

멀티미디어 데이터 전송을 위한 효율적인 전송 계획

임재환*, 방기천**

요약

스무딩은 가변 비트율로 저장된 비디오 데이터를 고정 비트율로 변환하는 전송 계획이다. 이러한 스무딩 알고리즘들에는 CBA, MCBA, MVBA, PCRTT, e-PCRTT 알고리즘 등이 있다. 이 중에서 PCRTT 알고리즘을 개선한 e-PCRTT 알고리즘은 전송률 변화 횟수가 주어지고 런의 크기가 일정하다. 이는 버퍼 크기가 크고 런의 크기가 작은 경우에 불필요한 전송률의 변화를 필요로 한다. 본 논문에서는 e-PCRTT 알고리즘의 단점을 해결하기 위해 동일한 전송률로 보다 많은 구간을 전송할 수 있게 e-PCRTT 알고리즘을 개선한 알고리즘을 제안하고 성능을 평가한다. 제안 알고리즘의 성능은 e-PCRTT 알고리즘과 전송률 변화 횟수, 침투 전송률, 전송률 변화량 등의 평가 요소들로 비교하여 우수함을 보였다.

An Efficient Transmission Plan for Multimedia Data Transmission

Jae-Hwan Lim*, Kee-Chun Bang**

Abstract

Smoothing is a transmission plan where variable rate video data is converted to a constant bit rate stream. Among them are CBA, MCBA, MVBA, e-PCRTT and others. e-PCRTT algorithm, which was improved from PCRTT, restricts the number of rate changes with fixed-size run. This causes unnecessary rate changes when run size is small and buffer size is large.

In this paper, to overcome a shortcoming of e-PCRTT algorithm, a smoothing algorithm is proposed, which is improved from e-PCRTT, where a transmission rate transmits more intervals as possible. Experiments demonstrated that the proposed algorithm outperformed e-PCRTT algorithm. In order to show the performance, various evaluation factors were used such as the number of transmission rate changes, peak rate, transmission rate variability and so on.

Keywords : Multimedia Data Transmission, Transmission Plan, Smoothing, Variable Bit Rate

1. 서론

인터넷의 사용 초기에는 용량이 작은 텍스트 데이터가 주로 전송되었다. 그러나 최근에는 용량이 큰 멀티미디어 데이터 전송에 대한 필요가 증대되면서 효율적인 멀티미디어 서비스에 대한 연구가 활발히 진행되고 있다. 멀티미디어 데이

터 스트림은 저장 장치의 효율적인 관리를 위하여 압축된 형태로 VOD 서버에 저장된다. 이러한 압축 기술에는 프레임 당 비트수가 일정한 CBR(Constant Bit Rate)과 프레임 당 비트수가 다른 VBR(Variable Bit Rate)이 있다. VBR방법은 CBR방법에 비해 비디오 화질이 우수하고 압축률이 높지만 버스트(Burst), 즉 특정 프레임이 다른 프레임보다 프레임 당 비트수가 지나치게 많은 경우가 발생하여 데이터를 전송하려는 경우에는 QoS(Quality of Service)를 보장하기 위해 급격하게 높은 전송률이 요구될 수 있다[1,2]. 또한 VOD 서버와 클라이언트 사이에는 한 개 이상의 네트워크가 존재하므로 VOD 서버에서 여러 네트워크를 통해 클라이언트로 비디오 스트림을 전송할 경우에 QoS를 보장하기 위해서

