

개선된 GOP 단위의 스무딩 알고리즘

이면재*

요약

스무딩은 가변 비트율로 저장된 비디오 데이터를 클라이언트로 전송할 때 일련의 고정 비트율로 전송할 수 있도록 전송 계획을 세우는 것으로 이러한 스무딩 알고리즘에는 CBA, MCBA, MVBA 등이 있다. 그러나, 이 알고리즘들은 프레임 단위로 전송 계획을 세우므로 프레임 버스트 또는 GOP 버스트가 발생할 수 있다. 이를 개선하기 위하여 MVBAG 알고리즘에서는 GOP 단위로 전송률을 계산한다. 그러나, 이 알고리즘에서는 계산된 전송률보다 급격히 프레임의 크기가 작거나 큰 경우에는 QoS를 보장하지 못할 수 있다. 본 논문에서는 MVBAG 알고리즘의 문제점을 개선하는 알고리즘을 제안하고 QoS를 만족시키지 못하는 프레임 개수, 프레임당 평균 전송률 변화량, GOP당 평균 전송률 변화량과 같은 평가 요소를 사용하여 성능을 평가한다. 실험 결과, 제안 알고리즘은 QoS를 만족하는 프레임 개수 비교에서 우수함을 보인다.

Enhanced Smoothing Algorithm Using GOP Unit

Myoun-Jae Lee*

Abstract

Smoothing is a transmission plan where variable rate video data is converted to a constant bit rate stream. These smoothing algorithms include CBA, MCBA, MVBA, PCRTT and others. But, these algorithms build a transmission plan per frame unit. So, these algorithms cause frame burst or GOP burst. In order to improve it, MVBAG algorithm build a transmission plan per GOP. But this algorithm may not guarantee QoS when frame's size is abruptly larger or smaller than the computed transmission rate.

In this paper, a smoothing algorithm is proposed to enhance MVBAG algorithm's problem. In order to show the proposed algorithm's performance, the proposed algorithm is compared with MVBAG algorithm using various evaluation factors such as number of frames that do not meet the QoS, average transmission rate variability per frame, average transmission rate variability per GOP. Experimental results show that the proposed algorithm outperforms MVBAG algorithm in number of frames that do not meet the QoS.

Keywords : GOP, Smoothing Algorithm, MPEG, Transmission Plan, Best-Effort Service, QoS

1. 서론

MPEG은 프레임간의 연관성을 이용해 동영상의 압축률을 높인 방법으로 I(Intra picture), B(Bidirectionally predicted picture), 그리고 P(Unidirectionally predicted picture)의 세 가지

프레임으로 구성된다[4]. GOP(Group of Picture)는 하나의 I 프레임에서 다음 I 프레임 직전 프레임까지의 집합으로 구성되며 임의로 접근 가능한 최소의 독립적인 단위이다.

MPEG 비디오 데이터는 저장 공간과 네트워크 자원의 효율적인 사용을 위해 프레임 당 비트 수가 다른 가변 비트율 형태로 저장된다. 그러나, 이와 같이 저장된 비디오 데이터를 전송 계획을 세우지 않고 그대로 전송하는 경우에는 QoS를 보장하기 위해서 전송률을 급격히 증가시켜야 되는 버스트 현상이 발생할 수 있다 [4-7]. 이 문제를 해결하기 위해서는 가변 비트

※ 제일저자(First Author) : 이면재
접수일:2011년 11월 15일, 수정일:2011년 12월 01일
완료일:2011년 12월 08일
* 백석대학교 정보통신학부 멀티미디어전공
davidlee@bu.ac.kr

율로 저장된 비디오 데이터를 고정 비트율로 전송할 수 있는 계획이 필요하다. 스무딩은 이를 위한 알고리즘으로 CBA(Critical Bandwidth Allocation)[8], MCBA(Minimum Changes Bandwidth Allocation)[9], MVBA(Minimum Variability Bandwidth Allocation)[10] 등이 있다.

그러나, 위에서 언급된 스무딩 알고리즘들은 프레임 단위로 전송 계획을 세우므로 프레임간 버스트와 GOP간 버스트가 발생할 수 있다.

