

스무딩 알고리즘들의 버스트 성능 평가

이면재*

요약

버스트는 가변 비트율로 저장된 비디오 데이터를 전송하려는 경우 급격하게 높은 전송률이 요구되는 것으로 네트워크 자원의 비효율적인 사용의 원인이 될 수 있다. 스무딩은 이를 방지하기 위해 가변 비트율로 저장된 비디오 데이터를 고정 비트율로 변환하는 전송 계획을 세우는 기법으로, CBA, MCBA, MVBA 알고리즘들이 있다.

본 논문에서는 기존 CBA, MCBA, MVBA 알고리즘들의 버스트 감소 정도를 평가하기 위해, 가변 비트율로 저장된 비디오 소스와 스무딩 알고리즘에서의 전송 계획을 버스트에 영향을 미치는 요소들로 비교한다. 사용된 평가 요소는 최대 프레임 바이트 수, 최대 GOP 바이트 수, 프레임당 전송률 변화량, GOP당 전송률 변화량이다.

실험 결과, 실험에 사용된 모든 스무딩 알고리즘들의 버스트 관련 평가 요소들이 특정한 경우를 제외하고 원래 비디오 소스보다 낮았다.

Performance Evaluation focused on Burst of Smoothing Algorithms

Myoun-Jae Lee*

Abstract

The burst is to require abruptly high transmission rate in case of transmitting pre-stored variable bit rate video data, and it causes to be inefficient use of network resource, resource reservation. To avoid these problems, smoothing is transmission plan where variable rate video data is converted to a constant bit rate stream. These smoothing algorithms include CBA, MCBA, MVBA and others.

To evaluate amount of burst reduction in the existing CBA, MCBA, MVBA algorithm, this paper compares the burst-related-factors of transmission plan in smoothing algorithms with original video sources which were stored Variable Bit Rate. There are maximum frame bytes, maximum GOP bytes, transmission rate variability per frame, transmission rate variability per GOP in burst-related evaluation factors.

Experimental result shows burst-related factors of smoothing algorithms which were used for experiment lower than that of pre-stored video data, except special case.

Keywords : Burst, Smoothing Algorithm, QoS, VBR, Transmission Plan

1. 서론

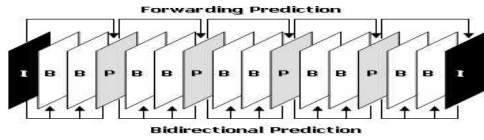
MPEG(Moving Picture Expert Group) 비디오 데이터는 압축 방법에 따라서 프레임당 비트 수가 다른 가변 비트율(Variable Bit Rate)과 프레임당 비트 수가 동일한 고정 비트율(Constant Bit

Rate) 방식으로 나뉘어진다[1]. 고정 비트율 방법은 데이터를 저장하는 것과 네트워크 자원 예약과 사용에 있어서 관리가 편하지만, 저장 공간의 낭비와 자원의 비효율적인 사용을 초래할 수 있다. 그래서, MPEG 비디오 데이터는 가변 비트율 방법으로 저장된다. 가변 비트율 방법은 대역폭 등의 네트워크 자원과 저장 공간을 효율적으로 사용할 수 있는 반면에 특정한 프레임 또는 GOP(Group of Picture)에서 갑자기 높은 전송률이 요구될 수 있다. GOP는 하나의 I 프레임에서 다음 I 프레임 직전 프레임까지의 집합으로 구성되며 임의로 접근 가능한 최소의 독립적인

※ 제일저자(First Author) : 이면재
접수일:2012년 03월 01일, 수정일:2012년 03월 21일
완료일:2012년 03월 27일
* 백석대학교 정보통신학부 멀티미디어전공
davidlee@bu.ac.kr

단위이다[1].

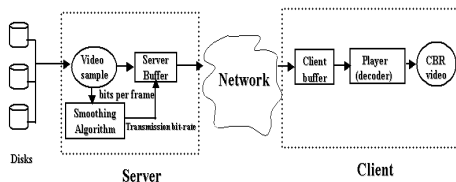
(그림 1)[1]은 MPEG GOP 패턴 예이다. (그림 1)에서 GOP를 구성하는 프레임의 개수는 15이다.



