위해서는 기본적으로 시스템간에 연결되어 있는 통신 네트워크 대역폭이 멀티미디어 시스템이 제공하는 전송 대역폭(Transmission Bandwidth)보다 커야 한다[8]. 또한 일반적으로 멀티미디어 시스템에 요구되는 대역폭은 통신 네트워크의 대역폭보다 증감의 주기가 짧고 변화량이 크기 때문에 이 대역폭을 효과적으로 관리하는 것이 필요하다[9, 10].

멀티미디어 정보 교환 서비스의 방법은 크게 두 가지이다. 첫째는 사용자와 사용자 간(End to End)의 멀티미디어 통신 서비스이다. 이 통신 서비스는 원격회의와 같이 멀티미디어 정보를 실시간으로 공유하는 특징을 가지고 있다. 둘째는 사용자와 서버 간(Client to Server)의 멀티미디어 통신 서비스이다. 이 방법에서 멀티미디어 정보의 흐름은 주문형 비디오 서비스에서와 같이 서버에서 사용자에게 멀티미디어 데이터를 제공하는 방향과 사용자가 멀티미디어를 재생하기 위한 제어 신호가 서버로 전달되는 방향이 있다[2]. 특히 주문형 비디오 서비스에서 사용되는 멀티미디어 서비스의

경우에 클라이언트가 원하는 비디오를 서버에게 요청하면, 서버는 저장 장소인 디스크에서 비디오를 검색하여 해당 비디오 스트림(Video Stream)을 클라이언트에게 전송한다. 그러나 제한된 네트워크 대역폭 내에서 비디오 스트림을 보내는 경우에, 일정한 시간 내에 보내야 되는 데이터의 양이 네트워크 대역폭을 초과할 수 있으며 이러한 경우에는 사용자가 데이터의 지연으로 끊김이 있는 형태의 화질을 볼 수 있다. 따라서 제한된 네트워크 대역폭 내에서 전송되는 데이터의 양을 줄이는 압축 기법이 필요하게 되었다.

비디오 데이터를 압축하는 기법은 프레임당 비트 수가 동일한 고정 비트율(CBR:Constant Bit Rate)과 비트 수가 다른 가변 비트율이 있다[3]. 특히, 가변 비트율로 압축된 비디오는 프레임 당 비트수(BPF:Bits Per Frame)의 차이가 매우 심한 버스트(Burst)가 발생할 수 있다[3, 4, 5, 6]. 버스트가 제한된 네트워크 대역폭을 초과하게 된다면 네트워크 내에서 통신 과부화(Congestion)가 발생되어서 QoS(Quality of Service)를 보장하지 못한다. 이를 해결하기 위해서 서버는 가변 비트율 형태인 비디오 스트림을 고정 비트율 형태의 전송률로 조절해야 하며, 이러한 방식이 스무딩 기법이다[7]. 이러한 스무딩 알고리즘에는 구간의 크기가 일정하고 QoS를 유지하기 위해 요구되는 버퍼 크기를 최소화하는 것을 목적으로 하는 PCRTT(Piecewise Constant Rate Transmission and Transport) 방법[8]이 있는데, 언더플로우를 피하기 위해 모든 구간에 옵션 값을 적용하므로 불필요하게 버퍼 크기가 커질 수 있다[9,10,11]. e-PCRTT(enhanced-PCRTT) 알고리즘[9,12]은 버퍼 크기가 커질 수 있는 PCRTT 방법의 단점을 개선하였지만, PCRTT 방법에서와 같이 구간마다 전송률을 변화 시켜야 하므로 불필요하게 전송률 변화 횟수가 증가될 수 있다.

