

주문형 비디오 서비스를 위한 효율적인 대역폭 스무딩 알고리즘

곽준원⁰ 이면재 송하운 박도순

홍익대학교 컴퓨터공학과

{jkwak⁰, mjlee, song, dspark}@cs.hongik.ac.kr

An Efficient Bandwidth Smoothing Algorithm for VoD Service

Joon-Won Kwak⁰, Myoun-Jae Lee, Ha Yoon Song, Do-Soon Park

College of Information and Computer Engineering, Hongik University

요약

주문형 비디오 서비스(VoD: Video on Demand)의 경우 네트워크의 자원 이용도와 QoS(Quality of Service)가 보장 되도록 구축되어야 한다. 대역폭 스무딩 알고리즘은 주문형 비디오 시스템에서 비디오 데이터를 보낼 경우 비디오의 데이터의 급격한 대역폭의 증가 또는 감소를 최소한 줄이는 기법이다. 본 논문에서는 기존의 대역폭 스무딩 기법과 비디오 압축 기법의 특성을 이용하여 새로운 모델과 스무딩 알고리즘을 제안한다. 제안된 알고리즘의 성능평가를 위해 기존 스무딩 알고리즘의 결과와 비교 분석한다.

1. 서론

최근에 사용자가 원하는 멀티미디어 정보를 빠르고 쉽게 얻을 수 있는 주문형 멀티미디어 정보(Multimedia information on-demand) 서비스가 시작되고 있다. 이러한 서비스를 수행하기 위해서는 멀티미디어 데이터의 저장 및 전송 기능을 담당하는 멀티미디어 서버(Multimedia server)가 필요하다. 현실에 나타나는 멀티미디어 서버는 주문형 비디오 서버가 있다[1].

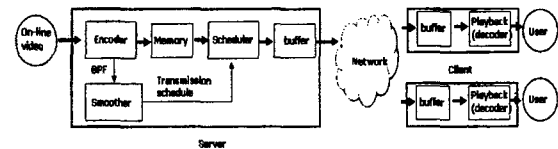
주문형 비디오 서버를 포함한 대부분의 응용분야에서는 저장 공간과 네트워크의 대역폭을 줄이기 위해서 비디오 압축기술을 사용한다. 이 비디오 압축기술은 크게 고정 비트율(CBR: Constant Bit Rate)과 가변 비트율(VBR: Variable Bit Rate)로 나뉜다. 고정 비트율은 비디오 압축 후에 프레임당 비트수가 일정하지만 가변 비트율은 비디오 압축 후에 프레임당 비트수가 서로 다르다. 가변 비트율의 압축률이 고정 비트율의 압축률보다 우수하지만, 가변 비트율로 저장된 비디오를 전송할 경우에는 버스트(Burst)가 생긴다. 특히, 제한된 네트워크 자원 내에서 고품질의 비디오를 전송할 때 버스트를 줄이지 않으면 클라이언트는 끊김이 있는 비디오를 시청하게 된다. 이에 대역폭 스무딩 알고리즘은 가변 비디오를 전송할 경우 클라이언트의 버퍼에서 언더플로우 또는 오버플로우가 발생되지 않을 범위 내에서 버스트를 해결하기 위해 전송물의 급격한 증가나 감소를 최소한 줄이는 기법이다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구를 소개하고, 3장에서는 제안한 스무딩 알고리즘을 기술하고, 4장에서는 실험 결과 그리고 5장은 결론 및 향후 연구 과제를 기술한다.

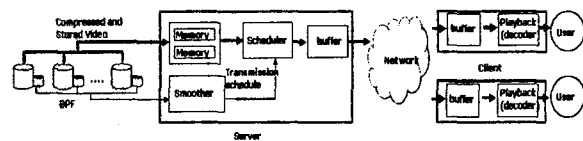
2. 관련연구

2.1 주문형 비디오 서버

주문형 비디오 서버는 클라이언트가 요청하는 비디오의 특성에 따라 크게 두 가지로 분류된다. 비디오를 실시간(On-line Video)으로 전송하기 위한 주문형 비디오 서버의 모델은 [그림 1]과 같다[2,3]. 실시간 비디오 스트림은 부호화기(Encoder)를 통해 압축된 가변 비트율의 형태로 비디오 스트림이 압축된다. 이 비디오 스트림은 서버측 메모리에 잠시 저장된다. 이때 스무더(Smoother)는 부호화기에서 얻은 프레임당 비트수(BPF: Bits Per Frame)와 클라이언트로부터 버퍼 크기의 정보들을 얻은 후 전송 스케줄을 계산한다. 즉, 전송 스케줄은 전체 비디오 스트림에 대한 프레임의 전송률들로 구성되며 스케줄러(Scheduler)는 서버 메모리에 잠시 저장된 비디오 스트림을 이 전송 스케줄에 따라 일련의 일정한 전송률로 서버