(그림 1) MPEG의 GOP 패턴 예

스무딩은 가변 비트율로 저장된 비디오 데이터를 그대로 전송하려는 경우 발생할 수 있는 버스트를 방지하기 위해서, 서버에서 클라이언트의 버퍼 크기와 저장된 비디오 데이터의 바이트 수에 대한 정보를 바탕으로 전송 계획을 세우는 것을 말한다.

(그림 2)[2,3,4]는 스무딩 기법을 위한 구조를 보여준다. 클라이언트에서 서버에게 비디오 스트림을 요청하는 경우, 서버에서는 클라이언트에서 요청한 비디오 스트림을 검색하여 비디오 스트림을 구성하는 모든 프레임들의 정보, 즉 각 프레임 당 비트 수를 스무딩 알고리즘에게 전달한다. 그러면 스무딩 알고리즘에서는 프레임 당 비트 수와 클라이언트 버퍼 크기를 고려하여 클라이언트 버퍼에서 언더플로우와 오버플로우가 발생되지 않을 전송률, 즉 QoS가 보장되는 전송률을 계산하여 해당 비디오 스트림을 클라이언트에게 전송한다.



(그림 2) 스무딩 기법의 구조

스무딩 기법을 위한 알고리즘에는 CBA(Critical Bandwidth Allocation)[5,6,7], MCBA(Minimum Critical Bandwidth Allocation)[8], MVBA(Minimum Variability Bandwidth Allocation)[9,10,11], PCRTT(Piecewise Constant Rate Transmission and Transport)[12], e-PCRTT(enhanced-PCRTT)[13] 등이 있다.

CBA 알고리즘[2,5,6,7,11]은 전송률 증가 횟수를 최소화하는 전송 계획을 세우며, MCBA 알고리즘[2,8,10,11]과 MVBA 알고리즘[2,9,10,11]에서는 각각 전송률 변화 횟수와 전송률 변화량을 최소화하는 전송 계획을 세운다. PCRTT 알고리즘[12]의 목적은 언더플로우가 발생되지 않으면서 버퍼 크기를 최소화하는 것이며, e-PCRTT 알고리즘[13]의 목적은 버퍼 크기가 커질 수 있는 PCRTT 방법에서의 문제점을 개선하는 것이다. [3-4]에서 제시된 스무딩 알고리즘은 불필요한 전송률 변화와 버퍼 크기가 작은 경우에 QoS를 만족하지 못하는 e-PCRTT 알고리즘의 문제점을 개선하는 것이 목적이다.

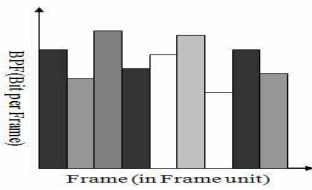
위에서 언급된 알고리즘들은 모두 가변 비트율로 저장된 비디오 데이터를 고정 비트율로 변환시켜서 버스트를 감소시킨다. 그러나, 기존 연구들([2-13])에서는 전송률 변화 횟수, 전송률 변화량, 전송률 증가 횟수 등의 스무딩 알고리즘의 목적을 중심으로 성능을 평가하였다.

본 논문에서는, 스무딩 알고리즘들의 버스트 감소 정도를 분석하기 위해, 버스트에 영향을 미치는 요소들을 중심으로 각 스무딩 알고리즘을 평가한다. 이를 위해, 스무딩 알고리즘을 구현하고, 구현된 스무딩 알고리즘을 다양한 버퍼 크기에서 여러 비디오 데이터를 사용하여 버스트에 영향을 미치는 요소로 평가한다. 버스트에 영향을 미치는 요소에는 프레임을 구성하는 최대 바이트 수 크기 즉 프레임 단위로 전송하는 경우에 요구되는 최대 전송률, GOP를 구성하는 바이트 수의 최대 크기 즉 GOP 단위로 전송할 경우에 경우의 최대 전송률, 프레임당 요구되는 전송률 변화량, GOP당 요구되는 전송률 변화량이다. 프레임을 구성하는 최대 바이트 수와 GOP를 구성하는 최대 바이트 수가 클수록 특정 프레임과 GOP에서 높은 전송률이 요구됨을 의미한다. 그리고, 프레임당 요구되는 전송률 변화량과 GOP당 요구되는 전송률 변화량이 클수록 갑자기 이전 프레임 또는 GOP에 비해 전송률의 증가가 큰 네트워크 자원을 예약해야 하거나, 일정 시간에 비디오 데이터에 대한 접근 속도가 빨라야 함을 뜻한다. 따라서, 이 연구는 가변 비트율로 저장된 비디오 데이터를 클라이언트로 전송하려는 경우에 특정 버스트 관련 요소들을 최적화하는 알고리즘을 선택할 때 유용할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 스무딩을 살펴보고, 3장에서는 성능 평가를 기술한다. 그리고 4장에서는 결론 및 추후 연구 방향을 기술한다.