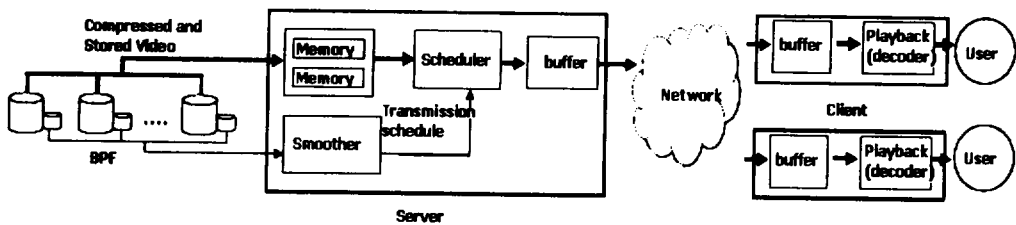
본 논문에서는 e-PCRTT의 단점을 해결하기 위해 현재 구간에 속한 프레임들과 연속적인 구간에 속한 프레임들을 1개의 전송률로 전송하는 경우에 클라이언트 버퍼에서 언더플로우 또는 오버플로우가 발생되지 않을 경우에는 다른 새로운 구간을 검색하고 QoS가 보장되지 않을 경우에는 현재 구간부터 QoS가 보장되는 구간까지를 동일한 전송률로 전송하는 알고리즘을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 살펴보고, 3장에서는 제안 알고리즘을 설명하고, 4장에서는 실험 결과를 기술하고, 5장에서는 결론 및 추후 연구 방향을 기술하였다.

## 2. 관련 연구

### 2.1 스무딩 기법을 위한 구조

스무딩 기법은 가변 비트율로 압축된 비디오 스트림을 일정한 비트율로 변화시키는 기법이다. 이를 위해서 서버에 위치하는 스무더(Smoother)가 프레임당 비트수, 클라이언트의 버퍼 크기, 그리고 네트워크의 대역폭에 대한 정보들을 얻는다. 이 정보들을 이용하여 스무더는 전송할 비디오 스트림에 대한 전송률을 계산한다. 그리고 서버는 계산된 전송률에 따라 비디오 스트림을 클라이언트에게 전송한다. 이때 전송되는 비디오 스트림이 재생하는 시간보다 일찍 도달해서 더 이상 클라이언트 버퍼에 저장할 수 없는 경우인 오버플로우(Overflow)와 전송되는 비디오 스트림이 재생하는 시간보다 늦게 도착하는 경우인 언더플로우(Underflow)가 발생하지 않도록 해야 한다. 스무딩 기법은 오버플로우나 언더플로우가 발생하지 않는 영역, 즉 QoS(Quality of Service)가 보장되는 영역 내에서 전송률을 조절하는 기법이다. (그림 1)은 스무딩 기법을 적용하기 위한 서버와 클라이언트간의 구조이다[13, 14].



(그림 1) 스무딩 기법 구조

### 2.2 스무딩 기법의 원리

(그림 2)는 스무딩 기법의 원리이다[15, 16]. X축은 시간 즉 프레임의 수를 의미하고 Y축은 누적된 비트 수를 의미한다. 그리고  $t$ 는  $t$  번째 프레임을 의미한다.  $V(t)$ 는 언더플로우 경계선으로 QoS를 보장할 수 있는 전송률 중에서 최소 전송률의 경계선이다. 즉, 서버가 비디오 스트림의 프레임을 언더플로우 경계선보다 낮은 전송률로 클라이언트 버퍼에 보내는 경우에 클라이언트는 서버로부터 받는 프레임을 자신의 속도보다 늦게 받기 때문에 정지된 화면을 보게 된다. 그러므로 비디오 스트림의

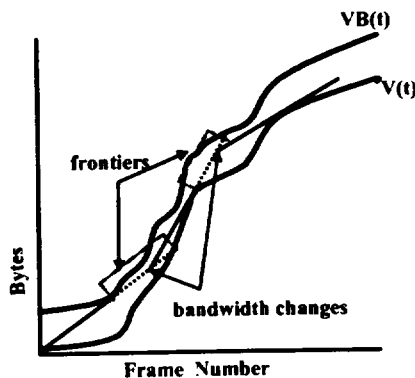
QoS을 보장하지 못한다.  $VB(t)$ 는 오버플로우 경계선으로 서버가 비디오 스트림의 프레임을 오버플로우 경계선보다 높은 전송률로 클라이언트 버퍼에 보내는 경우에 클라이언트는 서버로부터 받는 프레임을 자신의 속도보다 빠르게 받기 때문에 클라이언트 버퍼에 저장하지 못하고 프레임을 잃어버리는 상태가 발생된다. 따라서 클라이언트는 잃어버린 프레임을 재생하지 못하고 건너뛰는 형태로 화면이 보여진다. 역시 비디오 스트림의 QoS을 보장하지 못한다.  $b$ 는 클라이언트의 버퍼 크기이다.  $V(t)$ 와  $VB(t)$  사이 즉, QoS를 보장하는 영역을 제한 영역(Constraint region)이라고 한다. 이를 식으로 표현하면 식(1), 식(2), 그리고 식(3)과 같다.

