제한된 네트워크 대역폭을 갖는 VOD 서버를 위한 효율적인 스무딩 알고리즘

이면재[0] 박도순
홍익대학교 컴퓨터 공학과
{leemj[0], dspark}@cs.hongik.ac.kr

An Efficient Smoothing Algorithm for VOD Server With Limited Network Bandwidth

Myoun-Jae Lee[0] Do-Soon Park
Dept. of Computer Engineering, Hongik University

요 약

스무딩은 가변 비트율로 저장된 비디오 데이터를 클라이언트로 전송할 때 일련의 고정 비트율로 전송할 수 있도록 전송 계획을 세우는 것이다. 이러한 스무딩 알고리즘들에는 CBA, MCBA, MVBA, PCRTT, e-PCRTT 알고리즘 등이 있다. 그러나 이 알고리즘들에서는 전송률 감소량의 최대화를 고려하지 않아서 새로운 구간에서 전송률의 감소가 요구되는 경우에 서버에서 제공할 수 있는 여분의 대역폭에 대한 효율적인 사용을 어렵게 할 수 있다.

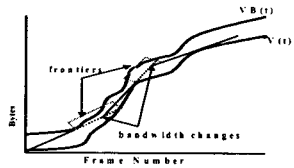
따라서, 본 논문에서는 제한된 네트워크 대역폭을 갖는 VOD 서버에서 보다 많은 비디오 데이터를 서비스할 수 있게 하기 위하여 새로운 구간의 전송률의 감소가 요구되는 경우에는 전송률 감소량을 최대로 하고 새로운 구간의 전송률의 증가가 요구되는 경우에는 전송률 증가량을 최소로 하는 스무딩 알고리즘을 제안하고 성능을 평가한다. 제안 알고리즘의 성능은 Final Four 비디오 소스를 이용하여 첨두 전송률, 전송률 변화 횟수, 평균 전송률 증가량, 평균 전송률 감소량을 비교 분석하였다.

1. 서 론

비디오 데이터는 저장 공간과 효율적인 네트워크 전송을 위하여 압축되어 저장된다. 이러한 압축 방식에는 프레임을 구성하는 비트 수가 동일한 가변 비트율(Variable Bit Rate) 방법과 동일하지 않는 고정 비트율(Constant Bit Rate) 방법이 있다[1]. 특히, 가변 비트율 방법으로 저장된 비디오 데이터는 프레임을 구성하는 비트 수의 차이가 심하므로 이것을 그대로 전송한다면 QoS(Quality of Service)를 보장하기 위해서 전송률을 급격히 증가시켜야 되는 버스트(Burst) 현상이 발생될 수 있다[1,2,3]. 스무딩 기법은 이러한 버스트 현상을 막기 위해 가변 비트율 방법으로 저장된 비디오 데이터를 일련의 고정 비트율로 전송할 수 있도록 비디오 데이터에 대한 전송 계획을 세우는 것이다[1,2,3].

[그림 1]은 스무딩 기법의 원리를 나타내는데[7], X축은 시간 즉 프레임 번호이며 Y축은 바이트 수이다. 식(1)은 언더플로우 경계선을 나타내며 f_i는 i번째 프레임의 바이트 수이다. 식(2)는 오버플로우 경계선으로 언더플로우 경계선에서 클라이언트의 버퍼 크기(b)를 더한 값이다. 스무딩 알고리즘은 언더플로우 경계선과 오버플로우 경계선 사이의 영역에서 동일한 전송률로 전송할 수 있는 연속적인 프레임들을 검색해야 하며, 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 스무딩을 위한 알고리즘에는 전송률 증가 횟수의 최소화가 목표인 CBA(Critical Bandwidth Allocation)[4], 전송률 변화 횟수의 최소화가 목표인 MCBA(Minimum Changes Bandwidth Allocation)[4], 전송률 변화량의 최소화가 목표인 MVBA(Minimum Variability Bandwidth Allocation)[5], 일정한 구간 크기에서 요구되는 버퍼 크기를 최소화하는 것을 목표로 하는 PCRTT (Piece wise Constant Rate Transmission and Transport)[6,7], 버퍼 크기가 커질 수 있는 PCRTT 알고리즘을 개선한 e-PCRTT(enhanced-PCRTT) 알고리즘[8] 등이 있다. 그러나 이 알고리즘들에서는 새로운 구간의 전송률의 감소가 요구되는 경우에 전송률 감소량의 최대화를 고려하지 않아서 해제되는 전송률의 크기가 작을 수 있다. 그래서 VOD 서버에서 보다 많은 비디오 데이터를 서비스하는 것을 어렵게 할 수 있다.