2. 스무딩

(그림 3)은 가변 비트율로 저장된 비디오 데이터를 보여준다[1,3]. 프레임당 비트 수가 다르므로 이를 그대로 전송하려는 경우 특정 프레임에서 급격하게 높은 전송률이 요구될 수 있다.

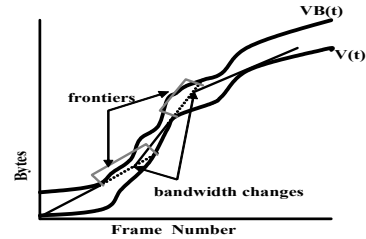


(그림 3) 가변 비트율 방식

이를 개선하기 위해서는, 가변 비트율로 저장된 비디오 데이터를 프레임 당 비트수가 동일한 고정 비트율로 변환시키는 전송 계획이 필요하다. 스무딩 기법은 이를 위한 것이다. (그림 4)는 스무딩 기법을 보여준다[2,5,10]. X축은 프레임 번호이며 Y축은 누적된 프레임들의 바이트 수이다. $V(t)$ 는 언더플로우 경계선으로, 프레임 0부터 프레임 t 까지의 누적된 바이트 수로 표현된다. $VB(t)$ 는 오버플로우 경계선으로 언더플로우 경계선에 클라이언트 버퍼 크기 b 를 더한 값으로 표현된다. 따라서, 서버에서 QoS를 만족시키기 위해서는 언더플로우 경계선보다 크거나 같고 오버플로우 경계선 보다 작거나 같은 전송 계획을 세워야 한다. 서버에서 언더플로우(오버플로우) 경계선보다 작은(큰) 전송률로 프레임을 전송하면 클라이언트에서 언더플로우(오버플로우)가 발생된다. 따라서 QoS를 만족시키는 영역은 언더플로우 경계선과 오버플로우 경계선 사이의 중간 영역이 된다.

스무딩 기법을 위한 알고리즘은 QoS를 만족시키는 영역에서 버스트를 감소시키기 위해 프레임당 비트 수가 동일한 연속적인 프레임들을 검색해야 한다. 이 경우에 검색된 연속적인 프레임들을 런(Run)이라고 하며 이 프레임들의 전송

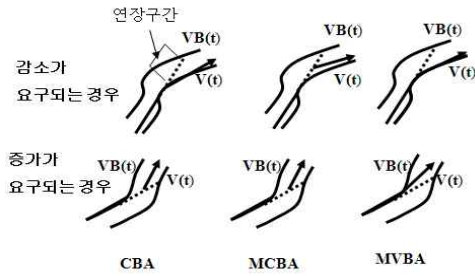
률에 의해 언더플로우(오버플로우) 경계선을 만나는 경우에 이 지점부터 오버플로우(언더플로우)를 발생시키는 프레임까지의 구간을 연장 구간(frontiers)이라고 한다. 이와 같이 스무딩 알고리즘에서는 가변 비트율로 저장된 비디오 데이터를 고정 비트율로 변환시킨다[3,4].



(그림 4) 스무딩 기법의 원리

이 원리를 바탕으로 QoS를 만족시키면서 전송률 변화 횟수, 전송률 변화량, 그리고 전송률 증가 횟수와 같은 특정 요소를 최적화하기 위한 목적으로 다양한 스무딩 알고리즘들이 연구되어져 왔다.