[그림 1] 실시간 주문형 비디오 서버 모델



[그림 2] 저장된 주문형 비디오 서버 모델

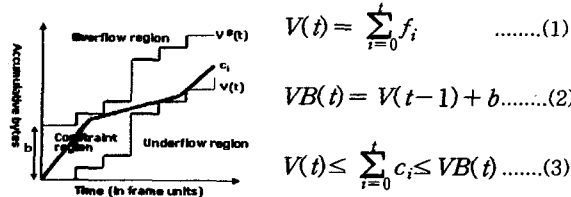
버퍼에게 전달한다. 서버 버퍼는 네트워크의 환경에 따라 비디오 스트림을 전송할 수 있는 역할을 담당한다.

주문형 비디오 서버의 모델에서 네트워크 환경은 일반적으로 네트워크 상태가 안정된 상태로 간주된다. 실시간 주문형 비디오 서버에서 중요한 점은 실시간으로 비디오 스트림을 여러 클라이언트에게 실시간으로 전송하기 때문에 부호화기와 스무더에게 동기화를 요구되는 점이다.

[그림 2]는 본 논문에서 제안하는 비디오(Stored Video)를 위한 주문형 비디오 서버의 모델을 보인다. 기존 모델[4]과의 차이점은 각 비디오 소스에 대한 프레임당 비트수의 정보를 저장하는 공간이 있다는 점이다. 즉, 비디오 소스는 먼저 비디오 압축을 시킨 후 비디오 저장 장소에 저장된다. 또한 압축 중에 생성되는 프레임당 비트수의 정보를 비디오 저장 장소 옆에 저장한다. 저장된 주문형 비디오 서버의 차이점은 서버에서 전송하는 비디오 소스와 서버 메모리의 수의 차이이다. 실시간 비디오 서버는 생방송 인터넷의 경우에 하나의 비디오 소스로 전송하지만 저장된 비디오 서버는 압축된 여러 비디오 소스가 저장되어 있고 각 비디오 소스에 해당하는 프레임당 비트수를 함께 저장된다. 이 정보들은 전송하기 전에 스무더를 통하여 전송 스케줄을 계산하는데 이용된다. 또한 실시간 비디오 서버는 비디오 소스가 하나이기 때문에 하나의 메모리만 필요하지만 저장된 비디오 서버는 두 가지 이상의 비디오를 동시에 전송하기 때문에 클라이언트의 수에 만큼 해당 메모리가 필요하다.

2.2 대역폭 스무딩 알고리즘

[그림 3]은 전송 스케줄 과정과 전송 스케줄에 사용된 수식이다[5]. 식 (1)의 $V(t)$ 는 서버에서 클라이언트로 비디오 스트림이 최소한 정지되지 않도록 보낼 수 있는 언더플로우 경계선이다. 식 (2)의 $VB(t)$ 는 클라이언트의 버퍼에서 비디오 스트림의 한 프레임도 분실되지 않도록 보낼 수 있는 오버플로우 경계선이며, b 는 클라이언트의 버퍼 크기이다. 식 (3)은 제한 영역(Constraint region) 즉 $V(t)$ 와 $VB(t)$ 사이 에 있는 영역으로 이 영역내의 전송률은 QoS가 보장되는 전송률을 의미한다. C_i 는 i 번째의 전송률이다.



[그림 3] 전송 스케줄 과정 및 수식

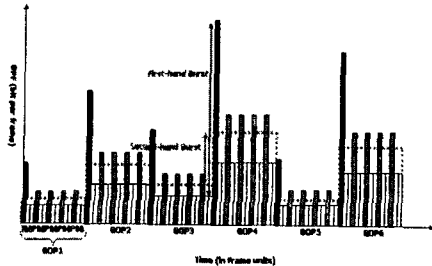
기존 스무딩 알고리즘은 MVBA, RCBS, ON-OFF 그리고 PCRTT 등이 있다[6]. MVBA(Minimum Variability Bandwidth Allocation)은 대역폭의 변화폭을 최소화하는 알고리즘이다[7]. 대역폭의 변경할 경우 변화폭이 심하다라도 완만한 증가 및 감소로 버스트를 최대한 줄일 수 있는 장점이 있다. RCBS(Rate - Constrained Bandwidth Smoothing algorithm)는 MVBA를 이용하여 최대 대역폭 제약조건을 구한다[8,9]. 이 제약조건 내에서 클라이언트 버퍼의 이용으로 버스트를 줄이고 기존의 전송률을 지속시켜서 이 버퍼의 이용률을 최소화하는 알고리즘이다. ON-OFF(On period and Off period bandwidth allocation)는 네트워크의 제약조건으로 최대 대역폭을 주어진 상태에서 ON 일 때 최대 대역폭으로 보내고 OFF일 때 비디오 스트림을 전혀 보내지 않는다[10]. PCRTT (Piecewise Constant Rate Transmission and Transport)는 전송 스케줄을 일정한 시간 간격으로 나눈 다음, 각 간격마다 누적된 비트수 합을 평균을 구한 후, 전송률을 오프셋(Offset)만큼 증가시키는 알고리즘이다[11].