$$V(t) = \sum_{i=0}^t f_i \quad \dots\dots(1)$$

$$VB(t) = b + \sum_{i=0}^t f_i \quad \dots\dots(2)$$

$$V(t) \leq \sum_{i=0}^t c_i \leq VB(t) \quad \dots\dots(3)$$

식(1), 식(2) 그리고 식(3)은 각 언더플로우 경계선, 오버플로우 경계선, 그리고 제한 영역을 의미한다. 식(1)의  $f_i$ 는  $i$  번째 프레임의 비트 수를 의미하며,  $V(t)$ 는 0부터  $t$  번째까지의 프레임들의 누적된 비트 수이다. 식(2)는  $V(t)$ 에서 클라이언트의 버퍼 크기( $b$ )를 더한 경계선이다. 식(3)은 오버플로우 또는 언더플로우가 발생되지 않는 영역인 QoS가 보장되는 전송률을 의미하며  $c_i$ 는  $i$  번째 프레임의 전송률을 나타낸다. 스무딩 알고리즘에서는 이 영역을 만족하면서 가변 전송률을 일련의 고정된 전송률로 변환하고 동일한 전송률로 전송할 수 있는 연속적인 프레임들을 검색해야 한

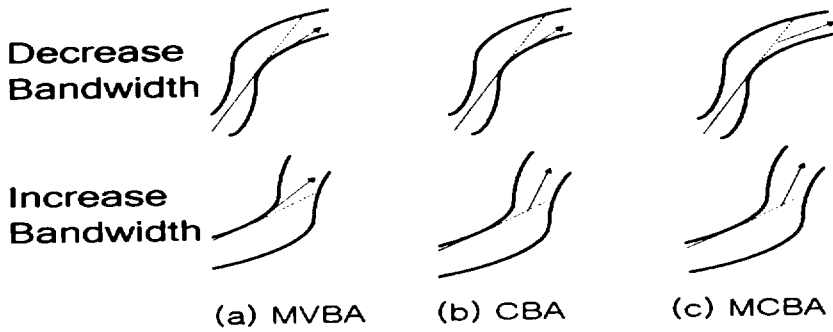


(그림 2) 스무딩 기법의 원리

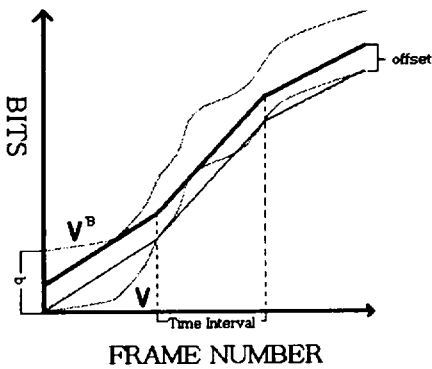
다. 이때 이 전송률에 의해 언더플로우 경계선(오버플로우 경계선)을 만나는 경우에 이 지점부터 오버플로우(언더플로우)를 발생시키는 프레임까지의 구간을 연장 구간(frontiers)이라 하고, 연장 구간에 속한 프레임들 중에서 스무딩 알고리즘의 목적에 적합한 프레임을 검색하여 이 프레임에서 전송률을 변화시키며, 동일한 전송률로 전송하는 구간을 런(Run)이라 한다. 이러한 스무딩 기법의 원리를 바탕으로 QoS를 만족하면서 전송률 변화 횟수, 전송률 변화량, 그리고 버퍼 크기와 같은 특정 요소를 최적화하기 위한 목적으로 다양한 스무딩 알고리즘들이 연구되어 왔다.