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

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

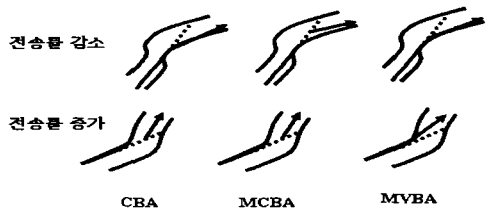
[그림 1] 스무딩 기법의 원리

따라서, 본 논문에서는 새로운 구간의 전송률의 감소(증가)가 요구되는 경우에는 감소량(증가량)을 최대(최소)로 하는 프레임을 검색하여 이 프레임까지를 런(Run)으로 설정하는 스무딩 알고리즘을 제안한다. 제안 알고리즘과 다른 스무딩 알고리즘들과의 성능 평가를 위해 1개의 비디오 데이터를 사용하여 첨두 전송률, 전송률 변화 횟수, 전송률 변화량, 평균 전송률 증가량, 평균 전송률

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

2. 관련 연구

스무딩 기법은 가변 비트율로 저장된 비디오 스트림을 전송할 때 버스트를 줄이기 위해 일련의 고정 비트율로 전송하는 방법으로, 연장 구간에 속하는 프레임 중에서 다음 런의 시작 프레임을 선택하는 방법은 알고리즘의 목적에 따라 다르다. [그림 2]는 CBA, MCBA, MVBA 알고리즘의 전송률 조절 방법을 나타낸다.



[그림 2] CBA, MCBA, MVBA 전송률 조절 방법

CBA 알고리즘[4]에서는 현재 전송률에 의해 오버플로우가 발생된다면 연장 구간의 첫 프레임을 다음 런의 시작 프레임으로 설정하며, 현재 전송률에 의해 언더플로우가 발생되는 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[2]의 목적은 전송률 변화 횟수의 최소화인데, 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. MVBA 알고리즘[3]은 전송률 변화량을 최소화하는 것이 목적이며, 다음 런의 전송률을 증가 또는 감소시켜야 할 때에 연장 구간의 첫 프레임을 다음 런의 시작 프레임으로 설정하는데 전송률 변화 횟수가 상대적으로 많다[7]. 전송률 변화 횟수가 주어지는 PCRTT 알고리즘[6]은 일정한 런의 크기를 갖으면서 요구되는 클라이언트 버퍼 크기를 최소화하는 것이 목적인데 모든 구간에 동일한 오프셋 값을 더하므로 요구되는 버퍼 크기가 커질 수 있다[8]. 이 문제점을 개선한 e-PCRTT 알고리즘은 각 구간의 언더플로우가 발생되지 않을 최소 전송률과 오버플로우가 발생되지 않을 최대 전송률의 평균을 해당 구간의 전송률로 설정한다. 그러나, e-PCRTT 알고리즘에서는 버퍼 크기가 작은 경우에 QoS를 만족시키지 못하는 구간이 발생될 수 있으며 강제적으로 구간마다 전송률을 변화 시켜야 되므로 불필요하게 전송률이 변화 될 수 있다[9]. 그러나, 이 알고리즘들에서는 전송률 감소량의 최대화를 고려하지 않아서 새로운 구간의 전송률 감소가 요구되는 경우에 감소되는 전송률의 크기가 작을 수 있다. 그래서, 제한된 네트워크 대역폭을 갖는 VOD 서버에서 보다 많은 비디오 데이터를 서비스 하는 것이 어려울 수 있다.

3. 제안 알고리즘

기존 스무딩 알고리즘들에서는 새로운 구간의 전송률의 감소가 요구되는 경우에 전송률 감소량의 최소화를 고려하지 않고 전송률을 설정하여서 서버에서 제공할 수 있는 여분의 대역폭의 크기가 작아질 수 있다. 그래서 제한된 네트워크 대역폭을 갖고 있는 서버에서 보다 많은 비디오 데이터에 대한 서비스가 어려워질 수 있다.

따라서, 본 논문에서는 새로운 런의 전송률의 감소가 요구되는 경우에는 연장 구간에서 가장 크게 전송률이 감소될 수 있는 프레임을 검색하여 이 프레임을 새로운 런의 시작 프레임으로 설정하고 새로운 런의 전송률의 증가가 요구되는 경우에는 연장 구간의 첫 프레임을 새로운 구간의 시작 프레임으로 설정하는 스무딩 알고리즘을 제안한다. 그래서, 제한적인 네트워크 대역폭을 갖는 VOD 서버에서 보다 많은 비디오 데이터를 서비스 할 수 있게 한다.