지금까지의 연구된 논문들은 서버측의 CPU 오버헤드를 고려하지 않았다. 이러한 결과로 주문형 비디오 서버 자체가 많은 사용자를 대상으로 하는 비디오 데이터 서비스를 수행하는 데 병목 현상의 주된 원인이 될 수 있다. 제안된 논문에서는 여러 스무딩 알고리즘 중에서 MVBA 알고리즘을 바탕으로 하여 GOP단위로 스무딩을 처리하는 알고리즘을 제시한다.

3. 제안된 MVBAG 알고리즘

본 논문에서는 기존의 스무딩 알고리즘들 중에서 전송률 변화폭이 제일 적은 특성을 가진 MVBA 알고리즘을 기반으로 하여 MPEG[12]에서 프레임의 그룹단위인 GOP(Group of Pictures)로 전송 스케줄을 계산한 새로운 알고리즘인 MVBAG(Minimum Variability Bandwidth Allocation over Group of Pictures)를 제시한다.

[그림 4]는 프레임간의 버스트(First-hand burst)와 GOP간



[그림 4] 프레임과 GOP단위의 버스트 관계

```

PROCEDURE MVBAG ()
Initialize queue TB and T
REPEAT
(1) update t, Cmax = Constraint Bandwidth, Cmin = V(first(T))
(2) IF Cmax < V(t) then
(3) REPEAT
(4) output (Cmax), delete the first element on TB
(5) Cmax = Constraint Bandwidth
(6) UNTIL TB is empty
(7) delete the first element on T
(8) ELSE IF Cmin > VB(t) then
(9) REPEAT
(10) output (Cmin), delete the first element on T
(11) Cmin = V(first(T))
(12) UNTIL T is empty
(13) delete the first element on TB
END IF
(14) search the convex lower bound of VB(t) in TB
(15) update (TB, t)
(16) search the concave upper bound of V(t) in T
(17) update (T, t)
(18) UNTIL t = M
END PROCEDURE
    
```

[표 1] MVBAG 알고리즘

의 버스트(Second-hand burst)를 보여준다. 각 막대 그래프는 프레임당 비트수를 의미하고 점선은 15개의 프레임은 하나의 GOP로 설정한 경우 GOP당 비트수를 의미한다. 프레임 단위들 기준으로 전송 스케줄을 구하면 한 GOP 내에서도 특정 프레임이 다른 프레임보다 비트수가 많은 경우가 생긴다. 특히 I 프레임은 다른 프레임보다 비트수가 심한 차이를 보이므로 큰 버스트가 생긴다. 이러한 프레임간의 버스트를 줄이기 위해서 GOP 단위를 기준으로 전송 스케줄을 구하면 GOP내의 버스트 즉 GOP를 구성하는 프레임간의 일차적 버스트가 감소한다. MVBAG 알고리즘은 이 GOP 내의 버스트를 일차적으로 제거하고 GOP 간의 버스트를 제거하기 위해 GOP단위로 전송률을 조절하는 알고리즘이다.

[표 1]의 (14)~(17)은 최대 전송률과 최소 전송률을 구하는 부분이고 (1)~(13)은 구한 전송률 중에서 QoS 보장하는지를 판단하는 부분이다. (1)에서 t는 t번째 GOP를 의미하며, 모든 큐에서 first()는 첫 번째 원소를 참조한다. V(t)는 언더플로우 경계선이고 VB(t)는 오버플로우 경계선이다. C_{max}은 t번째 GOP까지 QoS를 보장하는 최대 전송률이며 C_{min}은 최소 전송률이다. [13]에서는 이 알고리즘의 자세한 설명을 하고 있다. N은 한 비디오 스트림에서 전체 프레임의 개수이고, M은 한 스트림의 총 GOP의 개수인 경우 시간 복잡도는 MVBAG 알고리즘은 O(M)이지만 MVBAG 알고리즘을 제외한 모든 스무딩 알고리즘은 O(N)이다.