(그림 5)는 CBA, MCBA, MVBA 알고리즘의 전송률 조절 방법이다[2][10]. CBA 알고리즘[5,6,7]에서는 전송률 증가 횟수의 최소화를 위해 언더플로우가 발생되어 전송률 증가가 요구되는 경우에 연장 구간에 속한 프레임들 중에서 새로운 전송률로 QoS를 만족하면서 가장 많은 프레임을 전송할 수 있는 프레임을 검색하여 이 프레임을 새로운 런의 시작 프레임으로 설정한다. 전송률 감소가 요구되는 경우에는 연장 구간의 첫 프레임을 새로운 런의 시작 프레임으로 설정한다. MCBA 알고리즘[8]에서는 전송률 변화 횟수를 최소화하기 위해서 전송률 변화가 요구되는 경우에 연장 구간에서 가장 많은 프레임을 전송할 수 있는 프레임을 새로운 런의 시작 프레임으로 설정한다. MVBA 알고리즘[9,11]에서는 전송률 변화량을 최소화하기 위해 연장 구간의 첫번째 프레임을 새로운 런의 시작 프레임으로 설정한다.



(그림 5) CBA, MCBA, MVBA의 전송률 조절 방법

PCRTT 알고리즘[12]에서는 전송률 변화 횟수 즉 런의 개수가 제약조건으로 주어지고 언더플로우를 피하기 위해 모든 구간에 동일한 오프셋 값을 적용하므로 불필요하게 큰 버퍼 크기가 요구될 수 있다[13]. 이를 개선한 e-PCRTT 알고리즘[13]에서는 언더플로우 경계점과 오버플로우 경계점의 평균점을 해당 구간의 전송률로 설정한다. 그러나, e-PCRTT 알고리즘에서는 PCRTT 방법에서와 같이 전송률 변화 횟수에 대한 제한이 있고 런의 크기가 일정하다. 따라서 전송률 횟수가 큰 경우에는 런의 크기가 작아지므로 연속된 런에서 전송률의 변화가 필요하지 않을 경우에도 강제적으로 변화를 시켜야 하는 경우가 발생할 수 있고, 전송률 변화 횟수가 작은 경우에는 런의 크기가 커지므로 주어지는 버퍼 크기가 작은 경우에 QoS를 보장하지 못할 수도 있다[3-4].

e-PCRTT 방법에서의 문제점들을 해결하기 위한 연구들[3-4]에서는 런의 첫 프레임의 언더플로우 경계점과 오버플로우 경계점의 평균점을 시작점으로 하고 현재 검색중인 프레임의 언더플로우 경계점과 오버플로우 경계점의 평균점을 종점으로 하는 전송률을 설정한다. 그래서, 설정한 전송률이 QoS를 만족하는 경우에는 검색중인 프레임을 현재 런에 추가하고, QoS를 만족하지 않는 경우에는 현재 런을 완성하고 검색중인 프레임을 새로운 런에 배정한다.

3. 실험 결과

실험에 사용된 컴퓨터 사양은 Pentium 4인데, 속도는 2.8 Ghz이고 메모리는 1GB이다. CBA,

MCBA, MVBA 알고리즘 모두를 C 언어로 구현하여 성능을 평가한다.

평가에 사용된 요소는 최대 프레임 바이트 수, 최대 GOP 바이트 수, 프레임당 요구되는 전송률 변화량, GOP당 요구되는 전송률 변화량이다. 이 값들이 커질수록 버스트는 커진다. <표 1>[2,10,11,14]은 실험에 사용된 비디오 소스를 보여준다. Ave Frame Size는 프레임을 구성하는 평균 바이트 수, Max Frame Size와 Min Frame Size는 각각 프레임을 구성하는 바이트 수들 중에서 가장 큰 값과 작은 값이다. 또한, Std Dev는 프레임을 구성하는 바이트 수에 대한 표준 편차로써 이 값이 클수록 프레임들 사이에 바이트 수의 변화가 심하다. 즉 1993 Final Four는 바이트 수의 변화가 가장 심한 소스이고, Seminar는 바이트 수의 변화가 아주 적은 소스이다. 실험에 사용된 비디오 소스들에서 1개의 GOP는 6개의 프레임으로 구성된다.

<표 1> MPEG 비디오 소스들의 파라미터들

Name	Ave Frame Size (KB)	Max Frame Size (KB)	Min Frame Size (KB)	Std Dev (KB)
Croc.Dundee	10.52	18.98	1.233	10.764
E.T.100	15.28	29.836	6.666	15.71
1993 Final Four	16.07	28.872	2.504	16.57
Seminar	8.4	10.791	7.012	8.422

본 논문에서는 스무딩 알고리즘의 버스트를 평가하기 위한 요소로 프레임을 구성하는 최대 바이트 수, GOP를 구성하는 최대 바이트 수, 프레임당 요구되는 전송률 변화량, GOP당 요구되는 전송률 변화량을 계산하였다. 프레임을 구성하는 최대 바이트 수와 GOP를 구성하는 최대 바이트 수는 각각 스무딩 알고리즘을 수행한 후에 각 프레임에서 요구되는 전송률 중에서 가장 큰 값과 각 GOP에서 요구되는 전송률 중에서 가장 큰 값이다. 이 값들이 클수록 높은 전송률이 요구됨을 뜻한다.