※ 제일저자(First Author) : 임재환

접수일자:2007년01월07일, 심사완료:2007년03월10일

* 남서울대학교 멀티미디어학과 교수

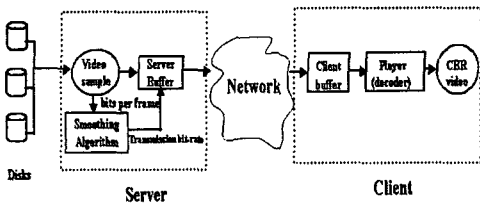
ezra@nsu.ac.kr

** 남서울대학교 멀티미디어학과 교수

▣ 이 논문은 2004년 남서울대학교 학술연구비 지원에 의하여 연구되었음

는 네트워크 상에서 전송률이 일정하게 지원되어야 하고 지연시간이 최소화되어야 한다. 그러나 멀티미디어 서비스를 이용하는 사용자의 수가 인터넷 통신망의 속도보다 빠른 속도로 증가되고 있어서 네트워크 혼잡(Congestion), 지연(Delay), 지터(Jitter), 그리고 패킷 손실로 인하여 QoS를 보장하기가 어렵다. 이러한 문제들의 해결책으로 VBR 형태의 비디오 데이터를 전송할 때 클라이언트의 버퍼에서 언더플로우와 오버플로우가 발생하지 않을 범위내에서 전송률을 계산하여 해당 전송률로 비디오 스트림을 보내는 스무딩 기법에 관한 연구가 시작되었다.

(그림 1)은 스무딩 기법을 위한 구조이다. 클라이언트가 서버에게 비디오 스트림을 요청하면, 서버는 클라이언트에서 요청한 비디오 스트림을 검색하여 비디오 스트림을 구성하는 모든 프레임들의 정보, 즉 프레임 당 비트 수를 스무딩 알고리즘에게 전달한다. 그러면 스무딩 알고리즘은 프레임 당 비트 수와 클라이언트 버퍼 크기를 고려하여 클라이언트 버퍼에서 언더플로우와 오버플로우가 발생되지 않을 전송률, 즉 QoS가 보장되는 전송률을 설정하여 해당 비디오 스트림을 클라이언트에게 전송한다.



(그림 1) 스무딩 기법의 구조

이러한 스무딩 기법을 바탕으로 여러 알고리즘들이 연구되어져 왔다. 구현의 용이함과 네트워크의 전송률 예약에 대한 통신 비용을 줄이려는 목적을 갖는 CRTT(Constant Rate Transmission and Transport) 알고리즘[6]에서는 런의 개수가 1개이어서 클라이언트 버퍼 크기와 초기 지연 시간(Initial Delay)이 크고, 침투 전송률의 이용률이 낮을 수 있다. 이러한 단점을 개선하여 전송률 변화 횟수 즉 런의 개수를 설정하므로써 CRTT 방법보다 버퍼 크기를 줄이려는 PCRTT-휴리스틱(Piecewise Constant Rate Transmissi

on and Transport-Heuristic) 방법[6]은 언더플로우를 피하기 위해 모든 구간에 동일한 옵션값을 적용하므로 버퍼 크기가 아직도 커질 수 있다. e-PCRTT(enhanced-PCRTT) 알고리즘[8]은 버퍼 크기가 커질 수 있는 PCRTT-휴리스틱 방법의 단점을 개선하였지만, PCRTT-휴리스틱 방법에서와 같이 전송률 변화 횟수에 대한 제한이 있고 런의 크기가 일정하다. 그러므로 전송률 변화 횟수가 큰 경우에는 런의 크기가 작아져서 연속된 런에서 전송률의 변화가 필요하지 않을 경우에도 강제적으로 변화를 시켜야 하는 경우가 발생될 수 있다.

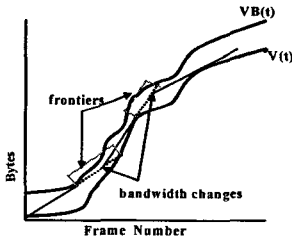
본 논문에서는 e-PCRTT 알고리즘의 단점을 개선한 알고리즘을 제안하고 성능을 평가한다. 즉, 제안 알고리즘에서는 새로운 전송률이 적용되는 첫 구간부터 현재 검색되는 구간까지에 속한 프레임들에 대하여 언더플로우가 발생되지 않을 최소 전송률과 오버플로우가 발생되지 않을 최대 전송률의 평균을 계산하여 이 전송률로 첫 구간부터 현재 검색되는 구간까지를 전송하는 경우에 QoS가 보장되는 경우에는 새로운 구간을 검색하고 QoS가 보장되지 않을 경우에는 현재 검색되는 구간까지를 동일한 전송률로 전송할 수 있는 마지막 구간으로 설정한다.