이 문제를 개선한 MVBAG(Minimum Variability Bandwidth Allocation over Group of Picture) 알고리즘[1,2]에서는 MVBA 알고리즘을 기반으로 하여 GOP 단위로 전송 계획을 세운다. 즉, GOP 단위로 전송 계획을 세우고, 해당 GOP의 전송률을 GOP를 구성하는 프레임의 개수로 나눈 값을 해당 GOP를 구성하는 프레임들의 전송률로 설정한다. 따라서, 현재 검색되는 GOP에서의 전송률이 언더플로우(오버플로우) 경계점에 근접하고 계산된 프레임의 전송률이 특정 프레임의 크기보다 작은(큰) 경우 QoS를 보장하지 못할 수 있다.

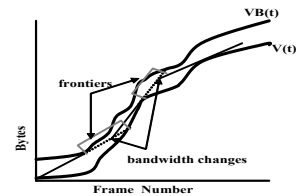
본 논문에서는 이를 개선하기 위한 방법을 제안하고 다양한 버퍼 크기에서 제안 알고리즘의 성능을 평가한다.

본문의 구성은 다음과 같다. 2장에서는 스무딩을 살펴보고, 3장에서는 개선된 MVBAG 알고리즘을 제시한다. 그리고 4장에서는 실험 결과를 설명하고, 5장에서는 결론 및 추후 연구 방향을 기술한다.

2. 스무딩

(그림 1)은 스무딩 기법의 원리[3,11]를 나타내는데, X축은 시간 즉 프레임 번호이며 Y축은 바이트 수이다. $V(t)$ 와 $VB(t)$ 는 각각 언더플로우 경계선과 오버플로우 경계선을 의미한다. 언더플로우 경계선은 프레임 0부터 프레임 t 까지의 누적된 바이트 수로 식 (1)과 같이 표현되는데, f_i 는 프레임 i 의 바이트 수를 의미한다. 오버플로우 경계선은 식 (2)와 같이 언더플로우 경계선에 클라이언트 버퍼 크기 b 를 더한 값으로 표현된다. 서버에서는 언더플로우(오버플로우) 경계선보다 작은(큰) 전송률로 프레임을 전송하면 클라

이언트에서 언더플로우(오버플로우)가 발생되어 QoS를 보장할 수 없다. 따라서, QoS를 만족하는 전송 계획을 세우기 위해서는 계산된 전송률이 언더플로우 경계선과 오버플로우 경계선 영역 내에 있어야 한다. 이 경우에 스무딩 알고리즘은 언더플로우 경계선과 오버플로우 경계선 사이의 영역에서 동일한 전송률로 전송할 수 있는 연속적인 프레임들을 검색해야 한다. 이때 검색된 프레임들을 런(Run)이라고 하며 이 전송률에 의해 언더플로우(오버플로우) 경계선을 만나는 경우에 이 지점부터 오버플로우(언더플로우)를 발생시키는 프레임까지의 구간을 연장 구간(frontiers)이라고 한다. 스무딩 알고리즘은 연장 구간에 속한 프레임들 중에서 스무딩 알고리즘의 목적에 적합한 프레임을 검색하여 이 프레임에서 전송률을 변화시킨다.



(그림 1) 스무딩 기법의 원리

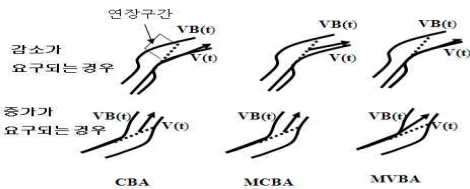
$$V(t) = \sum_{i=0}^t f_i \quad (1)$$

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

(그림 1)의 스무딩 기법의 원리를 바탕으로 QoS를 만족하면서 전송률 변화 횟수, 전송률 변화량, 그리고 전송률 증가 횟수와 같은 특정 요소를 최적화하기 위한 목적으로 다양한 스무딩 알고리즘들이 연구되어져 왔다.

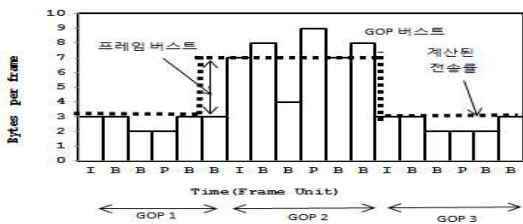
(그림 2)는 CBA, MCBA, MVBA 알고리즘의 전송률 조절 방법이다[8][11]. CBA 알고리즘[8]에서는 전송률 증가 횟수의 최소화를 위해 전송률 증가가 요구되는 경우에는 연장 구간에 속한 프레임들 중에서 새로운 전송률로 언더플로우나 오버플로우가 발생되지 않고 가장 많은 프레임을 전송할 수 있는 프레임을 검색하여 이 프레임을 새로운 런의 시작 프레임으로 설정한다. 전송률 감소가 요구되는 경우에는 연장 구간의 첫 프레임을 새로운 런의 시작 프레임으로 설정한