### 2.3 기존 스무딩 알고리즘

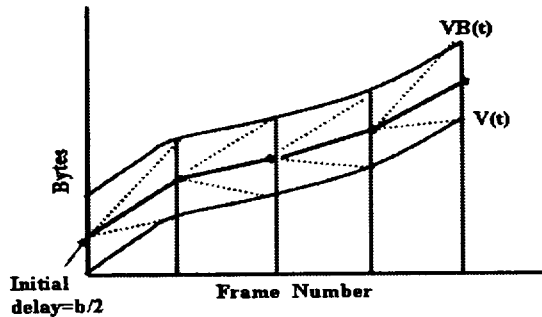
(그림 3)은 CBA(Critical Bandwidth Allocation), MCBA(Minimum Changes Bandwidth Allocation), MVBA(Minimum Variability Bandwidth Allocation) 알고리즘의 전송률 조절 방법을 나타낸다[19]. CBA 알고리즘[17]에서는 현재 전송률에 의해 오버플로우가 발생된다면 연장 구간의 첫 프레임을 다음 런의 시작 프레임으로 설정하며, 현재 전송률에 의해 언더플로우가 발생하는 경우에는 연장 구간에 속한 프레임들 중에서 새로운 전송률로 언더플로우나 오버플로우가 발생되지 않고 가장 많은 프레임을 전송할 수 있는 프레임을 검색하여 이 프레임을 다음 런의 시작 프레임으로 설정하여 전송률 증가 횟수를 최소화한다. MCBA 알고리즘[18]의 목적은 전송률 변화 횟수의 최소화인데, 이는 서버와 네트워크 사이의 전송률 조절에 관련된 통신 오버헤드를 감소시키고 서버에 저장된 비디오 데이터에 대한 접근 스케줄링을 용이하게 한다. 이를 위해 다음 런의 전송률을 감소 또는 증가시켜야 할 때에 연장 구간에 있는 프레임 중에서 새로운 전송률에 의해 가장 많은 프레임을 전송할 수 있는 프레임을 검색하여 이 프레임을 다음 런의 시작 프레임으로 설정하지만, 알고리즘의 계산 시간이 크다[19]. MVBA 알고리즘[20]은 전송률 변화량을 최소화하는 것이 목적이며, 시간 복잡도가  $O(n^2)$ 인 알고리즘[20]과  $O(n)$ 인 알고리즘[21]이 있다. 공유되는 네트워크의 자원들에게 이전 전송률보다 증가량이 큰 전송률을 요구하면 요구되는 전송률을 예약하지 못할 수도 있다. 그래서 이전 전송률보다 전송률의 증감이 완만한 전송 계획을 세우는 것이 좋다. 이를 위해 MVBA 알고리즘은 다음 런의 전송률을 증가 또는 감소시켜야 할 때에 연장 구간의 첫 프레임을 다음 런의 시작 프레임으로 설정하는데 전송률 변화 횟수가 상대적으로 많다[12].



(그림 3) MVBA, CBA 그리고 MCBA의 전송률 조절 과정



(그림 4) PCRTT 알고리즘의 전송률 조절 방법



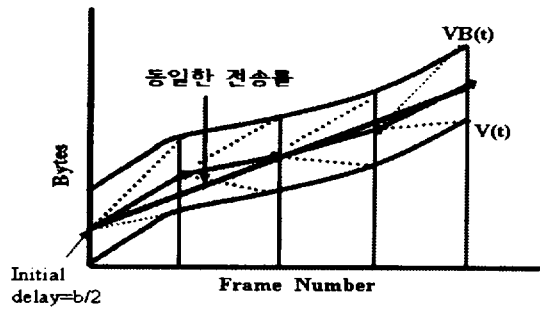
(그림 5) e-PCRTT 알고리즘의 전송률 조절 방법

(그림 4)는 PCRTT-휴리스틱 방법에서의 전송률 조절 방법[8,9,10,11]을 나타내는데, 전송률 변화 횟수가 주어지면 전체 비디오 스트림을 구성하는 프레임 수를 이 값으로 나눈 몫이 1개의 전송률로 보낼 수 있는 런 즉 구간의 크기가 된다. 이와같이 설정된 구간 크기로 비디오 스트림을 분할하여, 각 구간의 시작 프레임의 언더플로우 경계선을 시작점으로 하고 구간의 끝 프레임의 언더플로우 경계선을 종점으로 하는 연결선을 만드는데, 이 연결선이 해당 구간의 임시 전송률이다. 이와같이 모든 구간에서의 임시 전송률을 계산하여, 이 전송률에 의해 언더플로우가 가장 크게 발생하는 프레임이 속한 구간의 프레임들이 언더플로우를 피할 수 있는 최소의 오프셋 값을 구하여 이 오프셋 값을 모든 임시 전송률에 더해 각 구간의 전송률로 설정한다. 그러나 이 방법은 특정 구간에서만 언더플로우가 심하게 발생되더라도 모든 구간에 같은 오프셋 값을 더하므로 요구되는 버퍼 크기가 커질 수 있다[19]. PCRTT 알고리즘의 단점을 개선한 e-PCRTT 알고리즘[9, 12]에서는 (그림 5)와 같이 초기 지연 시간 동안