이러한 원리를 사용한 알고리즘이 [표 1]인데, n은 비디오 스트림을 구성하는 프레임 개수이고 t는 검색되는 프레임 번호이다. 그리고 Cmax는 QoS를 보장하는 전송률 중에서 가장 큰 전송률이고 Cmin은 최소 전송률이다. 단계(6)의 oc는 런의 시작 프레임부터 프레임 t의 오버플로우 경계점까지의 평균 전송률이고 단계 (7)의 uc는 런의 시작 프레임부터 프레임 t의 언더플로우 경계점까지의 평균 전송률이다. q는 버퍼에 채워져 있는 바이트 수이고 ts는 새로운 런의 시작 프레임이다. 단계 (8)에서 단계 (12)까지는 언더플로우가 발생되어 새로운 런에서 전송률 증가가 요구되는 경우에 연장 구간의 첫 프레임을 새로운 런의 시작 프레임으로 설정하는 과정이다. 단계 (13)에서 단계 (16)까지에서는 오버플로우가 발생되어 새로운 런에서 전송률의 감소가 요구되는 경우에 연장 구간에 있는 프레임들 중에서 전송률이 가장 크게 감소될 수 있는 프레임을 검색하여 이 프레임을 새로운 런의 시작 프레임으로 설정하는 과정이다.

```
proposed algorithm()
{
(1) t=ts+1
(2) compute Cmax, Cmin
(3) n=number of last frame
(4) Repeat
(5)   t=t+1
(6)   oc=(VB[t]-(V[ts]+q)/(t-ts)
(7)   uc=(V[t]-(V[ts]+q)/(t-ts)
(8)   if  (Cmax < uc) {
(9)     target=find a starting frame of frontier
(10)    output(ts~target-1, Cmax)
(11)    ts=target
(12)  }
(13)  else if (Cmin>oc){
(14)     target=find a frame the largest decreasing
        transmission rate
(15)    output(ts~target-1,Cmin);
(16)    ts=target
(17)  }else ;
(18)  compute Cmin,Cmax,q
(19) UNTIL t==n
}
```

[표 1] 제안 알고리즘

4. 실험 결과

C 언어로 스무딩 알고리즘을 구현하여 실험하였으며, [표 2]는 사용된 Final Four 비디오 소스에 대한 파라미터[7,10]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. 제안 알고리즘의 성능 평가를 위해 첨두 전송률[7], 전송률 변화 횟수[7], 평균 전송률 증가량, 평균 전송률 감소량을 비교하였다. 평균 전송률 증가량(감소량)은 모든 런의 전송률들 중에서 이전 런의 전송률에 비해 증가(감소)되는 전송률의 크기를 합하여 이 값을 전송률이 증가(감소)되는 횟수로 나눈 몫이다. 평균 전송률 감소량(증가

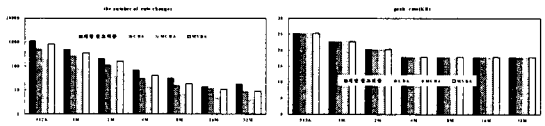
량)은 클(작을)수록 서버에서는 보다 많은 클라이언트들에게 비디오 데이터를 서비스 할 수 있다.

Video Clip Name	Length (min)	Ave Frame Size	Max Frame Size	Min Frame Size	Total Size	Std Dev
Final Four	41	16.07	28.872	2.504	1,178,013	16.57

[표 2] Final Four 비디오 소스들의 파라미터들

[그림 3]은 첨두 전송률 비교이다. 이 값이 낮을수록 서버와 클라이언트 간의 네트워크 자원들이 다수의 비디오 스트림을 서비스 할 수 있다[7]. 실험에 사용된 모든 알고리즘의 첨두 전송률이 동일한데, 이들 알고리즘들이 모두 첨두 전송률을 최적화하는 것을 기본적인 목적으로 하여 전송 계획을 세우기 때문이다.

[그림 4]는 전송률 변화 횟수 비교이다. 제안 알고리즘이 가장 많다. 이는 CBA 알고리즘과 MCBA 알고리즘에서는 각각 전송률의 증가 횟수와 전송률 변화 횟수를 최소화하기 때문이다.

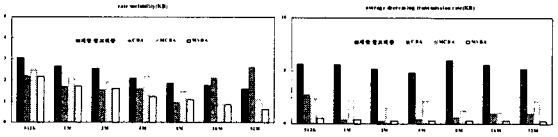


[그림 3] 첨두 전송률 비교 [그림 4] 전송률 변화 횟수 비교

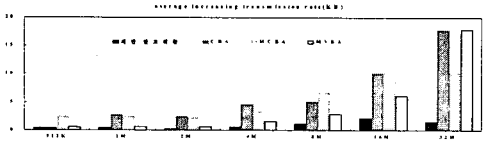
[그림 5]는 전송률 변화량 비교이다. 제안 알고리즘의 전송률 변화량이 큰 편에 속하는데 이는 새로운 런의 전송률이 증가되는 경우에는 전송률 증가량을 적게 하고 새로운 런의 전송률이 감소되는 경우에는 감소량을 크게 하기 때문이다. 전송률 변화량의 최소화가 목적인 MVBA가 가장 적다.

