

모바일 환경에서 MPEG-4 비디오 스트림 전송을 위한 실시간 적응형 스케줄링 프로토콜

김진환[†]

요약

모바일 환경의 비디오 스트리밍에서 클라이언트들은 작업 부하의 차이가 큰 비디오를 요청할 수 있기 때문에 적응성은 중요한 문제이다. 본 논문에서는 모바일 또는 무선 통신망에서 MPEG-4 비디오 스트림을 효율적으로 전송하기 위해 제한된 대역폭을 스케줄링하는 기법이 제시된다. 수락 제어 단계에서는 요청된 각 비디오 대역폭이 평균 대역폭 요건에 따라 할당된다. 실제 스케줄링 단계에서는 클라이언트들의 재생 상황에 더욱 적합한 스케줄링이 수행될 수 있도록 하기 위해 클라이언트의 버퍼 수준에 따라 대역폭이 동적으로 할당된다. 제시된 RTA(Real-Time Adaptive) 스케줄링 프로토콜에서는 버퍼 수준이 낮은 클라이언트에게 일시적으로 더 많은 대역폭이 할당될 수 있다. 버퍼 수준에 기반한 스케줄링 정책을 사용하는 RTA 프로토콜은 일시적인 과부하 영향을 최소화하는 한편 재생시 실시간적 성능을 최대화할 수 있다. RTA 프로토콜의 성능은 다양한 실험을 통하여 측정되었으며 BSBA(Buffer Sensitive Bandwidth Allocation) 프로토콜과 비교한 결과 RTA 프로토콜이 더 많은 수의 프레임들을 전송함으로써 우수한 성능을 가지는 것으로 분석되었다.

A Real-time Adaptive Scheduling Protocol for MPEG-4 Video Stream Transmission in Mobile Environment

Jinhwan Kim[†]

ABSTRACT

Adaptability is an important issue in video streaming over mobile environments, since the clients may request videos with great differences in their workload. In this paper, we propose the issues in limited bandwidth scheduling for efficient MPEG-4 video stream transmission over a mobile or wireless network. In the phase of admission control, the amount of bandwidth allocated to serve a video request is the mean bandwidth requirement of its requested video. The dynamic allocation of bandwidth in the phase of scheduling depends on the playback buffer levels of the clients with an objective to make it more adaptive to the playback situation of individual clients. In the proposed RTA scheduling protocol, more bandwidth may be allocated temporarily to the client whose buffer level is low. By employing the buffer level based scheduling policy, this protocol attempts to maximize the real-time performance of individual playback while minimizing the impact of transient overloading. Extensive simulation experiments have been performed to investigate the performance characteristics of the RTA protocol as comparing with BSBA protocol. This RTA protocol shows the better performance by transferring more frames than BSBA protocol. Computer simulations reveals that the standard deviation of the bit rate error of the proposed scheme is 50% less than that of the conventional method.

Key words: adaptive(적응형), real-time scheduling(실시간 스케줄링), MPEG-4 video(MPEG-4 비디오), bandwidth(대역폭), buffer level(버퍼 수준)

※ 교신저자(Corresponding Author) : 김진환, 주소 : 서울 성북구 삼선동3가 389(136-792), 전화 : 02)760-4340, FAX : 02)760-4488, E-mail : kimjh@hansung.ac.kr
접수일 : 2009년 4월 9일, 수정일 : 2009년 9월 1일

완료일 : 2009년 12월 14일

[†] 정회원, 한성대학교 멀티미디어공학과 교수

※ 본 연구는 2009년도 한성대학교 교내연구비 지원에 의하여 수행되었음

1. 서 론

최근에 디지털 카메라와 같은 영상기기의 화소수의 확대로 인하여 많은 양의 저장매체를 요구하고 있다. 예를 들어 2592×1944 크기의 사진 한 장을 RAW Format으로 저장하려면 $2592 \times 1944 \times 3 = 15116544$, 15Mbyte 저장매체를 요한다. 유선 통신망을 기반으로 하는 분산 비디오 재생장치 시스템에서 제시된 비디오 프레임 버퍼링, feedback 제어 메카니즘, 비디오 스트림 smoothing 기법[1-3] 등은 모바일 기기와 모바일 통신망의 특성으로 인하여 모바일 비디오 재생장치 시스템에 동일하게 적용하기에는 어려운 면이 있다[4]. 모바일 통신망의 비대칭적 대역폭은 모바일 클라이언트에서 서버로 전달하는 feedback 메시지의 양을 제한하며 비디오 스트림 전송을 위한 흐름-제어 메카니즘을 어렵게 하는 것이 주요인으로 분석되고 있다. 또한 클라이언트들의 이동성은 시스템내 작업 부하와 비디오 재생시 품질 유지를 위한 수락 제어(admission control) 메카니즘의 효율성에 심각한 영향을 미칠 수 있다. 즉 불안정한 통신망 환경에서는 재생 품질을 보장하기가 어렵게 된다. 모바일 통신망의 대역폭이 개선되고 있긴 하나 고품질 비디오에 대한 요구가 증가하고 있다. 화면의 해상도와 크기를 비롯한 작업부하가 다양한 비디오 요청들을 서비스하기 위하여 모바일 통신망의 대역폭을 효율적으로 할당하는 방법은 모바일 비디오 재생장치 시스템의 성능에 중대한 요소로 작용하게 된다[5].