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

2. 관련연구

(그림 2)는 스무딩 기법의 원리를 나타내는데[2, 7], X축은 시간 즉 프레임 번호이며 Y축은 바이트 수이다. 식(1)의 $V(t)$ 는 언더플로우 경계선으로 프레임 0부터 프레임 t 까지의 누적된 바이트 수이며, f_i 는 프레임 i 의 바이트 수이다. 서버가 이 경계선보다 낮은 전송률로 프레임을 전송하면 클라이언트에서 언더플로우가 발생되어 QoS를 보장할 수 없다. $VB(t)$ 는 오버플로우 경계선으로 식(2)와 같이 언더플로우 경계선에 클라이언트 버퍼 크기 b 를 더한 값으로 표현된다. 서버가 오버플로우 경계선 보다 큰 전송률로 프레임을 전송하면 클라이언트에서 오버플로우가 발생

되어 QoS를 보장할 수 없다. 따라서, 서버에서 계산된 전송률이 QoS를 만족하려면 언더플로우 경계선과 오버플로우 경계선 영역내에 있어야 하며, 이 경우에 스무딩 알고리즘은 언더플로우 경계선과 오버플로우 경계선 사이의 영역에서 동일한 전송률로 전송할 수 있는 연속적인 프레임들을 검색해야 한다. 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 변화시킨다. 이 경우에 동일한 전송률로 전송하는 프레임들의 그룹을 런(Run)이라 한다.



(그림 2) 스무딩 기법의 원리

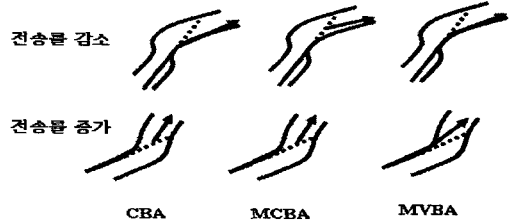
$$V(t) = \sum_{i=0}^t f_i \dots \dots \dots (1)$$

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

이러한 스무딩 기법의 원리를 바탕으로 전송률 변화 횟수, 첨두 전송률, 버퍼 크기 등의 특정 요소를 최적화하는 목적으로 다양한 스무딩 알고리즘이 연구되어 왔다. (그림 3)은 CBA, MCBA, MVBA 알고리즘의 전송률 조절 방법을 나타낸다[7].

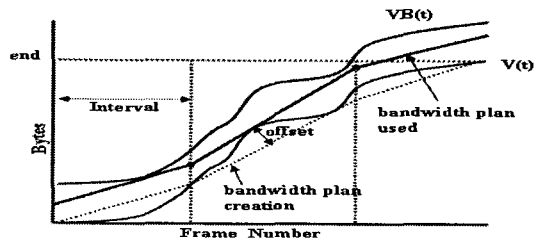
CBA 알고리즘[4]은 전송률 증가 횟수를 최소화 하는 것이 목적이며, MCBA 알고리즘[2]의 목적은 전송률 변화 횟수의 최소화이고, MVBA 알고리즘[3,5]의 목적은 전송률 변화량을 최소화 하는 것이다. 또한, PCRTT 알고리즘[6]의 목적은 일정한 전송률 변화 주기와 클라이언트 버퍼 크기를 최소화 하는 것이다. 특히, 전송률 변화

횟수가 주어지는 PCRTT 알고리즘은 전체 비디오 스트림을 구성하는 프레임 개수를 전송률 변화 횟수로 나눈 몫을 1개의 구간 크기로 설정한다. 이 구간 크기로 비디오 스트림을 분할하고 각 구간의 시작 프레임에서 구간의 끝 프레임과 언더플로우 경계선과 교차하는 지점을 연결하는데, 이 연결선이 해당 구간의 임시 전송률이다.