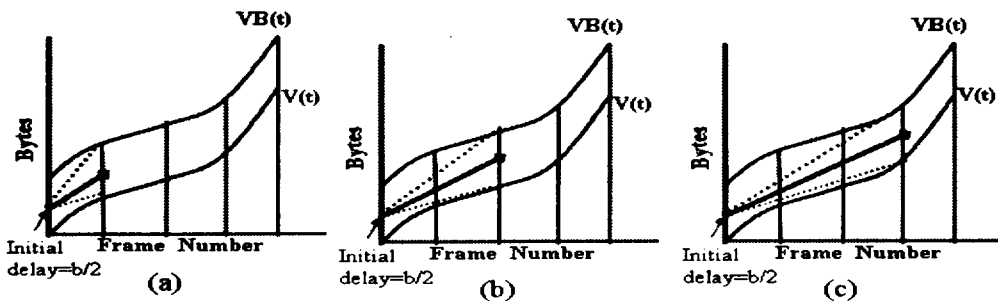
클라이언트 버퍼에 버퍼 크기의 1/2에 해당하는 바이트 수를 미리 저장하고 각 구간에서 언더플로우가 발생되지 않을 최소 전송률과 오버플로우가 발생되지 않을 최대 전송률의 평균을 해당 구간의 전송률로 설정한다.

### 3. 제안 알고리즘

구간의 크기가 일정하고 구간에 속한 프레임들을 1개의 전송률로 전송하는 e-PCRTT 알고리즘의 경우에 구간의 크기가 작게 구성되는 경우에는 (그림 6)에서와 같이 불필요하게 전송률 변화 횟수가 증가될 수 있다.



(그림 6) e-PCRTT 알고리즘에서 전송률 조절의 문제점



(그림 7) 제안 알고리즘의 전송률 조절 방법

따라서, 제안 알고리즘에서는 런의 시작 구간부터 새로운 구간의 끝 프레임까지 전송률을 계산하여 언더플로우 또는 오버플로우가 발생되지 않는 경우에는 연속되는 구간들을 계산되는 전송률로 설정하고 그렇지 않는 경우에는 새로운 전송률로 설정하여 전송률 변화 횟수를 줄이도록 한다. (그림 7)은 제안 알고리즘의 전송률 조절



방법이다. (그림 7) (a)에서는 첫 번째 구간에서 언더플로우가 발생되지 않을 최소 전송률과 오버플로우가 발생되지 않을 최대 전송률의 평균을 첫 번째 구간의 전송률로 설정하는 것을 보여준다. 첫 번째 구간의 시작 프레임부터 2 번째 구간의 끝 프레임까지 언더플로우가 발생되지 않을 최소 전송률과 오버플로우가 발생되지 않을 최대 전송률의 평균을 2개 구간의 전송률로 설정하여도 (그림 7) (b)와 같이 QoS가 보장되고 3번째 구간까지도 (그림 7) (c)와 같이 동일한 전송률로 설정하여도 QoS가 보장된다. 이 과정을 QoS가 보장되지 않을 구간까지 수행하여 런의 시작 구간부터 QoS가 보장되는 구간까지에 속한 프레임들을 1개의 전송률로 전송한다.