CBA, MCBA, MVBA 알고리즘의 최적화 목적인 전송률 증가 횟수, 전송률 변화 횟수, 전송률 변화량은 기존 연구들([2,5,6,7,8,9,10])에서 성능을 평가하였으므로 본 논문에서는 제외한다.

(그림 6)은 프레임을 구성하는 최대 바이트

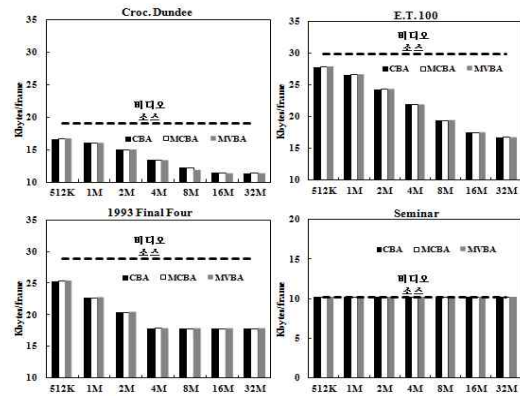
수를 비교한 결과이다. 식 (1)은 프레임을 구성하는 최대 바이트 수를 나타낸다. n은 전송률 변화 횟수, C_i 는 i 번째 프레임에서 전송이 요구되는 바이트 수이다.

$$\max_{1 \leq i \leq n} \{C_i\} \dots\dots\dots(1)$$

비디오 소스는 스무딩 알고리즘을 수행하기 이전에 프레임을 구성하는 최대 바이트 수를 나타낸다. 스무딩 알고리즘을 수행하기 이전에 프레임을 구성하는 바이트 수가 Seminar, Croc.Dundee, 1993 Final Four, E.T.100 비디오 데이터 순서대로 컸는데, 이와 동일한 순서로 각 스무딩 알고리즘을 수행한 후에 프레임을 구성하는 최대 바이트 수가 크다. 이는 원래 비디오 소스의 프레임을 구성하는 최대 바이트 수가 있는 부분을 스무딩 알고리즘들에서도 프레임을 구성하는 최대 바이트 수로 설정했기 때문이다.

모든 비디오 데이터에서 버퍼 크기가 커질수록 최대 프레임을 구성하는 바이트 수가 작아지는데, 이는 버퍼 크기가 커질수록 완만하게 전송률을 변화시킬 수 있기 때문이다.

전송률 변화량을 최소화하는 MVBA 알고리즘에서의 프레임을 구성하는 바이트 수가 다른 알고리즘에서 보다 작는데, 이는 MVBA 알고리즘에서 전송률을 완만하게 증가시키기 때문이다. 특히 MVBA 알고리즘의 경우 Croc.Dundee 비디오 데이터에서 버퍼 크기가 8M, 16M, 32M인 경우에서 특히 크다. 이는 다른 알고리즘의 경우 버퍼 크기가 클 때, 연장 구간이 커서 전송률이 증가되는 경우 이전 전송률에 비해 차이가 큰 전송률을 설정할 가능성이 커지기 때문이다. MCBA 알고리즘에서의 프레임을 구성하는 바이트 수의 크기가 CBA 알고리즘에서보다 크다. 이는, MCBA 알고리즘의 경우 전송률의 증가가 요구되는 경우와 감소가 요구되는 경우 두 경우 모두에서 요구되는 경우에 전송률을 변화량을 고려하지 않고 전송 계획을 세우지만, CBA 알고리즘에서는 전송률의 증가가 요구되는 경우에는 MCBA 알고리즘과 동일하지만, 전송률의 감소가 요구되는 경우에 전송률의 감소를 최소화하는 프레임을 새로운 런의 시작 프레임으로 설정하기 때문이다.