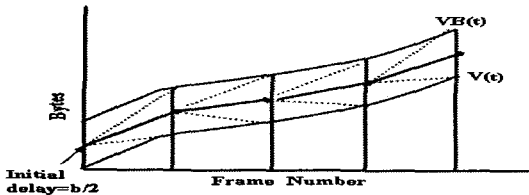


(그림 3) CBA, MCBA, MVBA 전송률 조절방법

이와 같이 모든 구간의 임시 전송률을 계산하고 언더플로우가 가장 크게 발생된 프레임이 속한 구간의 프레임들이 언더플로우를 피할 수 있는 최소의 오프셋 값을 구하고 이 오프셋 값을 모든 임시 전송률에 더해 각 구간의 전송률을 설정한다. PCRTT 알고리즘은 특정 구간만 언더플로우가 심하게 발생되었을 때에도 모든 구간에 동일한 오프셋 값을 더하므로 버퍼 이용률과 비디오 데이터의 전송에 요구되는 버퍼 크기가 커질 수 있다. 이 문제점을 개선하기 위하여 e-PCRTT 알고리즘은 각 구간의 언더플로우가 발생되지 않을 최소 전송률과 오버플로우가 발생되지 않을 최대 전송률의 평균을 해당 구간의 전송률로 설정한다. (그림 4)는 PCRTT 알고리즘의 전송률 조절 과정이고[6], (그림 5)은 e-PCRTT 알고리즘의 전송률 조절 과정이다[8].



(그림 4) PCRTT 알고리즘의 전송률 조절 과정



(그림 5) e-PCRTT 알고리즘의 전송률 조절 과정

3. 제안 알고리즘

구간마다 전송률이 변화되어야 하는 e-PCRTT 알고리즘은 불필요하게 전송률 변화 횟수가 증가 될 수 있다. (그림 5)는 전송률 변화 횟수 제한이 4인 경우에 e-PCRTT 알고리즘의 전송률 조절 과정을 보여주는데, 구간 크기가 고정적이고 구간 당 1개의 전송률로 보내야 하는 알고리즘의 특징 때문에 첫 번째 구간부터 4 번째 구간까지를 1개의 전송률로 보낼 수 있는 데도 전송률 변화가 4회 발생된다.

본 논문에서는, 이러한 e-PCRTT 알고리즘의 문제를 개선하기 위해 구간의 크기가 동일하면서도 동일한 전송률로 연속적인 구간들을 전송할 수 있는 알고리즘을 제안한다.

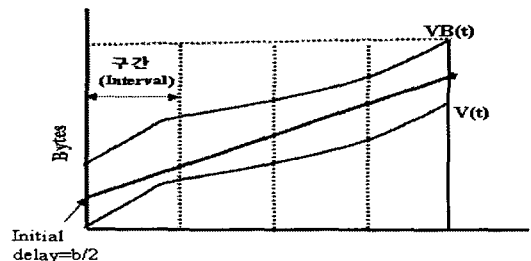
<표 1>은 제안 알고리즘이다. ts 는 새로운 전송률이 적용되는 구간의 첫 번째 프레임이고, q 는 버퍼에 채워져 있는 바이트 수이다. $interval_size$ 는 주어지는 전송률 변화 횟수 num 을 비디오 스트림을 구성하는 총 프레임 개수 N 으로 나눈 몫이다. C_{max} 는 구간의 시작 프레임부터 현재 검색되는 프레임까지 오버플로우가 발생되지 않을 최대 전송률이고, C_{min} 은 최소 전송률이다. 단계 (5)에서 (9)까지는 새로운 전송률로 전송할 수 있는 구간의 시작 프레임부터 현재 검색되는 구간의 끝 프레임까지 C_{max} 와 C_{min} 을 구하는 과정이다. 단계 (11)는 오버플로우가 발생되지 않을 최대 전송률과 언더플로우가 발생되지 않을 최소 전송률의 평균을 계산하는 과정이다. $QoS_Check(r)$ 함수는 ts 프레임부터 현재 검색되는 구간의 끝 프레임까지를 전송률 r 로 전송하려는 경우에 QoS를 만족하는지 여부를 판단하는 함수이다. $QoS_Check(r)$ 함수를 수행하여 이 함수의 반환 값이 참인 경우에는 단

계 (12)에서와 같이 새로운 구간을 검색하고 거짓 값을 반환하는 경우에는 단계 (13)부터 단계 (14)에서와 같이 런의 시작 프레임부터 QoS가 만족되는 구간의 마지막 프레임 i 까지의 전송률을 오버플로우가 발생되지 않을 최대 전송률과 언더플로우가 발생되지 않을 최소 전송률의 평균으로 설정한다. (단계 15)에서는 프레임을 전송하는 과정으로, $output(ts \sim i, r)$ 함수는 프레임 ts 부터 프레임 i 까지에 속한 프레임들을 전송률 r 로 전송하는 함수이다. (단계 16)에서는 새로운 전송률이 적용되는 구간의 첫 프레임을 설정한다.