다. MCBA 알고리즘[9]에서는 전송률 변화 횟수를 최소화하기 위해서 전송률 변화가 요구되는 경우에 연장 구간에서 가장 많은 프레임 전송할 수 있는 프레임을 새로운 런의 시작 프레임으로 설정한다. MVBA 알고리즘[10]에서는 전송률 변화량을 최소화하기 위해 연장 구간의 첫 프레임을 새로운 런의 시작 프레임으로 설정한다.



(그림 2) CBA, MCBA, MVBA의 전송률 조절 방법

그러나, 위에서 언급된 스무딩 알고리즘들은 프레임 단위로 전송 계획을 세우므로, 프레임간의 버스트와 GOP간의 버스트가 발생할 수 있다. (그림 3)은 이를 보여준다. 이 예에서는 GOP를 구성하는 프레임 개수를 6이라고 가정한다. 기존 스무딩 알고리즘들의 경우 1번째 GOP의 6번째 프레임에서와 같이 GOP 내에서 전송률이 변화되는 경우 동일한 GOP 내에서도 프레임 버스트가 일어날 수 있다. 또한 첫 번째 GOP와 두 번째 GOP 2의 예에서와 같이 각 GOP에서 요구되는 전송률의 차이가 커지는 버스트가 발생할 수 있다. 즉 첫 번째 GOP에서 보내야 되는 바이트 수는 22바이트인데, 두 번째 GOP에서 보내야 되는 바이트 수는 42 바이트로 첫 번째 GOP 2배 정도가 된다.

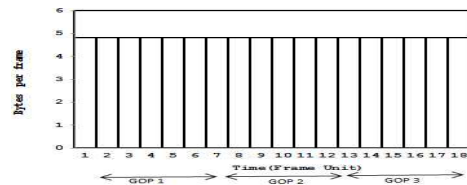


(그림 3) 기존 스무딩 알고리즘들의 문제점 예

MVBAG 알고리즘[1,2]에서는 (그림 3)과 같은

기존 스무딩 알고리즘들의 문제점을 개선하기 위하여, MVBA 알고리즘을 기반으로 하되, GOP 단위로 전송률을 계산한다. 그리고, 해당 GOP의 전송률을 GOP를 구성하는 프레임의 개수로 나눈 값을 각 프레임에서 요구되는 전송률로 설정한다.

(그림 4)는 (그림 3)에서의 문제점을 개선한 MVBAG 알고리즘의 전송 계획을 보여준다. 이때 이 구간에서 전송률 증가가 요구된다고 가정한다. 첫 번째, 두 번째, 세 번째 GOP의 바이트 수는 각각 16, 42, 15가 되어, 각 GOP에서 누적된 바이트 수는 16, 58, 73이 된다. 첫 번째 GOP에서의 전송률은 16(16/1), 두 번째 GOP에서의 전송률은 29(58/2), 세 번째 GOP의 전송률은 24.3(73/3)로 임시적으로 계산된다. 이 중에서, 전송률 증가가 요구되므로 이 구간의 전송률은 MVBA 알고리즘에서와 같이 QoS를 만족하는 전송률 중에서 가장 큰 전송률인 29로 설정된다. 따라서, 모든 GOP의 전송률은 29로 설정되어 각 프레임들의 전송률은 4.83(29/6)이 된다.



(그림 4) (그림 3)에 대한 MVBAG 알고리즘의 전송 계획

이와 같이 MVBAG 알고리즘에서는 GOP내에서 전송률이 변화되지 않으므로 (그림 4)에서와 같이 GOP 내에서 프레임 버스트가 일어나지 않으며, GOP 단위로 전송률 변화량을 최소화하여서 GOP간 버스트도 기존 스무딩 알고리즘에 비해 훨씬 작아진다.