4. 실험 결과

[표 2]는 실험에 사용된 비디오 소스의 정보이다. GOP는 15개 프레임을 포함하며 프레임의 크기는 352X240이다. 사용된 단위는 Kbyte이다. 제안된 알고리즘이 프레임간의 버스트와 GOP간의 버스트에 어떤 영향을 주는지 분석하기 위하여 BPF의 표준편차(Stdv.: standard deviation)와 GOP당 비트수의 표준편차를 구했다. 스타워즈는 다른 비디오 소스보다 BPF의 표준 편차가 GOP간의 표준 편차보다 심하다. 뉴스는 화면의 움

Video Source	# of Frame	Avg. BPF	Max BPF	Min BPF	Frame Stdv.	GOP Stdv.
Jurassic Park	70000	15.3	279.2	0.75	10.35	8.08
Star Wars	120000	32.8	358.8	0.5	39.99	26.94
News	22500	10.3	28.6	2.62	4.41	2.89
Yuppi Girl	212000	15.0	198.6	0.6	14.2	9.27
Opera	180000	17.6	188.3	0.12	16.9	7.93
Galaxy Express	100000	12.0	105.2	0.12	10.0	4.69

[표 2] MPEG 비디오 소스의 파라미터들

직입이 거의 없는 비디오 소스이므로 프레임 표준편차와 GOP 표준편차가 거의 차이가 없다.

[그림 5]는 [표 2]의 비디오 소스 중 프레임당 표준편차가 큰 스타워즈와 거의 차이가 없는 뉴스를 가지고 MVBA와 MVBAG를 비교한 실험결과이다. [그림 5]의 (a)에서 스타워즈는 MVBAG 알고리즘이 전송횟수 변화가 더 적다. 즉 MVBAG 알고리즘을 이용하여 비디오 데이터를 전송하면 전송률 변화에 대한 비용을 줄일 수 있다. (b)는 두 알고리즘의 실행 시간 비교이다. MVBAG는 MVBA보다 적어도 15배 이상 실행시간이 적게 소요된다. MVBA 알고리즘에서는 프레임 단위로 전송 스케줄을 구하지만 MVBAG 알고리즘에서는 15개의 프레임을 합친 GOP 단위로 전송 스케줄을 구하기 때문이다. 이러한 결과는 MVBAG 알고리즘을 이용하면 스무딩을 처리하는 서버의 CPU 오버헤드를 기존 MVBA보다 15배 이상 줄인다는 의미이다. 즉 MVBAG 알고리즘을 이용하여 스무딩을 수행하면 다수의 서비스 요구에 대한 서버의 계산 병목 현상을 상당히 해결할 수 있다. MVBAG 알고리즘은 QoS를 보장하기 위해서 클라이언트의 버퍼 크기가 적어도 256Kbytes 이상 요구되어 클라이언트에서 구현 비용이 높아지는 단점이 있으나 현재의 기술 추세에서 클라이언트의 큰 버퍼는 높은 비용을 요구하지 않는다.

[그림 6]의 (a)는 유라기 공원 그리고 (b)는 스타워즈의 비디오 소스들을 가지고 MVBAG와 MVBA 알고리즘의 전송 스케줄이다. 첨두 전송률(Peak rate)을 서로 비교할 때 두 가지의 그래프에서 첨두 전송률과 전송률의 변화 횟수를 기준으로 MVBAG가 우수하며 비디오 데이터 전송에 필요한 네트워크 비용을 줄일 수 있다.