<표 1> 제안 알고리즘

```

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



(그림 6) 제안 알고리즘에 의한 (그림 5)의 전송률 조절 방법의 예

(그림 6)은 제안 알고리즘에 의한 (그림 5)의 전송률 조절 과정을 보여준다. 첫 번째 구간부터 4번째 구간까지 동일한 전송률로 전송하여서 e-PCRTT 알고리즘보다 전송률 변화 횟수가 적다.

4. 실험 결과

C 언어로 제안 알고리즘과 e-PCRTT 알고리즘을 구현하였다. <표 2>는 MPEG-2로 저장된 비디오 소스들의 파라미터이다[9]. Length는 비디오 재생 시간, Ave Frame Size는 비디오 스트림의 각 프레임을 구성하는 바이트 수의 평균, Std.Dev는 프레임을 구성하는 바이트 수의 표준편차로서 이 값이 클수록 프레임을 구성하는 바이트 수의 변화가 심하다. 1993 Final Four는 프레임 당 바이트 수의 변화가 가장 심한 비디오 소스이고 Seminar는 프레임 당 바이트 수의 변화가 아주 적은 비디오 소스이다.

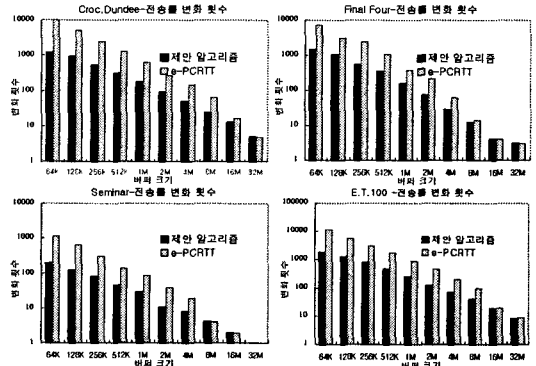
<표 2> MPEG 비디오 소스의 파라미터들

Video Clip Name	Length (min)	Ave Frame Size (KB)	Std.Dev (KB)
Crocodile Dundee	94	10.52	2.28
1993 Final Four	41	16.07	4.041
Seminar	63	8.4	0.58
E.T.100	100	15.38	3.22

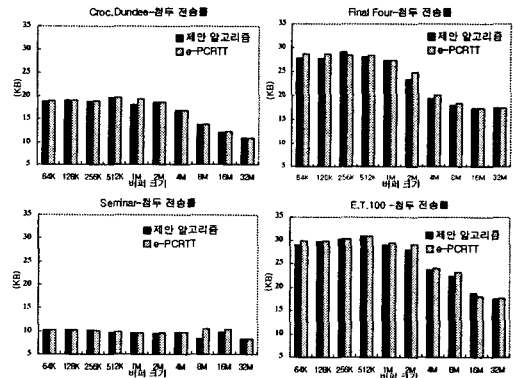
전송률 변화 횟수의 제한이 있고 구간의 크기가 일정한 e-PCRTT 알고리즘과 제안 알고리즘의 성능 평가를 위해 전송률 변화 횟수, 첨두 전송률, 전송률 변화량, 첨두 전송률 이용률을 비교하였다. e-PCRTT에서 구간 크기는 언더플로우 또는 오버플로우가 발생되지 않으면서 구성할 수 있는 구간 크기의 최대 값으로 설정하였다. 클라이언트 버퍼 크기는 64K부터 32M로 다양하게 설정하였다[7].

(그림 7)은 각 버퍼 크기에 대해 전송률 변화 횟수를 비교한 결과이다. 전송률 변화 횟수가 적을수록 서버와 네트워크간의 전송률 변화로 인한 통신 오버헤드가 적다. 제안 알고리즘의 전송률 변화 횟수가 e-PCRTT 알고리즘보다 적다. 두 알고리즘 모두에서 버퍼 크기가 커질수록 1개의 전송률로 많은 프레임을 전송할 수 있어서

전송률 변화 횟수가 적어진다.



(그림 7) 전송률 변화 횟수 비교



(그림 8) 첨두 전송률 비교

(그림 8)은 첨두 전송률 비교이다. 이 값이 낮을수록 서버와 클라이언트 간의 네트워크 자원들이 다수의 비디오 스트림을 서비스 할 수 있다[7]. Croc.Dundee에서 버퍼 크기가 64K인 경우와 같이 일부 경우에서 제안 알고리즘의 첨두 전송률이 e-PCRTT 알고리즘에서보다 높다. 이는 동일한 전송률로 전송하는 구간들간의 바이트 수의 차이가 e-PCRTT 알고리즘의 구간간의 바이트 수의 차이보다 컸기 때문이다. 두 알고리즘 모두에서 첨두 전송률은 버퍼 크기가 클수록 바이트 수가 큰 프레임들로 구성되는 런의 전송률을 완만하게 설정할 수 있어서 대체적으로 작아진다. 그러나, 일부 버퍼 크기에서 제안 알고리즘과 e-PCRTT 알고리즘의 첨두 전송률은 버퍼 크기가 커져도 감소되지 않는 경우가 있다. 이는 두 알고리즘 모두 구간을 구성하