[그림 6]은 평균 전송률 감소량이다. 새로운 런의 전송률의 감소되는 경우에 전송률의 감소량의 차이를 가장 크게 갖는 프레임을 검색하여 이 프레임을 새로운 런의 시작 프레임으로 설정하는 제안 알고리즘이 가장 크다. 전송률 변화량의 최소화가 목적인 MVBA가 가장 적다.

[그림 7]은 평균 전송률 증가량이다. 제안 알고리즘이 가장 작아 우수한 결과를 보인다. 특히, MVBA 알고리즘보다 평균 전송률 증가량이 적은데 이는 제안 알고리즘의 전송률 증가 횟수가 MVBA보다 훨씬 크기 때문이다.



[그림 5] 전송률 변화량 비교 [그림 6] 평균 전송률 감소량 비교



[그림 7] 평균 전송률 증가량 비교

5. 결론 및 추후 연구 방향

기존 스무딩 알고리즘들에서는 전송률의 감소가 요구되는 경우에 감소량을 최대로 하는 것을 고려하지 않았다. 그래서, 전송률이 감소되는 경우에 제한된 네트워크 대역폭을 갖는 VOD 서버에서 보다 많은 비디오 데이터를 클라이언트들에게서 서비스 하는 것이 어려울 수 있다. 따라서, 본 논문에서는 제한된 네트워크 대역폭을 갖고 있는 VOD 서버에서 보다 많은 비디오 데이터를 서비스하기 위해 새로운 구간에서 전송률의 감소가 요구되는 경우에는 연장 구간에서 전송률 감소량이 가장 큰 프레임을 새로운 구간의 시작 프레임으로 설정하고 새로운 구간에서 전송률의 증가가 요구되는 경우에는 연장 구간의 첫 프레임을 새로운 구간의 시작 프레임으로 설정하는 스무딩 알고리즘을 제안하고 다른 스무딩 알고리즘들과 전송률 변화 횟수, 첨두 전송률, 전송률 변화량, 평균 전송률 감소량, 평균 전송률 증가량 등의 평가 요소로 비교 분석하였다. 그 결과, 제안 알고리즘은 다른 스무딩 알고리즘들과 첨두 전송률은 동일하고 평균 전송률 증가량, 평균 전송률 감소량은 우수한 결과를 보이고 전송률 변화량과 전송률 변화 횟수는 다른 알고리즘들에 비해 큰 결과를 보였다. 이는 새로운 구간의 전송률의 증가가 요구되는 경우에는 증가량이 가장 적은 프레임을 검색하고 새로운 구간의 전송률의 감소가 요구되는 경우에는 감소량이 가장 큰 프레임을 검색하여 검색된 프레임을 새로운 런의 시작 프레임으로 설정하기 때문이다. 그 결과, VOD 서버의 전송 계획으로 제안 스무딩 알고리즘을 사용하는 경우에는 다른 스무딩 알고리즘들에서보다 많은 비디오 데이터를 서비스 할 수 있다.

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

참고 문헌

[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 Prerecoded Video", ACM/Springer-Verlag Multimedia Systems, vol. 5, no. 5, pp. 297-309, Sept 1997.

[3] J. D. Salehi, et. al. "Supporting stored video: Reducing rate va riability and end-to-end resource requirements through optimal smo othing", in Proc. of ACM SIGMETRICS, pp. 222-231, May 1996.

[4] W. Feng, S. Sechrest, "Critical Bandwidth Allocation for the Delivery of Compressed Prerecorded Video", Computer Communic ations, Vol 18, No. 10, pp. 709-717, Oct. 1995.

[5] J. D. Salehi, "Scheduling network processing on multimedia an d multiprocessor servers", Ph.D. dissertation, Univ. Massachusetts, Amherst, Sept. 1996.

[6] J. M. McManus et. al.,"Video-on-Demand Over ATM: Constan t-Rate Transmission and Transport", in IEEE Journal on selected a reas in comm., Vol. 14 No. 6, August 1996.

[7] W. Feng and J. Rexford, "Performance evaluation of smoothin g algorithms for transmitting prerecorded VBR video," IEEE Tran s. 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] 이면재, 이준용, 박도순,"가변 비트율 비디오 전송을 위한 효율적인 스무딩 알고리즘",한국정보처리학회논문지 C, 2004.12.

[10] http://www.cis.ohio-state.edu/~wuchi.