〈표 1〉은 제안 알고리즘이다.  $ts$ 는 새로운 전송률로 전송을 하는 구간의 시작 프레임이다.  $q$ 는 버퍼에 채워져 있는 바이트 수이다.  $C$ 는 구간의 시작 프레임부터 현재 검색되는 프레임까지 오버플로우가 발생되지 않을 최대 전송률이고,  $C$ 은 최소 전송률이다. 단계 (4)에서 (8)까지는 새로운 전송률로 전송할 수 있는 구간의 시작 프레

〈표 1〉 제안 알고리즘

```

PROCEDURE proposed algorithm(){
(1)   $ts=0, q=b/2$ 
(2)  for( $i=0;i \leq N;i+=interval\_size$ ) {
(3)     $C=0; C = \infty$ 
(4)    for( $j=ts;j \leq interval+i;j++$ ) {
(5)       $v=(V[j]-V(ts)+q)/(j-ts)$ 
(6)      if ( $C < v$ )  $C=v$ 
(7)       $v=(VB[j]-V[ts]+q)/(j-ts)$ 
(8)      if ( $C > v$ )  $C=v$ 
(9)    }
(10)   $r=(Cmax+Cmin)/2$ 
(11)  if ( $QoS\_check(r)$ ) continue
(12)  compute  $C, C$  from  $ts$  frame to  $i$  frame
(13)   $r=(C+C)/2$ 
(14)  output( $ts \sim i, r$ ),
(15)  compute  $q$ 
}
}

```

임부터 현재 검색되는 구간의 끝 프레임까지 C와 C을 구하는 과정이다. 단계 (10)는 오버플로우가 발생되지 않을 최대 전송률과 언더플로우가 발생되지 않을 최소 전송률의 평균으로 전송률을 계산하는 과정이다. QoS\_Check(r) 함수는 ts 프레임부터 현재 검색되는 구간의 끝 프레임까지 전송률 r로 전송했을 경우에 QoS를 만족하는 여부를 판단하는 함수이다. 단계 (11)에서는 QoS\_Check(r) 함수가 참값을 반환하는 경우에는 새로운 구간을 검색하며 거짓 값을 반환하는 경우에는 단계 (12)부터 단계 (13)에서와 같이 전송률을 계산한다. output(ts~i, r)는 프레임 ts부터 프레임 i까지의 전송률을 r로 설정하는 함수이다.

#### 4. 실험 결과

C 언어로 스무딩 알고리즘들을 구현하여 실험하였으며, <표 2>는 실험에 사용된 MPEG-2로 저장된 Croc.Dundee 비디오 소스의 파라미터들[22, 23]이다. Length는 비디오 재생 시간, Ave Frame Size는 각 프레임들의 평균 바이트 수, Max Frame Size와 Min Frame Size는 프레임들의 바이트 수들 중에서 가장 큰 값과 작은 값이고, Total Size는 프레임들의 바이트 수의 합이다. 또한, Std Dev는 프레임들의 바이트 수에 대한 표준 편차로써 이 값이 클수록 프레임들 사이에 바이트 수의 변화가 심하다. e-PCRTT 알고리즘에서 구간 크기의 크기는 언더플로우 또는 오버플로우가 발생되지 않으면서 구성할 수 있는 최대 구간 크기로 설정하였다.

<표 2> croc.dundee 비디오 소스의 파라미터

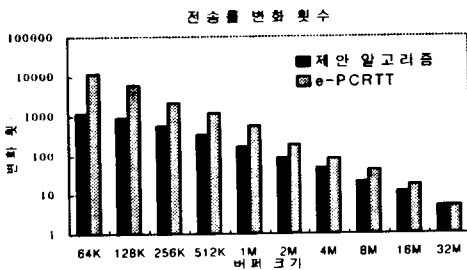
Video Clip Name	Length (min)	Ave Frame Size (KB)	Max Frame Size (KB)	Min Frame Size (KB)	Total Size (KB)	Std Dev (KB)
Croc.Dundee	94	10.52	18.98	1.233	1,773,379	10.764

(그림 8)은 전송률 변화 횟수를 비교한 결과이다. 전송률 변화 횟수는 적을수록 서버와 네트워크간의 전송률 예약에 관한 통신 오버헤드가 감소될 수 있고 서버에 저장되어 있는 비디오 데이터를 추출하여 서비스 하는 과정이 단순해 질수 있다[19].