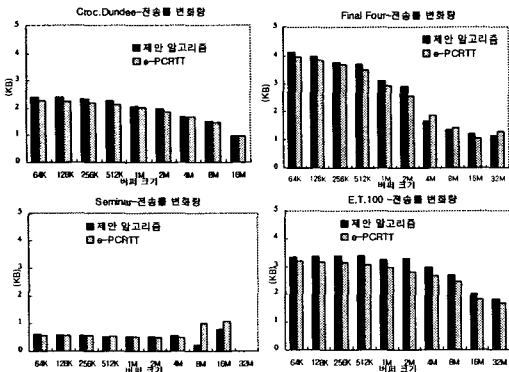
는 프레임들의 바이트 수에 따라서 구간의 전송률이 결정되기 때문이다.

(그림 9)는 각 버퍼 크기에 대해 전송률 변화량을 비교한 결과이다. 제안 알고리즘의 전송률 변화량은 바이트 수의 표준 편차가 큰 Final Four와 Seminar 비디오에서 버퍼 크기가 큰 경우를 제외하고 e-PCRTT 알고리즘보다 크다. 이는 동일한 전송률로 전송할 수 있는 구간간의 바이트 수의 차이가 e-PCRTT 알고리즘에서보다 대부분 커질 수 있기 때문이다.

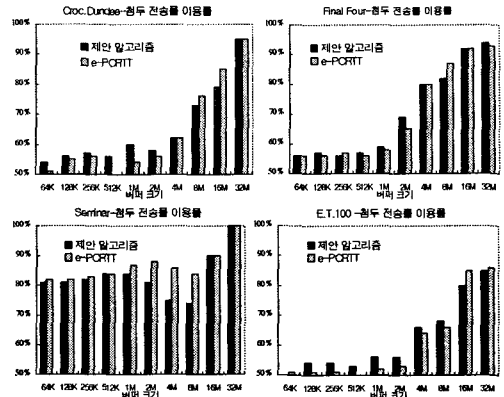
(그림 10)은 각 버퍼 크기에 대해 첨두 전송률의 이용률 비교이며 식(3)[4]에 의해 계산되고, 이 값이 높을수록 첨두 전송률의 낭비가 없음을 의미한다. 여기에서 n은 전송률 변화 횟수, c_i 는 i번째 런의 전송률이다.

$$\frac{\sum_{i=0}^{n-1} (c_i)}{\max\{c_i\} \times n} \dots\dots (3)$$

제안 알고리즘의 첨두 전송률 이용률은 Croc, Dundee에서 버퍼 크기가 8M, E.T100에서 버퍼 크기가 8M인 경우에서와 같이 일부 경우에서 e-PCRTT 알고리즘보다 높다. 이 경우들을 제외하면 제안 알고리즘의 첨두 전송률 이용률은 e-PCRTT보다 대부분 높은데, 이는 (그림 8)에서와 같이 제안 알고리즘의 첨두 전송률이 e-PCRTT 알고리즘에서보다 작기 때문이다.



(그림 9) 전송률 변화량 비교



(그림 10) 첨두 전송률 이용률 비교

5. 결론 및 추후 연구 방향

버퍼 크기가 커질 수 있는 PCRTT 알고리즘을 개선한 e-PCRTT 휴리스틱 방법에서는 PCRTT 알고리즘에서와 같이 전송률 변화 횟수에 대한 제한이 있고 런의 크기가 일정하다. 그러므로 전송률 변화 횟수가 큰 경우에는 런의 크기가 작아져서 연속된 런에서 전송률의 변화가 필요하지 않을 경우에도 불필요하게 변화시키는 경우가 발생할 수 있다.

본 논문에서는 e-PCRTT 알고리즘의 단점을 개선하기 위하여 현재 구간에서 새로운 구간까지 QoS를 유지하면서 전송할 수 있는 전송률들 중에서 최소 전송률과 최대 전송률의 평균을 계산하여 이 전송률로 현재 구간과 새로운 구간에 속한 프레임들을 전송하는 경우 QoS가 보장되는지 여부를 점검한다. QoS가 보장되는 경우에는 새로운 연속적인 구간을 검색하고 QoS가 보장되지 않을 경우에는 현재 구간까지를 동일한 전송률로 전송할 수 있는 마지막 구간으로 설정하는 스무딩 알고리즘을 제안하였다. 그리고, 제안 알고리즘의 성능 평가를 위해 e-PCRTT 알고리즘들과 전송률 변화 횟수, 첨두 전송률, 전송률 변화량, 첨두 전송률 이용률을 비교하였다. 그 결과, 제안 알고리즘은 e-PCRTT 알고리즘들보다 첨두 전송률은 낮고 전송률 변화 횟수 비교에서는 우수한 결과를 보였으나 전송률 변화량과 첨두 전송률 이용률 비교에서 일부 버퍼 크기에서 좋지 않은 결과를 보였다. 이는 제안

알고리즘이 구간간의 바이트 수의 차이를 고려하지 않고 전송률 변화 횟수를 줄이는 것만을 고려하여 전송 계획을 세우기 때문이다. 따라서, 제안 알고리즘은 구간의 크기가 일정하고 전송률 변화 횟수와 침투 전송률을 줄이면서 비디오 데이터를 전송하려는 경우에 적합하다.