3. 개선된 MVBAG 알고리즘

MVBAG 알고리즘에서는 계산된 각 GOP의 전송률을 GOP를 구성하는 프레임 개수로 나누는데, 이 값이 해당 프레임에서의 전송률이 된다. 그러나, 이 방법은 현재 검색되는 GOP의 전송률이 언더플로우(오버플로우) 경계점에 근접하

고 설정된 프레임의 전송률보다 훨씬 작은(큰) 프레임이 있을 경우 언더플로우(오버플로우)가 발생할 수 있다. <표 1>은 이를 보여준다. GOP의 개수는 6이라고 가정하고 크기 단위는 Byte이다. GOP를 구성하는 바이트 수는 36이며, 이 값을 GOP를 구성하는 개수인 6으로 나누면 각 프레임에서의 전송률은 6이 된다. 그래서 프레임 1에서는 6 바이트를 보내는데, 이 경우 언더플로우 경계선과 오버플로우 경계선 사이에 있어서 QoS를 만족한다. 그러나, 프레임 3과 프레임 4에서는 누적된 전송률이 언더플로우 경계선보다 작은 값을 갖게 되어 언더플로우가 발생한다.

<표 1> MVBAG 알고리즘에서 QoS를 만족하지 못하는 예

번호	1	2	3	4	5	6
크기	4	7	8	9	2	6
V(t)	4	11	19	28	30	36
VB(t)	14	21	29	38	40	46
전송량	6	12	18	24	30	36

이 문제점을 개선하기 위해서, 본 논문에서는 MVBAG 방식에서와 같이 GOP 단위로 전송률을 계산하고, 이 전송률로 프레임들을 보내는 경우에 QoS를 만족하는 여부를 점검한다. 만약 QoS가 만족되지 않는 경우 QoS를 만족시킬 수 있는 새로운 런을 구한다. 즉 설정된 프레임의 전송률로 프레임들을 전송하려고 할때 언더플로우(오버플로우)가 발생하는 경우에는 MVBA 알고리즘에서와 같이 QoS를 만족하는 전송률 중에서 최대(최소) 전송률을 해당 런의 전송률로 계산한다. 이때 런의 마지막 프레임은 특정 GOP의 마지막 프레임으로 구성하여 동일한 GOP내에서 전송률이 급격하게 변화되지 않도록 한다. 이와같은 방법으로 제안 방법에서는 GOP 단위로 전송률을 계산하면서도, QoS를 보장한다. <표 2>는 <표 1>의 프레임들에 대한 제안 방법의 전송 계획을 보여준다. 계산된 프레임의 전송률로 프레임들을 전송하려는 경우, 프레임 3과 프레임 4에서 언더플로우가 발생된다. 따라서, QoS를 만족하는 최대 전송률과 최소 전송률을 각각 구하여 새로운 런을 계산한다. 즉 프레임 1에서까지의 QoS를 만족하는 최소(최대) 전송률은 4(14)가 된다. 프레임 2에서의 QoS를 만족하

는 최소 전송률과 최대 전송률은 각각 11과 21을 현재까지의 프레임 개수인 2로 나눈 값인 5.5와 10.5가 된다. 동일한 방법으로 각 프레임에서 QoS를 만족하는 최대 전송률과 최소 전송률을 각각 구한다. 이중에서, 기존 MVBAG 알고리즘으로 프레임을 보내는 경우 언더플로우가 발생되므로, QoS를 만족하는 최대 전송률인 7.7을 해당 런의 전송률로 설정한다. 이 결과, 각 프레임에서 누적된 전송량은 언더플로우 경계선과 오버플로우 경계선 사이에 있게 되어 QoS를 만족하게 된다.

<표 2> 개선된 제안 알고리즘

번호	1	2	3	4	5	6
크기	4	7	8	9	2	6
V(t)	4	11	19	28	30	36
VB(t)	14	21	29	38	40	46
최소전송률	4	5.5	6.3	7	6	6
최대전송률	14	10.5	9.7	9.5	8	7.7
전송량	7.7	15.4	23.1	30.8	38.5	46.2

4. 실험 결과

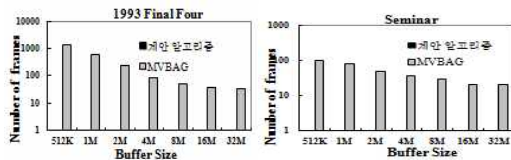
실험에 사용된 컴퓨터 사양은 Pentium 4인데, 속도는 2.8 Ghz이고 메모리는 1GB이다. 제안 알고리즘과 MVBAG 알고리즘 모두 C 언어로 구현하여 성능을 평가한다.