클라이언트의 버퍼 수준을 감안한 후 요청된 비디오 스트림에 모바일 통신망의 대역폭을 할당하는 기존의 스케줄링 기법들은[4,6] 동적으로 변하는 비디오 트래픽에 효과적으로 대처하기 어려운 면이 있으며 또한 반드시 여분으로 사용할 수 있는 대역폭이 존재할 때만 이 기법들을 적용할 수 있는 문제점이 있다. 휴대폰이나 PDA 등 모바일 단말기를 이용하는 클라이언트의 버퍼는 노트북이나 데스크탑 컴퓨터에 비하면 아직 크기가 작은 편이다. 이미 MPEG-4[7]를 기반으로 하는 영상 회의(video conferencing) 또는 VOD(video on demand) 등의 멀티미디어 응용은 모바일 또는 무선 통신망 기반의 클라이언트-서버 시스템을 활용하여 구축되고 있다[8-10].

본 논문은 모바일 통신망에서 MPEG-4 비디오 스트림을 효율적으로 전송하기 위한 실시간 적응형 스

케줄링 기법을 제시한다. 다수의 모바일 클라이언트들이 요청한 작업을 서비스하기 위하여 서버는 MPEG 표준으로 압축된 비디오 프레임을 패킷으로 구성하여 전송하게 된다. MPEG 압축 기법의 VBR (Variable Bit Rate) 특성상[11] 각 프레임은 종류에 따라 상이한 크기를 갖게 되므로 비디오 트래픽의 변동성과 일시적인 과부하 현상을 유발할 수 있다 [12]. 본 논문의 스케줄링 프로토콜에서는 각 클라이언트가 요청한 비디오를 서버가 전송할 때 발생하는 일시적 과부하 현상의 영향을 최소화하여 전송 중단되는 프레임 수를 감소시킴으로써 비디오 재생에 관한 실시간적 성능과 품질을 향상시키고자 한다. 제시된 스케줄링 프로토콜은 수락 제어 단계에서는 각 클라이언트가 요청한 비디오의 평균 프레임 크기에 따라 모바일 통신망의 대역폭이 정적으로 할당된다. 그러나 실제 스케줄링 단계에서는 비디오 트래픽의 변동성과 일시적 과부하로 인하여 할당된 대역폭보다 큰 대역폭이 필요할 수 있다. 이러한 경우 서버는 각 클라이언트의 버퍼 수준에 따라 동적으로 대역폭을 할당하게 된다. 즉 버퍼 수준이 낮은 클라이언트인 경우 할당된 대역폭보다 큰 대역폭이 일시적으로 할당됨으로써 종료시한 내에 전송과 재생이 중단되는 프레임의 수를 감소시킬 수 있게 된다. 그리고 버퍼 수준이 높은 클라이언트인 경우에는 버퍼에 있는 현재 프레임들을 재생한 후 다음 주기에 서버로부터 새로운 프레임들을 수신함으로써 비디오 재생의 연속성을 유지하게 된다.

본 논문은 2장에서 모바일 통신망 기반의 분산 모바일 재생장치 시스템 특성이 기술되며 3장에서 MPEG-4 비디오 스트림을 패킷으로 전송하기 위한 대역폭 할당과 버퍼 수준에 기반한 실시간 적응형 스케줄링 알고리즘을 기술한다. 그리고 4장에서 다른 스케줄링 기법들과 비교된 성능을 분석하며 5장에서는 결론을 기술한다.

2. 분산 모바일 재생장치 시스템

본 논문에서는 모바일 통신망을 기반으로 하는 분산 모바일 비디오 재생장치 시스템을 고려한다(그림 1 참조). 모바일 통신망에서는 기지국과 모바일 클라이언트들이 cell 단위를 기반으로 연결되며 기지국은 고속 통신망을 통하여 비디오 서버와 연결된다. 모바

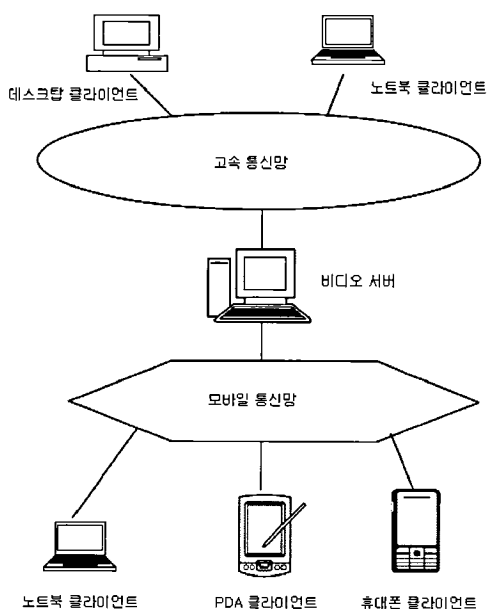


그림 1. 모바일 통신망 기반 분산 모바일 재생장치 시스템

일 통신망 대역폭은 고속 통신망의 대역폭에 비하면 크기가 매우 작기 때문에 병목 현상이 있는 자원으로 가정된다[13]. 본 논문에서는 제한된 모바일 통신망의 대역폭을 효과적으로 사용함으로써 모바일 클라이언트의 요청을 서비스하는 효율적인 스케줄링 프로토콜을 비디오 서버에 구성하고자 한다.