추후에는 다양한 비디오 데이터와 평가 요소를 이용하여 제안 알고리즘을 비교 분석할 예정이다.

참고문헌

- [1] D. Le Gall, "MPEG: A video compression standard for multimedia applications", Communications of the ACM, vol.34, pp.47-58, April 1991.
- [2] W. Feng, F. Jahanian, S. Sechrest, "An Optimal Bandwidth Allocation Strategy for the Delivery of Compressed Preencoded Video", ACM/ Springer Multimedia Systems, vol. 5, no. 5, pp. 297-309, Sept 1997.
- [3] 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.
- [4] W. Feng, S. Sechrest, "Critical Bandwidth Allocation for the Delivery of Compressed Preencoded Video", Computer Communications, Vol 18, No. 10, pp. 709-717, Oct. 1995.
- [5] J. D. Salehi, "Scheduling network processing on multimedia and multiprocessor servers", Ph.D. dissertation, Univ. Massachusetts, Amherst, Sept. 1996.
- [6] J. M. McManus et. al., "Video-on-Demand Over ATM: Constant-Rate Transmission and Transport", in IEEE Journal on selected areas in comm., Vol. 14 No. 6, August 1996.
- [7] W. Feng and J. Rexford, "Performance evaluation of smoothing algorithms for transmitting prerecorded VBR video," IEEE Trans. on Multimedia, September 1999.
- [8] 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.
- [9] <http://www.cis.ohio-state.edu/~wuchi>.

임 재 환



1991년 숭실대학교
전자공학과 (학사)
1993년 숭실대학교
전자공학과(석사)
1998년 숭실대학교
전자공학과(박사)

1999년 - 현 재: 남서울대학교 멀티미디어학과교수
관심분야: 멀티미디어 통신, 초고속 정보통신망, 인터넷 프로그래밍

방 기 천



1981년 : 서울대학교
전자공학과(학사)
1988년 : 성균관대학교
정보처리학과(석사)
1996년 : 성균관대학교
전산통계학전공(박사)

1984년 - 1995년: MBC 기술연구소
1995년 - 현 재: 남서울대학교 멀티미디어학과 교수
관심분야 : 멀티미디어콘텐츠, 멀티미디어 응용, 인터넷 방송 등