<표 3>은 사용된 비디오 소스에 대한 파라미터들[11,12]이다. Frame Std.Dev와 GOP Std.Dev는 각각 프레임과 GOP의 바이트 수에 대한 표준 편차로써 이 값이 클수록 프레임들과 GOP 사이에 바이트 수의 변화가 심하다. 즉 1993 Final Four는 바이트 수의 변화가 심한 소스이고, Seminar는 바이트 수의 변화가 적은 소스이다. GOP 패턴은 IBBPBB이다. 제안 알고리즘의 성능 평가를 위해 버퍼 크기를 다양하게 설정하면서 QoS를 만족하는 프레임 개수, 프레임당 전송률 변화량, GOP당 전송률 변화량을 MVBAG 알고리즘과 비교한다.

<표 3> MPEG 파라미터

Name	Frame Std.Dev	GOP Std.Dev
1993 Final Four	16.57	23.93
Seminar	8.422	3.47

(그림 5)는 QoS를 만족하는 프레임 개수 비교이다. 두 개의 비디오 데이터 모두에서 제안 알고리즘에서의 QoS를 만족하지 못하는 프레임 개수는 0인데, 이는 제안 알고리즘에서는 QoS를 만족하지 않는 경우에 QoS를 만족하는 전송률을 설정하기 때문이다. MVBAG 알고리즘에서 실험에 사용된 두 개의 비디오 데이터 모두 버퍼 크기가 커질수록 전송률을 완만하게 변화시킬 수 있어서 QoS를 만족하지 못하는 프레임 개수가 적어진다.



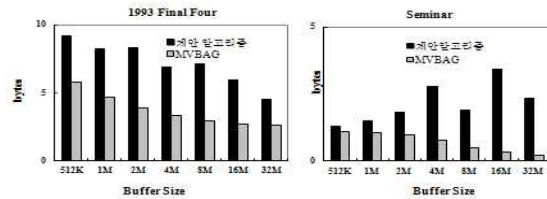
(그림 5) QoS를 만족하지 못하는 프레임 개수 비교

(그림 6)은 각 프레임 당 평균 전송률 변화량이다. 전송률 변화량은 식 (3)에 의해 계산되며, n 은 비디오 프레임을 구성하는 프레임 개수, c_i 는 i 번째 프레임에서의 전송률, 그리고 $stddev$ 는 표준 편차이다.

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

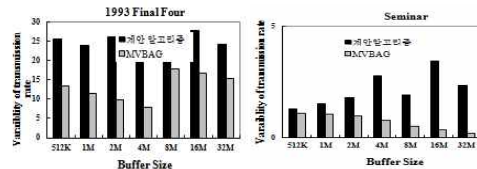
제안 알고리즘의 전송률 변화량이 MVBAG 알고리즘에서보다 크다. 이는, 제안 알고리즘에서 QoS를 만족하지 않는 경우 전송률 변화량을 최소화시키는 GOP에서 전송률을 변화시키지 않기 때문이다. MVBAG 알고리즘의 경우 버퍼 크기가 커질수록 완만하게 전송률을 변화시킬 수 있어서 전송률 변화량이 작아진다. 그러나, 제안 알고리즘에서는 Seminar 비디오 데이터의 버퍼 크기가 16M인 경우에서와 같이 전송률을 변화시켜야 할 경우에 현재 런의 전송률과 큰 차이

가 발생할 수 있는 프레임이 새로운 런의 시작 프레임으로 검색되기 때문에 급격하게 전송률 변화량이 큰 경우도 발생된다. 두 알고리즘 모두에서 가변 비트율로 저장된 비디오 데이터를 고정 비트율로 변환시키는 전송 계획을 세우기 때문에 (표 3)의 비디오 데이터를 구성하는 프레임 바이트 수의 표준 편차보다 작다. 즉, 원래 비디오 데이터보다 두 알고리즘 모두 프레임 버스트가 감소된다.



(그림 7) 프레임당 전송률 변화량 비교

(그림 7)은 각 GOP 당 평균 전송률 변화량이다. 이 값은 식 (3)에서 프레임을 GOP로 변경하여 계산된다. 제안 알고리즘에서의 전송률 변화량은 전송률 변화량을 최소화하는 MVBAG 알고리즘에서보다 훨씬 크다. 이는, 이전 GOP의 전송률보다 훨씬 차이가 큰 GOP가 런의 시작 GOP로 설정되기 때문이다. 두 알고리즘 모두 GOP 단위로 전송 계획을 세우기 때문에 (표 3)에서의 GOP를 구성하는 바이트 수의 표준 편차보다 훨씬 적은 값을 갖는다. 즉, GOP간의 버스트도 감소된다.