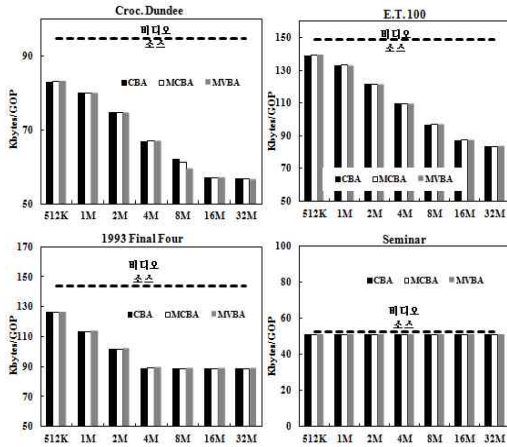
(그림 6) 프레임을 구성하는 최대 바이트 수를 비교한 결과

MCBA 알고리즘에서의 프레임을 구성하는 바이트 수가 CBA, MVBA 알고리즘에서보다 다소 크다. Seminar 비디오 데이터의 경우 원 비디오 소스 자체가 프레임간의 크기가 거의 없어서 버퍼 크기가 커지더라도 프레임을 구성하는 최대 바이트 수의 크기에 큰 변화가 없다. 따라서, 프레임을 구성하는 최대 바이트 수를 최적화시키기 위해서는 MVBA 알고리즘을 사용하는 것이 적합하다.

스무딩 알고리즘을 수행한 후 프레임을 구성하는 최대 바이트 수는 <표 1>의 Max Frame Size가 큰 순서대로 크다. 즉 Seminar, Croc.Dundee, 1993 Final Four, E.T.100 순서대로 크다.

(그림 7)은 GOP를 구성하는 최대 바이트 수를 비교한 결과이다. 이 값은 전송 계획을 세운 후에, 각 GOP를 구성하는 프레임들의 전송률을 더한 값들 중에서 가장 큰 값이다. 식 (2)는 이를 나타낸다. GOP_n 은 총 GOP 개수, GOP_i 는 i 번째 GOP에서 요구되는 전송률 즉 바이트 수이다.

$$\max_{1 \leq i \leq GOP_n} \{GOP_i\} \dots\dots\dots(2)$$



(그림 7) GOP을 구성하는 최대 바이트 수를 비교한 결과

이 값이 클수록 클라이언트에서 비디오 데이터를 랜덤으로 접근하려는 경우 갑자기 높은 전송률이 요구될 수 있으며 화면의 변화가 심한 비디오 영상을 클라이언트에서 시청할 수 있다. 대략적으로, (그림 6)에서의 값들보다 약 6배 정도 큰데, 이는 GOP를 구성하는 프레임들의 개수가 6이기 때문이다. (그림 6)에서와 동일한 이유로 버퍼 크기가 커질수록 GOP를 구성하는 최대 바이트 수도 작아진다. 전송률 변화량을 고려하지 않는 MCBA 알고리즘에의 GOP를 구성하는 최대 바이트 수가 CBA, MVBA 알고리즘보다 크다. 결과적으로, GOP를 구성하는 최대 바이트 수를 줄이려는 전송 계획을 얻고자 하는 경우에 CBA와 MVBA 알고리즘을 사용하는 것이 적합하다. 즉, 랜덤하게 비디오 데이터를 빠르게 재생하기 원하려는 경우에 CBA, MVBA 알고리즘이 MCBA 알고리즘보다 유리할 것이다.

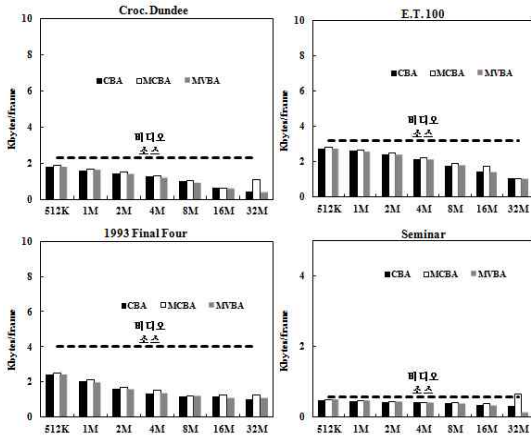
(그림 8)은 프레임당 요구되는 전송률의 변화량을 비교한 결과이다. 이 값이 클수록 프레임간의 화면 변화가 심해지며, 급격하게 높은 전송률이 요구될 수 있다. 이 값은 식 (3)에 의해 계산되며, n은 비디오 데이터를 구성하는 프레임들의 개수, c_i 는 i 번째 프레임에서의 전송률, 그리고 $stddev$ 는 표준 편차이다.