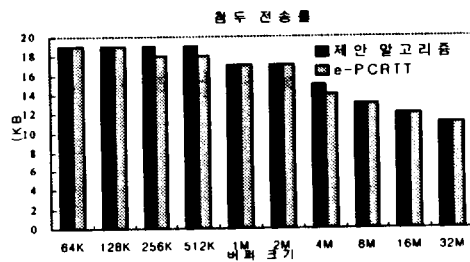
제안 알고리즘의 전송률 변화 횟수가 e-PCRTT 알고리즘보다 적는데, 이는 e-PCRTT 알고리즘에서는 구간마다 전송률을 변화시켜야 하기 때문이다.

(그림 9)은 첨두 전송률을 비교한 결과이다. 첨두 전송률은 서버에서 계산되는 전송률 중 가장 높은 전송률이다. 첨두 전송률이 낮을 수록 제한된 대역폭을 갖는 네트워크 자원, 서버에서는 다수의 비디오 스트림을 서비스 할 수 있다[19]. 제안 알고리즘이 버퍼 크기가 256K, 512K, 4M인 경우에 e-PCRTT 알고리즘에 비해 높다. 이는 제안 알고리즘에서는 2개 이상의 구간을 1개의 전송률로 보내기 때문에 구간을 구성하는 프레임들의 바이트 수가 차이가 큰 구간들을 1개의 전송률로 보내는 경우에 이 전 구간에 비해 급격하게 높은 전송률이 요구될 수 있기 때문이다.

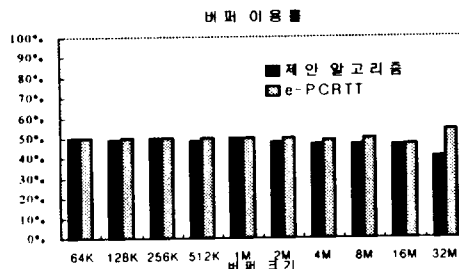
(그림 10)은 버퍼 이용률을 비교한 결과이다. 버퍼 이용률은 다수의 응용 프로그램들이 1개의 버퍼를 공유하고 있을 경우에 다른 응용프로그램들이 사용할 수 있도록 적은 것이 좋다[19]. 제안 알고리즘의 버퍼 이용률이 e-PCRTT 알고리즘보다 낮다. 특히 e-PCRTT 알고리즘에서 버퍼 크기가 32M인 경우 아주 큰데, 이는 서버에서 계산되는 전송률이 오버플로우 경계선에 편중되기 때문이다.



(그림 8) 전송률 변화 횟수 비교



(그림 9) 첨두 전송률 비교



(그림 10) 버퍼 이용률 비교

## 5. 결론 및 추후 연구 방향

구간의 크기가 일정하면서 버퍼가 커질 수 있는 PCRTT 알고리즘의 단점을 개선한 e-PCRTT 알고리즘은 PCRTT 알고리즘에서와 같이 구간의 크기가 일정하고 구간마다 새로운 전송률을 설정한다. 그러므로 구간의 개수가 크게 주어지는 경우에 구간을 구성하는 프레임의 개수가 적어지게 되므로 불필요하게 전송률이 증가될 수 있다.

본 논문에서는 이러한 e-PCRTT 알고리즘의 문제점을 개선하기 위해 런의 시작 구간부터  $g$  현재 구간까지를 1개의 전송률로 전송할 경우에 QoS를 만족하는 경우에는 새로운 구간을 검색하고 QoS를 만족하지 않을 경우에는 런의 시작 구간부터 QoS를 만족시키는 구간까지를 1개의 전송률로 전송하여 e-PCRTT 알고리즘보다 전송률 변화 횟수를 줄이는 알고리즘을 제안하였다. 그리고 e-PCRTT 알고리즘과 전송률 변화 횟수, 침투 전송률, 버퍼 이용률을 비교하였다. 제안 알고리즘은 e-PCRTT 알고리즘에 비해 전송률 변화 횟수와 버퍼 이용률은 낮고 침투 전송률은 대체적으로 우수한 결과를 보였다.

추후에는 버퍼 크기가 다양하게 주어지고 전송률 변화량, 침투 전송률 이용률등의 다양한 요소로 제안 알고리즘과 e-PCRTT 알고리즘의 성능을 비교 분석할 예정이다.

