

MPEG-4 비디오 스트림을 위한 실시간 스케줄링 기법

김진환*, 정인환**, 송용호***, 이정훈****

*한성대학교 멀티미디어공학과 (kimjh@hansung.ac.kr)

**한성대학교 컴퓨터공학과(ihjung@hansung.ac.kr)

**한양대학교 정보통신대학(yhsong@hanyang.ac.kr)

***제주대학교 전산통계학과 이정훈(jhlee@cheju.ac.kr)

1. 서론

최근 MPEG-4는 64Kbps의 저속 통신망부터 4Mbps급의 고속통신망에 이르기까지 다양한 화질의 비디오 스트리밍 응용에 사용되고 있다. 비디오를 위한 서비스 품질의 요건으로 fps 즉 초당 프레임 수가 필요하며 이러한 요건을 충족하기 위하여 실시간 스케줄링 기법이 반드시 필요하다. 범용 운영체제는 멀티미디어 태스크를 위한 실시간 서비스를 제공하지 않기 때문에 비디오 재생시 jitter 현상이 발생하며 상호작용성이 저하되는 문제가 발생한다. MPEG-4 비디오 스트림을 수신하는 단말기는 휴대용 전화기 또는 PDA 등이며 자원이 제한된 환경을 가지고 있다. 이러한 단말기는 시스템에 유입되는 멀티미디어 데이터를 시간적으로 적절하게 처리할 수 있도록 다양한 비트율을 수용할 수 있어야 한다. 비디오 스트림 재생시 심각한 품질의 저하가 발생하지 않도록 하려면 충분한 컴퓨팅 자원을 디코딩 태스크에 할당할 수 있는 실시간 스케줄러가 필요하다. 비디오 스트림을 디코딩하는 멀티미디어 태스크는 약한 시간적 제약성을 가지므로 연성 실시간적 태스크로 간주된다. 본 논문에서는 멀티미디어 태스크들의 실시간 성능을 최대화하고자 태스크의 평균 실행시간을 고려한 후 CPU 대역폭의 할당을 효율적으로 제어하는 스케줄링 기법을 제시하고 있다.

2. 요약

MPEG-4는 오디오-비디오 객체의 시간적 인스턴스인 VOP(Video Object Plane)를 세가지로 분류한다. 즉 기준이 되는 I(ntra picture)-VOP, P(redicted picture)-VOP와 이를 참조하는 B(idirectional predicted picture)-VOP 등이 있으며 I-VOP부터 다음 I-VOP 전까지의 VOP들을 한 GOV(Group of VOP)로 구성한다. 이는

MPEG-2의 GOP(Group of Pictures)와 동일한 구성방식으로 간주된다. 한 GOV내에서 I-VOP가 가장 먼저 디코딩되어야 하며 이후 P-VOP와 B-VOP가 차례로 디코딩되어야 한다. 특히 CBR 방식보다 압축 효율이 우수한 VBR 방식을 MPEG-4 비디오 스트림에 적용할 경우 각 VOP마다 압축률이 달라지며 P-VOP나 B-VOP보다 I-VOP를 디코딩할 때 더 많은 시스템 자원이 필요하게 된다.

각 VOP의 데이터 크기와 디코딩 순서를 고려하여 본 논문에서는 I-VOP를 디코딩하는 태스크의 클래스를 수준이 가장 높은 1번 클래스로 설정하며 P-VOP와 B-VOP를 디코딩하는 태스크의 클래스를 2번 클래스와 3번 클래스로 각각 설정한다. 본 논문의 스케줄러는 클래스 번호가 작은 태스크를 먼저 스케줄링하는 정책을 가진다. 그리고 동일한 클래스에 있는 태스크들이 다수일 경우 각 태스크마다 설정된 종료시한(deadline)이 빠를수록 높은 우선순위를 부여함으로써 먼저 스케줄링되도록 한다. 각 VOP를 디코딩하는 태스크 T_i 는 평균실행시간 C_i 와 주기 P_i 가 설정되며 태스크가 시스템에 도착한 시간에 주기를 더한 시간이 태스크의 종료시한으로 설정된다. 다수의 태스크들이 존재할 경우 주기가 가장 작은 태스크를 서버의 주기로 설정하며 서버의 주기와 최대실행시간은 각각 P_s 와 Q_s 로 정의한다. 태스크 T_i 는 자신의 주기와 서버의 주기를 고려하여 서버의 주기내에서 $C_i * P_s / P_i$ 실행시간이 설정된다. 태스크의 수가 n 일 경우 Q_s 는 다음과 같이 설정된다.