(그림 7) GOP당 평균 전송률 변화량 비교

5. 결론 및 추후 연구 방향

기존 CBA, MCBA, MVBA 스무딩 알고리즘들에서는 프레임 단위로 전송 계획을 세우기 때문에 GOP간의 버스트와 동일한 GOP 내에서 프레임간 버스트가 발생 될 수 있다. MVBAG 알

고리즘에서는 이 문제점을 개선하기 위하여 GOP 단위로 전송률을 설정하여 전송 계획을 세운다. 그리고 계산된 GOP의 전송률을 GOP를 구성하는 프레임 개수로 나눈 값으로 프레임의 전송률을 설정한다. 그러나, 이 방법에서는 계산된 GOP의 전송률보다 변화량이 큰 프레임이 GOP내에 있는 경우에 언더플로우 또는 오버플로우가 발생될 수 있다.

본 논문에서는, 이 문제점을 개선하기 위하여 MVBAG 알고리즘에서와 같이 GOP 단위로 전송률을 계산하고, 계산된 전송률로 프레임들을 보내려는 경우에 QoS를 만족하지 않는 경우에 QoS를 만족시키기 전송률을 새롭게 계산하여 이 전송률로 해당 런의 전송률을 설정하는 알고리즘을 제안하였다. 그리고, 2개의 비디오 데이터 소스를 가지고 버퍼 크기를 다양하게 설정하여 QoS를 만족시키지 않는 프레임 개수, GOP당 평균 전송률 변화량, 프레임당 평균 전송률 변화량을 MVBAG 알고리즘과 비교하였다. 제안 알고리즘은 모든 경우에서 QoS를 만족시켰지만, MVBAG 알고리즘에서보다 GOP당 평균 전송률 변화량, 프레임당 평균 전송률 변화량은 컸다. 이는 MVBAG 알고리즘이 전송률 변화량을 최소화시키는 MVBA 알고리즘을 기반으로 했기 때문이다.

추후에는 다양한 비디오 소스를 이용하여 제안 알고리즘과 CBA, MCBA, MVBA 알고리즘 등과 성능을 평가할 예정이다.

참 고 문 헌

[1] 박준원, MPEG의 특성을 이용한 최소 변화폭 전송률 조절 알고리즘, 홍익대학교. 2003.
 [2] 박준원, 이면재, 송하윤, 박도순, "MPEG 동영상 전송을 위한 GOP 단위의 최소 변경 대역폭 할당 기법", 한국정보처리학회논문지C, 제9-C권 제 5호, 2002.10.
 [3] 이면재,이준용, 박도순, "가변 비트율 비디오 전송을 위한 효율적인 스무딩 알고리즘", 정보처리학회논문지C, 제11-1C권 제 7호, 2004.12
 [4] D. Le Gall, "MPEG: A video compression standard for multimedia applications", Communications of the ACM, vol.34, pp.47-58, April 1991.
 [5] Wu-chi Feng, Ming Liu, "Critical Bandwidth Allocati

on Technique for Stored Video Delivery Across Best-Effort Network", pp.25, (OSU- CISRC-8/98-TR3 2) Electronic report under 1998/TR32.ps.
 [6] Wu-Chi Feng, Ming Liu, "Extending critical bandwidth allocation Techniques for stored video delivery across best-effort networks", International Journal of COMMUNICATION SYSTEMS, Vol.14, pp.925-940, Sep 2001.
 [7] 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.
 [8] W. Feng, S. Sechrest, "Critical Bandwidth Allocation for the Delivery of Compressed Pre-recorded Video," Computer Communications, Vol.18, No.10, pp.709-717, Oct 1995.
 [9] W.Feng, F.Jahanian, S.Sechrest, "An Optimal Bandwidth Allocation Strategy for the Delivery of Compressed Pre-recorded Video", ACM/Springer-Verlag Multimedia Systems, vol. 5, no. 5, pp.297-309, Sept 1997.
 [10] 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.
 [11] W. Feng and J. Rexford, "Performance evaluation of smoothing algorithms for transmitting pre-recorded VBR video", IEEE Trans. on Multimedia, September 1999.
 [12] <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