$$\frac{stddev\{c_0, c_1, \dots, c_{n-1}\}}{\frac{1}{n} \times \sum_{i=0}^{n-1} c_i} \dots (3)$$

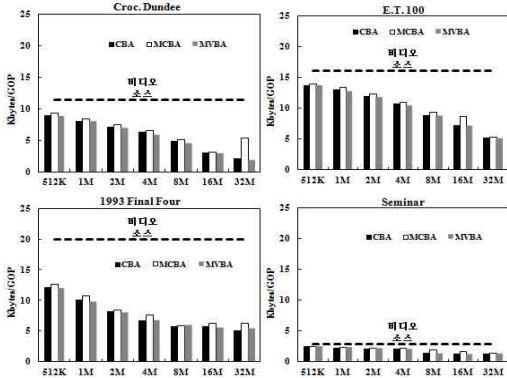
전송률 변화량을 고려하지 않는 MCBA 알고리즘에서의 프레임당 요구되는 전송률 변화량이 가장 크다. 특히, Croc.Dundee 비디오 데이터에서 버퍼 크기가 32M인 경우와 E.T.100 비디오 데이터에서 버퍼 크기가 16M인 경우, 그리고 Seminar 비디오 데이터에서 버퍼 크기 32M인 경우에, 연장 구간의 크기가 커져서 이전 전송률에 비해 급격하게 전송률이 변화되는 프레임들이 새로운 런의 시작 프레임으로 설정된 경우가 많아서 프레임당 요구되는 전송률 변화량이 다른 알고리즘들보다 훨씬 크다. <표 1>에서 비디오 소스의 프레임을 구성하는 바이트 수의 변화가 가장 적은 Seminar가 프레임당 요구되는 전송률 변화량이 가장 적다. 그러나, MCBA 알고리즘에서 비디오 데이터가 Seminar이고 버퍼 크기가 32M인 경우에 프레임당 전송률 변화량이 비디오 소스의 프레임 변화량 보다 훨씬 크다. 이는, Seminar 비디오 데이터의 경우 연속적인 프레임들간의 바이트 수의 변화가 아주 작는데, 버퍼 크기가 충분히 커서 새로운 런의 시작 프레임으로 설정된 프레임들이 이전 런을 구성하는 프레임들의 바이트 수보다 차이가 심한 프레임들이 설정되는 경우가 많았기 때문이다. 따라서, MCBA 알고리즘의 경우 프레임당 전송률 변화량이 CBA, MVBA 알고리즘에서보다 크므로 이전 런의 대역폭보다 훨씬 큰 대역폭을 요구하는 경우에 CBA, MVBA 알고리즘들보다 불리할 수 있다.

(그림 9)는 GOP당 요구되는 전송률 변화량을 비교한 결과이다. 이 값이 클수록 GOP간의 화면 변화가 심하며, 랜덤하게 비디오 데이터를 접근하려는 경우 급격하게 높은 전송률이 요구될 수 있다. 식 (4)는 GOP당 요구되는 전송률의 변화량을 나타낸다. GOP_n 은 총 GOP 개수, GOP_i 는 i 번째 GOP에서 요구되는 전송률 즉 바이트 수이다. 그리고 $stddev$ 는 표준 편차이다.

$$\frac{stddev\{GOP_0, GOP_1, \dots, GOP_{n-1}\}}{\frac{1}{GOP} \times \sum_{i=0}^{GOP-1} GOP_i} \dots (4)$$



(그림 8) 프레임당 요구되는 전송률의 변화량을 비교한 결과



(그림 9) GOP당 전송률 변화량을 비교한 결과

실험에 사용된 모든 알고리즘들에서 GOP당 요구되는 전송률을 변화량을 비교한 결과는 (그림 8)에보다 약 6배 정도 큰 결과를 보인다. 또한, (그림 8)에서와 같은 이유로, MCBA 알고리즘들이 다른 알고리즘들에 비해 큰 값을 갖고 MVBA 알고리즘이 다른 알고리즘들에 비해 작은 값을 갖는다. GOP당 요구되는 전송률 변화량은 (그림 8)의 프레임당 요구되는 전송률 변화량의 크기와 동일하다. 이는 GOP는 프레임으로 구성되기 때문이다. 실험 결과, MVBA 알고리즘에서는 클라이언트에서 랜덤하게 비디오 데이터 재생을 원하러는 경우에 다른 알고리즘들에서보다 쉽게 네트워크 대역폭을 얻을 수 있을 것이다.