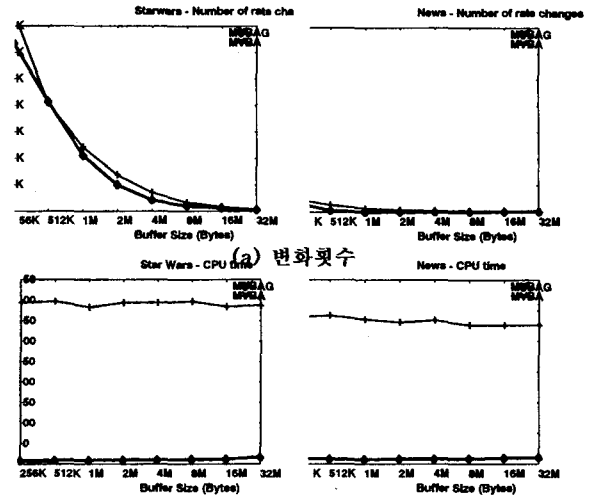
5. 결론 및 향후계획

본 논문에서는 가변 비트율 비디오 스트림에 대한 주문형 비디오 서비스를 제공할 때 서버측의 CPU 오버헤드 및 변화횟수를 줄이는 MVBAG 알고리즘을 제시하였다. 이 알고리즘은 MVBA 알고리즘을 기반으로 MPEG에서 프레임의 그룹단위인 GOP의 특성을 이용하여 개선된 알고리즘이다. 반면에 기존의 스무딩 알고리즘은 프레임 단위로 스무딩을 수행하므로 스무딩 알고리즘의 실행시간이 더 소요된다. 또한 기존 스무딩 알고리즘들은 프레임단위로 스무딩을 수행하므로 주문형 비디오 서버가 여러 클라이언트의 요청을 동시에 서비스를 하는 경우 서버의 과부하가 생길 수 있다. 그러나 MVBAG 알고리즘은 그룹단위 전송 스케줄을 처리하므로 계산 오버헤드를 줄일 수 있다. 또한 MVBAG 알고리즘은 프레임을 구성하는 비트수에 대한 표준편차가 GOP를 구성하는 비트수에 대한 표준편차보다 큰 경우 더 유용하다.

향후에 MVBAG 알고리즘을 보완하여 다양한 요소들로써 기존의 알고리즘과 비교할 예정이다.

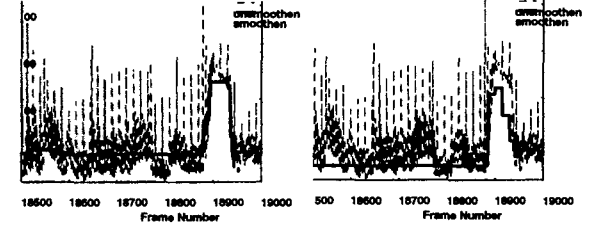
참고 문헌

[1] S. Ramesh, et. al., "Multicast with cache: An Adaptive Zero-Delay VoD Service," in IEEE INFOCOM 2001.
 [2] N. Feamster, "Adaptive Delivery of Real-Time Streaming Video," Ph.D. Univ. Massachusetts, Amherst, May. 2001.
 [3] D. Wu, et. al., "Streaming Video over the Internet: Approaches and Directions," IEEE Trans. on C&S for Video Tech., Feb. 2001.
 [4] J. Kwak, et. al., "An Effective Smoothing Algorithm of VBR Video Stream over Internet", in KISS, vol.1, p.436-438, April 2002.
 [5] J. Zhang and J. Hui "Applying traffic smoothing techniques for QoS control in VBR video transmissions", Rutgers University, Piscataway New Jersey 1998.
 [6] W. Feng and J. Rexford, "Performance evaluation of smoothing algorithms for transmitting prerecorded VBR video," IEEE Trans. on Multimedia, Sep. 1999.
 [7] J. D. Salehi, "Scheduling network processing on multimedia and multiprocessor servers," Ph.D. Univ. Massachusetts, Amherst, Sept. 1996.

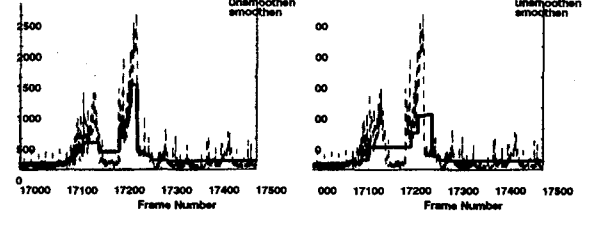


(b) 실행시간

[그림 5] 실험 결과



(a) 유라기 공원



(b) 스타 워즈

[그림 6] 전송 스케줄

[8] W. Feng, "Rate-constrained bandwidth smoothing for the delivery of stored video", in SPIE Multimedia N&C p.58-66, Feb.1997.
 [9] J. Dey, et. al., "Playback restart in interactive streaming video applications," in Proc. IEEE on Multimedia C&S p.458-465, June 1997.
 [10] J. Zhang, et. al., "Traffic characteristics and smoothness criteria in VBR video transmission," in IEEE on Multimedia C&S, June 1997.
 [11] J. McManus et. al., "VoD Over ATM: Constant-Rate Transmission and Transport", in IEEE on comm., Vol.14, No.6, p.1087-1098, Aug. 1996.
 [12] C. Lin, et. al., "MPEG Video Streaming with VCR Functionality," in IEEE Trans. on C&S for Video Tech., Vol.11, No.3 p.415-425, Mar. 2001.
 [13] J. Kwak, et. al, "An Efficient Transmission Rate Control Algorithm for MPEG VOD Service," in KCIES, Vol. 3 No. 8, Aug. 2002.