$$Q_s = \sum_{i=1}^n C_i \frac{P_s}{P_i}$$

그리고 서버는 1번 클래스부터 태스크들중 종료시한이 가장 빠른 태스크를 선정하여 CPU를 사용할 수 있도록 한다. 본 논문에서는 각 태스크

의 평균실행시간이 예상시간인 점을 감안하여 서버의 주기내에 설정된 자신의 실행시간을 초과할 경우에도 해당 태스크가 CPU를 계속 사용할 수 있도록 허용함으로써 해당 VOP의 디코딩 작업에 대한 연속성을 보장하였다.

3. 실험 결과

제시된 스케줄링 방법을 VB(Variable Bandwidth) 기법으로 정의하며 태스크에 우선순위를 적용하지 않는 기존의 CBR(Constant Bandwidth Reservation) 기법에 기초한 FB(Fixed Bandwidth)기법과 성능을 비교 분석하였다. I-VOP, P-VOP, B-VOP의 평균 디코딩 시간을 각각 16.2ms, 7.7ms, 4.3ms로 가정하였으며 주기가 30ms이고 평균실행시간이 6ms인 다섯개의 태스크를 12000ms동안 생성하여 실험하였다. 디코딩이 수행된 총 VOP의 수는 2001개이며 표 1과 같이 종료시한내에 디코딩이 완료된 VOP의 수(N_{comp})는 VB 기법이 724개이며 FB 기법이 381개으로써 VB 기법이 종료시한내에 373개 만큼의 더 많은 VOP를 디코딩한 것을 의미한다. 디코딩이 완료된 태스크의 수를 전체 VOP 수 2001로 나눈 성공률은 VB 기법이 FB 기법보다 18.64% 증가된 성공률을 나타내고 있다. 본 논문에서는 종료시한내에 디코딩이 완료된 VOP의 수를 최대화함으로써 멀티미디어 시스템의 실시간적 성능을 향상시키고자 하였다.

FB 기법은 항상 10%이하의 성공률을 나타내는 반면 VB 기법은 U가 증가할수록 성공률이 낮아지며 FB 기법보다 높은 성공률을 보이고 있다. 결과적으로 VB 기법이 FB 기법보다 우수한 실시간적 성능을 가지고 있는 것으로 분석된다.

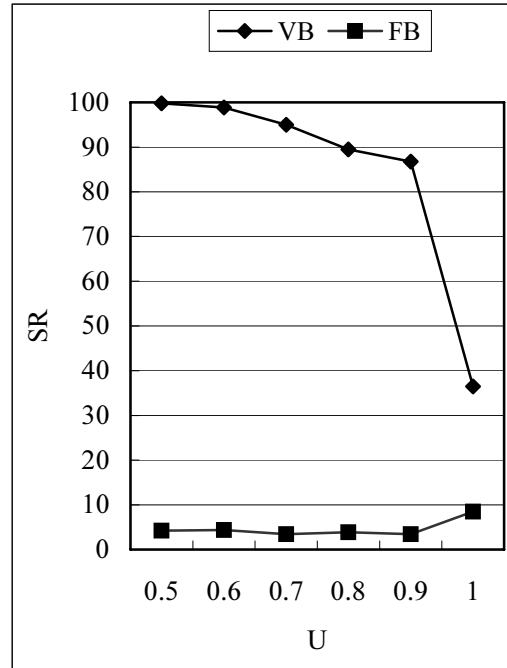


그림 1. 자원 활용률에 따른 성공률

표 1. 디코딩이 완료된 VOP 수

스케줄링 기법	VB	FB
N_{comp}	724	381
성공률	36.18%	17.54%

그림 1에서 VB 기법과 FB 기법의 성공률 SR을 U가 0.5에서 1.0까지 증가하는 동안 비교하였다. CPU 자원의 활용율을 의미하는 U는 다음과 같이 정의된다.

$$U = \sum_{i=1}^n \frac{C_i}{P_i}$$