## 〈참고문헌〉

- [1] J. Son, "Storage and Retrieval Scheme for VBR streams in Video Servers", Ph. D. Univ. Seoul, Aug 1997.
- [2] S. Ramesh, et. al., "Multicast with cache: An Adaptive Zero-Delay VoD Service", In IEEE INFOCOM 2001.
- [3] D. Le Gall, "MPEG: A video compression standard for multimedia applications", Communications of the ACM, vol.34, pp.47-58, April 1991.
- [4] J. Zhang and J. Hui, "Applying traffic smoothing techniques for quality of service control in VBR video transmissions", Computer Communications, pp.375~389, April 1998.
- [5] J. Zhang and J. Y. Hui, "Traffic Characteristics and Smoothness Criteria in VBR Video Traffic Smoothing", in Proc. of the ICMC and Systems, June 3-6, Ottawa, Ontario, Canada, IEEE Computer Society, 1997.
- [6] P. Thiran, et. al., "Network calculus applied to optimal multimedia smoothing" in INFOCOM, 2001.
- [7] W. Feng, F. Jahanian, S. Sechrest, "An Optimal Bandwidth Allocation Strategy for the Delivery of Compressed Preencoded Video", ACM/Springer-Verlag Multimedia Systems, vol. 5, no. 5, pp. 297-309, Sept 1997
- [8] J. McManus and K. Ross, "Video on demand over ATM: Constant-rate Transmission and Transport" in proc. of ACM SIGMETRICS, pp.222~231, May 1996.
- [9] Ofer Hadar Reuven Cohen, "PCRTT Enhancement for Off-Line Video Smoothing", The Journal of Real Time Imaging. Vol. 7, No. 3, pp.301~314, June 2001.
- [10] Wu-chi Feng, "A Comparison of Bandwidth Smoothing Techniques for the Transmission of Preencoded Compressed Video", in Proc. IEEE INFOCOM, pp. 58~66, April 1997.
- [11] W. Feng, et.al., "Optimal Buffering for the Delivery of Compressed Preencoded Video", Proc. of the IASTED/ISMM Intl Conference on Networks, January 1995.

- [12] O. Hadar S. Greenberg, "Statistical multiplexing and admission control policy for smoothed video streams using e-PCRTT algorithm", International Conference on Information Technology: Coding and Computing, March 2000.
- [13] D. Wu, et. al., "Streaming Video over the internet: Approaches and Directions", IEEE Trans.on C&S for Video Tech., Feb. 2001.
- [14] P. Thiran, et. al., "Network Calculus applied to optimal multimedia smoothing", in INFOCOM 2001.
- [15] W. Feng, "Rate-constrained bandwidth smoothing for the delivery of stored video", in SPIE Multimedia Networking and Computing, 1997.
- [16] M. Lee, J. Kwak, H. Song, D. park, "An Efficient Transmission Rate Control Algorithm for MPEG VOD Services", Journal of the KCIES, VOL. 3, No. 8, August, 2002.
- [17] W. Feng, S. Sechrest, "Critical Bandwidth Allocation for the Delivery of Compressed Prerecorded Video", Computer Communications, Vol 18, No. 10, pp. 709-717, Oct. 1995.
- [18] W. Feng, F. Jahanian, S. Sechrest, "An Optimal Bandwidth Allocation Strategy for the Delivery of Compressed Preencoded Video", ACM/Springer-Verlag Multimedia Systems, vol. 5, no. 5, pp.297-309, Sept 1997.
- [19] W. Feng and J. Rexford, "Performance evaluation of smoothing algorithms for transmitting prerecorded VBR video", IEEE Trans. on Multimedia, September 1999.
- [20] J. D. Salehi, et. al. "Supporting stored video: Reducing rate variability and end-to-end resource requirements through optimal smoothing", in Proc. of ACM SIGMETRICS, pp.222-231, May 1996.
- [21] J. D. Salehi, "Scheduling Network Processing on Multimedia and Multiprocessor Servers", Ph. D. dissertation, Univ. Massachusetts, Amherst, Sept 1996.
- [22] Wu-Chi Feng, "VIDEO-ON-DEMAND SERVICES: EFFICIENT TRANSPORTATION AND DECOMPOSITION OF VARIABLE BIT RATE VIDEO", PH. D, Computer Science and Engineering, 1996.
- [23] <http://www.cis.ohio-state.edu/~wuchi>