비디오 서버는 MPEG-4 표준으로 압축된 비디오를 가지고 있으며 각 비디오 스트림은 GOP(Group of Picture) 시퀀스로 구성되어 있다. 각 GOP는 I, P, B 프레임들로 구성되며 통상 I 프레임의 크기는 B 프레임이나 P 프레임보다 훨씬 크다. GOP마다 크기가 상이하며 비디오의 고정된 재생율 fps(frames per second)에 따라 각 비디오 스트림의 대역폭 요건은 가변적 특성을 가지게 된다.

모바일 클라이언트들은 기지국을 통하여 비디오 서버에 비디오 재생을 요청하게 된다. PDA, 휴대폰, 포켓 PC 등을 사용하는 thin 클라이언트들은 노트북 컴퓨터를 사용하는 thick 클라이언트에 비하여 비디오 버퍼 크기가 수십 kbyte로 정도로 제한이 있게 된다[14]. 클라이언트들마다 작업 부하의 크기가 상이한 비디오를 요청하게 되며 예를 들어 고성능 PC를 이용하는 클라이언트의 비디오 작업 부하량은 포켓 PC에 비하여 훨씬 커지게 된다.

모바일 클라이언트는 서버로부터 수신한 프레임

들을 자신의 버퍼에 저장하며 버퍼 수준이 사전에 정의된 수치에 도달하면 비디오의 재생을 시작하게 된다. 즉 복원 과정이 시작되는 것이며 실제 비디오의 재생은 복원된 프레임을 비디오 재생장치에 한 개씩 전송함으로써 수행된다. 그러나 비디오 프레임의 재생 시간을 놓치게 되면 해당 프레임은 즉시 재생이 취소되며 폐기된다. 이러한 현상은 seamless video streaming 유지에 심각한 영향을 미치게 되며 재생율에 관한 서비스 품질과 실시간적 성능도 저하시키게 된다[5].

서버가 다수의 클라이언트로부터 비디오 요청을 수신할 때 수락 제어 과정을 실시하게 되며 다수의 비디오 요청 작업을 동시에 서비스할 수 있다. 서버가 비디오 요청의 수락을 허용한 경우에는 해당 비디오 파일을 저장 장치에서 검색하며 비디오 프레임들을 요청별로 지정된 비디오 버퍼에 저장하게 된다. 이 비디오 프레임들은 패킷으로 구성되며[15] 기지국과 모바일 통신망을 통하여 해당 모바일 클라이언트에게 전송된다. 비디오 패킷 전송시 통신 오류가 발생할 수 있으며 재전송 과정이 필요하게 된다. 본 논문에서는 모바일 통신망의 오류 발생에 대한 구체적인 기술은 생략한다.

모바일 비디오 재생장치 시스템은 유선 통신망에 기반한 분산 비디오 재생장치 시스템과 비교할 때 여러 가지 차이점을 가지고 있다. 우선 모바일 통신망과 무선 LAN의 대역폭이 점차 개선되고 있긴 하나 아직도 유선 통신망의 대역폭 크기에 비하면 작다. 또한 전송시 오류 발생 확률이 크고 통신망 내에서의 단절 현상이 빈번하며 모바일 통신망의 cell 단위에서는 상향 속도와 하향 속도가 비대칭적 특성을 가지고 있다. 모바일 통신망에서 jitter 현상을 최소화하기 위해 클라이언트에 버퍼 기법이 필요하나 thin 클라이언트 경우 thick 클라이언트에 비하면 버퍼 용량이 충분하지 않다. 결국 버퍼 크기가 작은 클라이언트일수록 버퍼가 큰 클라이언트에 비하여 통신망 오류에 민감하게 되며 비디오 재생시 과부하에 따른 성능 저하가 발생하게 된다[5]. 고화질 비디오 재생을 위한 비디오의 가변적 비트율은 모바일 멀티미디어 시스템에서 해결하기 더욱 어려운 문제로 간주된다. 그리고 클라이언트마다 다른 해상도와 화질을 요구함에 따라 작업부하의 크기가 달라지고 일시적인 과부하가 자주 발생하므로 이를 공평하고 효율

적으로 해결할 수 있는 기법들이 필요하게 된다.