4. 결론 및 추후 연구 방향

급격하게 높은 전송률이 요구되는 버스트는 스무딩 알고리즘의 전송 계획에 많은 영향을 미친다. 버스트가 많이 발생하거나 큰 경우에 대역폭 등의 네트워크 자원이 많이 사용될 수 있으며, 네트워크 자원 예약 또한 어려워질 수 있다.

본 논문에서는 스무딩 알고리즘의 버스트 감소되는 정도를 살펴보기 위해, 전송 계획을 세운 후에 최대 프레임 바이트 수와 최대 GOP 바이트 수, 그리고 각 프레임당 요구되는 전송률 변화량과 각 GOP당 요구되는 전송률 변화량을 여러 비디오 데이터를 사용하여 다양한 버퍼 크기에서 평가하였다.

평가 결과, 최대 프레임 바이트 수, 최대 GOP 바이트 수, 프레임당 요구되는 전송률 변화량, GOP당 요구되는 전송률 변화량 모두 MCBA 알고리즘에서 Seminar 비디오 데이터의 버퍼 크기가 32M인 경우를 제외하고 원래 비디오 소스보다 모두 감소되어졌다. 이는 스무딩 알고리즘을 수행하는 경우 버스트가 감소됨을 보여준다. 또한, MVBA 알고리즘이 실험에 사용된 모든 버스트 관련 평가 요소에서 CBA, MCBA 알고리즘에 비해 우수하였다. 따라서, 버스트 관련 요소를 최적화시키려는 경우에 MVBA 알고리즘의 사용이 가장 적합할 것이라고 판단된다.

추후에는 프레임간 버스트와 GOP간 버스트를 감소시키는 알고리즘을 연구할 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] Wu-chi Feng, "A Comparison of Bandwidth Smoothing Techniques for the Transmission of Pre-recorded Compressed Video", in Proc. IEEE INFOCOM, pp.58-66, April 1997.
- [3] 이면재, "비디오 데이터 전송을 위한 스무딩 알고리즘에 관한 연구, 홍익대학교, 2005.
- [4] 이면재, 이준용, 박도순, "가변 비트율 비디오 전송을 위한 효율적인 스무딩 알고리즘", 정보처리학회논문지C, 제11-1C권 제 7호, 2004.12.
- [5] 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.

[6] W. Feng, et. al., "Smoothing and buffering for delivery of prerecorded compressed video", in Proc. of ISE T/SPIE Symp. on Multimedia Comp. and Networking, pp. 234-242, Feb 1995.

[7] W. Feng, et.al., "Optimal Buffering for the Delivery of Compressed Prerecorded Video", Proc. of the IAS TED/ISMM Intl Conference on Networks, January 1995.

[8] W.Feng, F.Jahanian, S.Sechrest, "An Optimal Bandwidth Allocation Strategy for the Delivery of Compressed Prerecorded Video", ACM/Springer-Verlag Multimedia Systems, vol. 5, no. 5, pp.297-309, Sept 1997.

[9] 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.

[10] W. Feng and J. Rexford, "Performance evaluation of smoothing algorithms for transmitting prerecorded VBR video", IEEE Trans. on Multimedia, September 1999.

[11] Wu-Chi Feng, "VIDEO-ON-DEMAND SERVICES: EFFICIENT TRANSPORTATION AND DECOMPOSITION OF VARIABLE BIT RATE VIDEO", PH.D, Computer Science and Engineering, 1996.

[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] 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.

[14] <http://www.cis.ohio-state.edu/~wuchi>.

이 면 재



1992년 : 홍익대학교 전자계산학과(학사)
 1994년 : 홍익대학교 전자계산학과(석사)
 2006년 : 홍익대학교 전자계산학과(박사)

2009년~현재 : 백석대학교 정보통신학부 교수
 관심분야: 게임 프로그래밍, 게임 엔진, 기능성 게임, 멀티미디어