3. 버퍼 수준 기반 실시간 적응형 스케줄링 프로토콜

3.1 수락 제어

모바일 통신망의 비디오 서버는 클라이언트가 요청한 MPEG-4 비디오 스트림을 전송하기 전 수락 제어(admission control) 절차를 수행한다. cell 단위의 모바일 통신망에서 기지국과 클라이언트들 간에 사용가능한 최대 대역폭 BW_{total} 중 오류 발생시 재전송을 위한 대역폭과 비디오 스트림 전송 이외의 목적으로 사용되는 대역폭을 제외한 사용 가능한 대역폭을 수식 1에서 BW_{avail} 로 정의한다.

$$BW_{avail} \leq BW_{total} \quad \text{수식 (1)}$$

클라이언트들이 요청한 비디오 스트림 n 개의 대역폭 합을 BW_{sum} 이라 할때 이는 BW_{avail} 보다 작거나 같아야 하므로 수식 2는 다음과 같이 정의된다.

$$\begin{aligned} BW_{sum} &\leq BW_{avail} \\ BW_{sum} &= BW_1 + BW_2 + \dots + BW_n \quad (n \geq 1) \\ &= \sum_{i=1}^n BW_i \quad \text{수식 (2)} \end{aligned}$$

BW_1, BW_2, \dots, BW_n 등은 n 개의 비디오 스트림의 평균 대역폭을 각각 의미한다. 새로운 클라이언트가 요청한 $n+1$ 번째 비디오 스트림의 평균 대역폭 BW_{n+1} 이 추가된 대역폭의 합 BW_{sum} 이 수식 2의 관계를 만족하면 서버가 이를 수용하지만 그렇지 않은 경우에는 이를 거부할 수 있다. 그리고 거부된 비디오 스트림은 나중에 서버가 재수용할 수 있으나 이에 관한 구체적인 과정은 본 논문에서 기술하지 않는다.

$i(1 \leq i \leq n)$ 번째 비디오 스트림의 평균 대역폭 BW_i 는 해당 클라이언트의 재생율과 프레임의 평균 크기에 의해 결정된다. 예를 들어 재생율이 25이고 프레임의 평균 크기가 1000 바이트인 경우 BW_i 는 다음과 같이 계산된다.

$$\begin{aligned} BW_i &= 25 \text{프레임/초} \times 1000 \text{바이트/프레임} \times 8 \text{비트/바이트} \\ &= 200,000 \text{비트/초} \\ &\approx 0.2 \text{Mbps} \end{aligned}$$

3.2 버퍼 수준 기반 스케줄링

수락 제어 이후 각 비디오 스트림의 실제 대역폭

은 VBR 특성[11]에 따라 크기가 가변적이므로 BW_{sum} 은 BW_{avail} 을 초과하는 경우가 발생할 수 있다. 이 경우 BW_{avail} 을 초과하는 각 비디오 스트림의 프레임들은 전송이 불가능하기 때문에 서버에 의해 폐기되는 것이 원칙이다. 그러나 전송이 중단되는 프레임이 많아질수록 해당 클라이언트는 seamless video streaming 유지가 어렵게 되고 재생율 및 서비스 품질(QoS; Quality of Service)이 저하되는 현상이 발생한다.

본 논문에서는 이러한 현상을 최소화시키고자 각 클라이언트의 버퍼 수준을 고려하여 서버가 대역폭을 동적으로 할당한다. 각 클라이언트는 휴대폰 또는 PDA 등(이하 단말기로 기술함)에 설치된 버퍼에 재생율을 고려한 일정 시간 동안 프레임들이 서버로부터 도달하면 이때부터 버퍼에 있는 프레임은 디코딩하여 재생장치로 전달하게 된다. 따라서 버퍼 수준이란 클라이언트의 단말기가 일정 시간 동안 전송된 프레임들을 저장할 수 있는 버퍼의 크기를 의미한다. 클라이언트의 버퍼 수준이 높을수록 실제 버퍼의 크기가 큰 것을 의미하는 것이다. 모바일 클라이언트는 단말기의 특성상 기억장치의 크기가 상이하므로 실제 버퍼의 수준도 상이하게 된다. 버퍼 수준이 높을수록 오랜 시간 동안 재생할 수 있는 많은 수의 프레임들을 버퍼에 유지하게 되며 통신망의 일시적 장애나 오류 발생시에도 덜 민감하게 반응할 수 있는 장점이 있다[9]. 그러나 버퍼 수준이 높을 경우 비디오 재생이 시작되는 시간이 늦어지고 기억장치 활용도가 저하되는 문제가 있으며 모바일 단말기에 대용량의 버퍼를 유지하기 어려운 면이 있다.

그림 2에서 기술된 바와 같이 서버는 단말기 버퍼 수준을 고려하여 전송하고자 하는 실제 대역폭의 합이 사용가능한 통신망 대역폭의 합보다 클 경우 버퍼 수준에 여유가 있는 클라이언트의 비디오 스트림은 가급적 다음번에 전송함으로써 전송이 중단되는 프레임들의 수를 최소화하고 버퍼 수준에 여유가 없는 비디오 스트림일수록 전송되는 프레임의 수를 최대화하는 것이다.

서버가 일단 전송을 실시한 비디오 스트림에 대한 단말기 버퍼 수준은 초기값으로 다시 설정함으로써 다음 주기의 스케줄링 과정에서 단말기 버퍼 수준의 여유를 확보하게 된다. 단 그림 2의 2-2-1처럼 현재의 버퍼 수준을 고려하여 조정된 대역폭의 합이 모두

1. B_{sum} 이 BW_{avail} 이하이면 현재 대역폭을 전송하고 버퍼 수준을 초기화
2. B_{sum} 이 BW_{avail} 을 초과하면 버퍼 수준의 여유를 점검
 - 2-1. 버퍼 수준의 여유가 없으면 현재 대역폭에서 전송 중단할 대역폭을 차감 후 전송하고 버퍼 수준을 초기화
 - 2-2. 버퍼 수준의 여유가 있으면 현재 대역폭에서 다음에 전송할 대역폭을 차감한 조정된 대역폭을 설정하고 버퍼 수준을 1씩 감소시킴
 - 2-2-1. 조정된 대역폭의 합이 BW_{avail} 이하이면 조정된 대역폭을 전송하고 여유가 없는 버퍼 수준만 초기화
 - 2-2-2. 조정된 대역폭의 합이 BW_{avail} 을 초과하면 조정된 대역폭에서 전송 중단할 대역폭을 차감한 후 전송하며 버퍼 수준을 초기화

그림 2. 단말기 버퍼 수준에 기반한 스케줄링 정책

전송되고도 버퍼 수준에 아직 여유가 있는 경우에는 해당 클라이언트의 버퍼 수준을 초기화하지 않는다. 그림 2의 2-1과 2-2-2에서 전송이 중단되는 프레임들의 경우 중요도가 가장 낮은 B 프레임들이 먼저 중단되며 이후 P 프레임, I 프레임 순으로 결정되는 우선순위 정책이 본 논문에서 적용된다.

3.3 실시간 적응형 스케줄링 알고리즘

i번째 비디오 스트림을 위한 단말기의 버퍼 수준을 BL_i 라 하며 시간 단위인 초를 이용하여 크기를 설정한다. 본 논문에서 모든 비디오 스트림의 버퍼 수준은 1보다 크거나 같은 값을 가지는 것으로 가정한다. 단말기의 버퍼 수준에서 숫자의 의미는 비디오 스트림의 프레임들을 클라이언트 단말기 버퍼에 저장할 수 있는 시간을 초 단위로 표현한 것이다. 따라서 단말기의 버퍼 수준이 1이라 하더라도 비디오 스트림에 따라 1초간 저장되는 프레임들의 평균 크기가 다를 수 있으므로 단말기 버퍼의 실제 크기는 상이할 수 있다.

본 논문에서는 모든 단말기는 임의의 비디오 스트림을 최소 1초간 저장할 수 있는 정도의 버퍼 크기를 가지고 있는 것으로 가정한다. 서버는 각 스트림의 버퍼 수준을 일정 주기(본 논문에서는 1초로 가정함)마다 파악하며 버퍼 수준의 합 BL은 다음과 같이 정의된다.

$$BL = \sum_{i=1}^n BL_i \quad \text{수식 (3)}$$

비디오 서버는 매초마다 각 스트림이 요구한 프레임들을 저장장치에서 검색 후 서버의 버퍼에 저장하며 이 크기는 BW_{sum} 이 아닌 B_{sum} 으로 기술한다. 일정 주기마다 i번째 스트림이 요청한 실제 프레임들의 크기를 비트로 표시한 B_i 와 B_{sum} 은 수식 4의 관계로 성립된다.

$$B_{sum} = \sum_{i=1}^n B_i \quad \text{수식 (4)}$$

그림 3은 매초마다 비디오 서버가 수행하는 실시간 적응형 스케줄링 알고리즘을 의사 언어로 기술하고 있다. 각 스트림이 요구한 프레임들의 합인 B_{sum} 이 BW_{avail} 보다 작거나 같을 경우에는(알고리즘의 1행) 서버가 검색한 버퍼내의 모든 프레임들을 전송할 수 있기 때문에 i번째 스트림의 프레임들은 해당 클라이언트 단말기로 전송되며(5행에서 $Transfer(B_i)$ 로 표현함) 버퍼 수준 BL_i 는 초기값으로 다시 설정된다(6행에서 $Initialize(BL_i)$ 로 표현함). 그러나 B_{sum} 이 BW_{avail} 을 초과할 경우(9행)에는 현재 버퍼 수준의 합 BL을 점검한다. BL이 n보다 작거나 같을 경우에는(11행) 단말기 버퍼에 여유가 없음을 의미한다.

따라서 i번째 스트림의 실제 대역폭 B_i 가 평균 대역폭 BW_i 를 초과할 경우(15행) 이에 대한 합 $\sum_{i=1}^n (B_i - BW_i)$ 을 먼저 구하게 된다. 이후 자신의 초과분 $B_i - BW_i$ 를 이 합으로 나눈 값을 BW_{avail} 을 초과한 대역폭인 $B_{sum} - BW_{avail}$ 과 곱하면 전송을 포기해야 하는 대역폭이 결정된다(16행). 이 대역폭을 알고리즘에서 $Bdrop_i$ 로 표현한다. 본 알고리즘에서는 전송을 중단해야 하는 대역폭은 평균 대역폭을 초과한 크기에 비례하여 결정하게 된다. 그러나 B_i 가 평균 대역폭 BW_i 이하일 경우에는(17행) $Bdrop_i$ 가 0으로 설정된다. 전송이 중단되는 프레임들의 경우 우선순위 정책이 적용된다. B_i 에서 $Bdrop_i$ 를 차감한 대역폭이 해당 클라이언트 단말기로 전송되며(19행) 버퍼 수준에 여유가 더 이상 없기 때문에 BL_i 도 다시 초기화된다(20행).

BL이 n보다 크면(23행) 단말기 버퍼에 여유가 없음을 의미한다. 따라서 1보다 큰 BL_i 를 가진 스트림에(27행) 대해서는 BW_{avail} 을 초과한 대역폭($B_{sum} - BW_{avail}$)을 단말기 버퍼 수준의 여유분에 비례하여 다음 주기에 전송할 수 있도록 서버의 버퍼에 남겨두

```

1: if (Bsum ≤ BWavail)
2:   begin
3:     for i=1 to n do
4:       begin
5:         Transfer(Bi);
6:         Initialize(BLi);
7:       end
8:     end
9:   else // Bsum > BWavail
10:  begin
11:    if (BL ≤ n)
12:      begin
13:        for i=1 to n do
14:          begin
15:            if (Bi > BWi)
16:              Bdropi=(Bsum-BWavail)*  $\frac{B_i - BW_i}{\sum_{i=1}^n (B_i - BW_i)}$ 
17:            else
18:              Bdropi=0;
19:            Transfer(Bi-Bdropi);
20:            Initialize(BLi);
21:          end
22:        end
23:      else // BL > n
24:        begin
25:          for i=1 to n do
26:            begin
27:              if (BLi > 1)
28:                begin
29:                  Boveri=(Bsum-BWavail)*  $\frac{BL_i - 1}{BL - n}$ 
30:
31:                  B'i=Bi-Boveri;
32:                  BLi=BLi - 1;
33:                end
34:              else
35:                B'i=Bi;
36:            end // for i=1 to n
37:
38:            B'sum=  $\sum_{i=1}^n B'_i$ 
39:            B'sum 처리 알고리즘(그림 4 참조)
40:          end // else BL > n
41:        end // else Bsum > BWavail

```

그림 3. 버퍼 수준에 기반한 실시간 스케줄링 알고리즘

는 대역폭을 Bover_i로 기술한다(29행). BL-n은 여유가 있는 단말기 버퍼 수준의 합을 의미하며 BL_i-1은 i번째 스트림의 버퍼 여유 수준을 의미하게 된다. 이때 원래 필요한 대역폭 B_i에서 Bover_i를 차감한 후

```

1: if ( B'sum ≤ BWavail )
2:   begin
3:     for i=1 to n do
4:       begin
5:         Transfer(B'i);
6:         if (BLi ≤ 1)
7:           Initialize(BLi);
8:         end
9:       end
10:  else // B'sum > BWavail
11:  begin
12:    for i=1 to n do
13:      begin
14:        if (B'i > BWi)
15:          B'dropi=(B'sum-BWavail)*  $\frac{B'_i - BW_i}{\sum_{i=1}^n (B'_i - BW_i)}$ 
16:        else
17:          B'dropi=0;
18:        Transfer(B'i-B'dropi);
19:        Initialize(BLi);
20:      end // for i=1 to n
21:    end // else B'sum > BWavail

```

그림 4. B'_{sum} 처리에 대한 스케줄링 알고리즘

전송하려는 조정된 대역폭이 B'_i(31행)이며 BL_i는 1만큼 차감된다(32행). BL_i가 1이하인 경우(34행) B'_i는 B_i와 동일하다(35행). BL_i가 차감된 클라이언트 단말기에서는 서버로부터 새로운 프레임들을 수신하는 대신 단말기 버퍼에 있는 현재 프레임들을 디코딩하여 재생하게 된다. n 개의 스트림에 대해 조정된 대역폭 B'_i의 합인 B'_{sum}(37행)에 대한 스케줄링 과정은 그림 4에서 기술된다.

서버가 단말기 버퍼 수준을 고려하여 설정한 B'_{sum}이 BW_{avail}보다 작거나 같을 경우(그림 4의 1행, 이하 행 번호는 모두 그림 4의 행 번호를 의미함)에는 모든 B'_i가 전송되며(5행) BL_i가 최소값 1보다 작거나 같은 경우(6행)에만 다시 초기화된다(7행). 즉 BL_i가 1보다 큰 경우에는 아직 버퍼 수준에 여유가 있기 때문에 현재값을 그대로 유지함으로써 다음 주기의 스케줄링에서 해당 클라이언트 단말기의 버퍼 수준이 적용될 수 있도록 하는 것이다. 그러나 B'_{sum}이 BW_{avail}보다 여전히 클 경우(10행)에는 그림 3의 16행과 유사한 방법으로 평균 대역폭 BW_i를 초과한 B'_i의(14행) 합 $\sum_{i=1}^n (B'_i - BW_i)$ 을 먼저 구하게 된다.

이후 i번째 스트림의 초과분 B'_i-BW_i를 이 합으로

나는 값과 전체 초과분인 $B'_{sum} - BW_{avail}$ 을 곱하여 전송을 포기해야 하는 대역폭 B'_{drop_i} 를 결정하게 된다 (15행). B'_i 가 BW_i 이하일 경우(16행)에는 B'_{drop_i} 가 0이 된다(17행). 서버는 B'_i 에서 B'_{drop_i} 를 차감한 대역폭을 전송하며(18행) 버퍼 수준 BL_i 를 다시 초기화한다(19행). 그리고 그림 3의 16행과 동일하게 서버는 전송을 중단해야 하는 프레임들에 대해서는 우선 순위 정책을 적용하여 B, P, I 프레임 순서로 중단시킨다.

4. 성능 분석 및 평가

4.1 여분의 대역폭이 없는 경우의 프레임 전송 중단 비율

본 논문에서는 압축률이 우수한 VBR(Variable Bit-Rate) 방식으로 부호화된 MPEG-4 비디오 스트림 “Jurassic Park One”[16]을 대상으로 실험을 수행하였다. 이 비디오 스트림은 QCIF 형식이며 GOP내 프레임 수는 12이고 GOP 패턴은 IBBPBBPBBPBB이다. 총 108000개의 프레임 크기를 분석한 결과 프레임의 최소값은 8바이트, 최대값은 2903바이트이며 평균값은 276.872바이트로 파악되었다. 프레임의 최대값을 평균값으로 나눈 비율은 약 10.49로서 이 비율이 13.36인 “Silence of the Lambs”와 17.29인 “Star Wars IV”에 비하여 다소 작은 편이다. 이 비율이 클수록 비디오 스트림 전송시 평균 대역폭의 범위를 초과하는 기회가 많아져서 전송 중단되는 프레임의 비율도 커질 것으로 사료된 바 본 논문에서는 “Jurassic Park One”을 대상으로 실험을 수행하였다. 각 클라이언트가 요청하는 이 비디오 스트림의 재생율을 24로 동일하게 적용할 경우 i 번째 비디오 스트림에 필요한 평균 대역폭 BW_i 는 다음과 같이 설정된다.

$$\begin{aligned}
 BW_i &= 8\text{바이트/바이트} \\
 &\quad \times 276.872\text{바이트/프레임} \times 24 \text{ 프레임/초} \\
 &= 53159.424 \text{ 비트/초} \\
 &\approx 53.16\text{kbps}
 \end{aligned}$$

서버는 동일한 비디오 스트림을 동일한 재생율로 20개의 모바일 클라이언트에게 전송한다. 20개의 스트림의 평균 대역폭 $BW_i(1 \leq i \leq 20)$ 의 합 BW_{sum} 은 다음과 같다.

$$\begin{aligned}
 BW_{sum} &= \sum_{i=1}^{20} BW_i \\
 &\approx 53.16\text{kbps} \times 20 \\
 &\approx 1.0632\text{Mbps}
 \end{aligned}$$

모바일 통신망의 최대 대역폭 BW_{total} 을 2Mbps로 가정하며 비디오 스트림 전송에 사용되는 평균 대역폭의 합 BW_{sum} 과 BW_{avail} 은 동일한 것으로 가정할 때 다음 관계가 성립된다.

$$BW_{sum} \leq BW_{avail} \leq BW_{total}(=2\text{Mbps})$$

즉 본 논문에서 20개의 비디오 스트림 전송시 서버가 사용할 수 있는 최대 대역폭 BW_{avail} 은 1.0632Mbps이다. 모바일 통신망의 통신 장애로 인한 프레임 재전송에 필요한 대역폭은 BW_{avail} 에 포함되지 않는 것을 가정하였다. 20개 비디오 스트림은 1초부터 120초 사이의 임의의 시간에 전송이 시작되며 단말기의 버퍼 수준이 1인 스트림 7개, 버퍼 수준이 2인 스트림 7개 그리고 버퍼 수준이 3인 스트림이 6개로 각각 구성된다.

단말기 버퍼 수준을 사전에 파악하여 비디오 스트림별로 통신망 대역폭을 정적으로 할당하는 BSBA(Buffer Sensitive Bandwidth Allocation) 기법[4]과 제시된 RTA(Real-time Adaptive) 스케줄링 기법의 성능이 비교 분석되었다. BSBA 기법은 이용가능한 통신망 대역폭을 스트림의 수로 나눈 후 이 값을 각 스트림의 평균 대역폭과 비교하여 더 작은 값을 해당 스트림의 실제 대역폭으로 결정한다. 이후 모든 스트림의 실제 대역폭의 합이 이용가능한 통신망 대역폭보다 작은 경우 여분의 통신망 대역폭이 존재하는 것으로 가정하며 이 여분의 대역폭은 스트림별로 단말기 버퍼 수준을 고려하여 추가 할당하게 된다. 즉 버퍼 수준이 낮을수록 버퍼 수준이 높은 스트림에 비하여 여유 대역폭을 더 많이 할당받은 후 프레임들을 전송하게 된다. 그러나 BSBA 기법은 스트림별로 통신망 대역폭을 정적으로 할당한 후 전송 중 발생하는 단말기 버퍼 수준을 별도로 고려하지 않기 때문에 이러한 사항을 고려하여 동적으로 대역폭을 조절하는 RTA 기법과 성능에 차이가 발생하게 된다. BSBA 기법과 RTA 기법은 표 1에 기술된 동일한 실험 환경에서 성능이 비교 분석되었다.

그림 5는 RTA 기법과 BSBA 기법 모두 여분의 통신망 대역폭이 없는 경우에서 매 1000초마다 프레임 전송중단 비율을 비교한 결과를 나타내고 있다.

표 1. 실험 환경 변수

스트림의 프레임 수	108000개
사용가능한 최대 통신망 대역폭	1.0632Mbps
최대 클라이언트 수	20
클라이언트별 평균 대역폭	53.16kbps
버퍼 수준	1-3초
시뮬레이션 시간	4500초

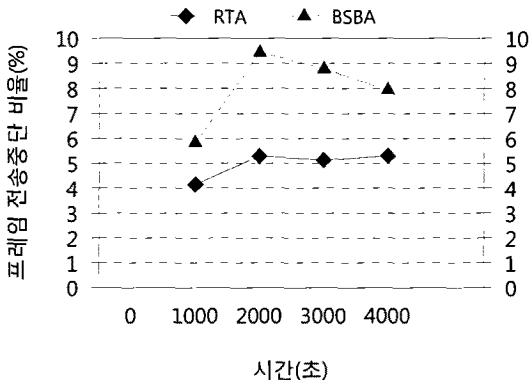


그림 5. 대역폭에 여유가 없는 경우의 프레임 전송중단 비율

1000초 이하에서는 BSBA 기법이 RTA 기법보다 프레임 전송중단 비율이 약 1.7% 더 큰 것으로 나타났고 1000초와 2000초 사이에는 평균 대역폭보다 많은 양의 대역폭이 필요했기 때문에 BSBA 기법의 프레임 전송중단 비율이 약 4%를 초과한 것으로 분석되었다. 3000초와 4000초에서도 BSBA 기법은 RTA 기법에 비하여 프레임 전송중단 비율이 약 3.6%와 2.8% 정도 큰 것으로 나타났다. 서버로부터 프레임을 수신받지 못한 클라이언트는 비디오 재생시 이 프레임들을 재생할 수 없기 때문에 seamless video streaming 유지가 어렵고 재생율이 저하되며 결과적으로 사전에 정의된 QoS 보장이 어렵게 된다.

통신망 대역폭을 사전에 할당한 후 프레임들을 스트림별로 전송하는 BSBA 기법은 통신망 대역폭을 버퍼 수준별로 동적으로 할당하는 RTA 기법에 비하여 스트림들의 실제 대역폭 합이 통신망 대역폭을 초과할 경우 효율적으로 대처하기 어렵기 때문에 프레임 전송중단 비율이 커지는 것으로 분석된다. 그리고 본 논문의 RTA 기법에서는 프레임 전송중단시 우선순위를 반영하기 때문에 표 2와 같이 전송중단된 프레임들 중 우선순위가 낮은 B 프레임의 비율이 93.57%로 가장 크며 중간 우선순위인 P 프레임은

표 2. RTA 기법에서 우선순위별 프레임 전송중단 비율

프레임	I 프레임	P 프레임	B 프레임
전송중단 비율	0.53%	5.90%	93.57%

5.90% 비율로 나타났다. 우선순위가 가장 높은 I 프레임이 전송 중단된 비율은 0.53% 이하로 분석되었다. 비디오 재생시 중요도가 가장 큰 I 프레임의 전송중단 비율을 최소화하고자 우선순위정책이 프레임 전송중단 과정에서 적용되었다.

4.2 여분의 대역폭이 있는 경우의 프레임 전송 중단 비율

여분의 통신망 대역폭이 있는 경우에 BSBA 기법과 RTA 기법의 성능을 비교한 결과가 그림 6과 그림 7에 나타나고 있다. 그림 6은 20개 스트림 중 1개를 제외한 19개 스트림을 대상으로 원래 BW_{avail} 중 5%가 여분의 대역폭(53.16kbps)으로 사용된 결과이며 그림 7은 20개 스트림 중 2개를 제외한 18개 스트림을 대상으로 BW_{avail} 의 10% (106.32kbps)가 여분의 대역폭으로 사용된 결과이다. 5% 여분의 대역폭을 가진 경우 RTA 기법은 BSBA 기법에 비하여 프레임 전송중단 비율이 약 2.2%에서 3.8% 정도 더 낮은 것으로 분석되었다. 그리고 10% 여분의 대역폭이 사용된 경우에는 RTA 기법이 1.6%에서 3.1% 정도의 범위로 5% 여분의 대역폭을 사용한 경우보다 프레임 전송중단 비율의 차이가 감소하였다.

그림 7은 그림 6에 비하여 두 기법 모두 프레임 전송중단 비율이 다소 감소한 것으로 나타났다. 즉 여분의 통신망 대역폭이 많을수록 두 기법에서 프레

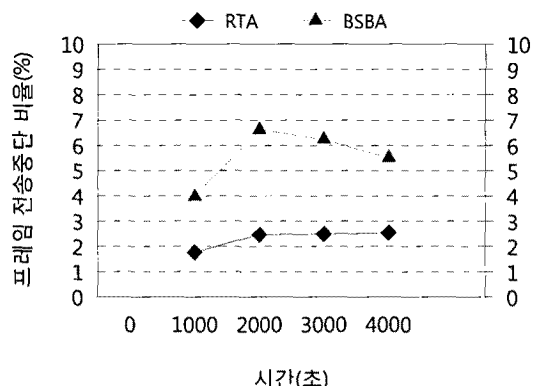


그림 6. 5% 여분의 대역폭이 있는 경우 프레임 전송 중단 비율

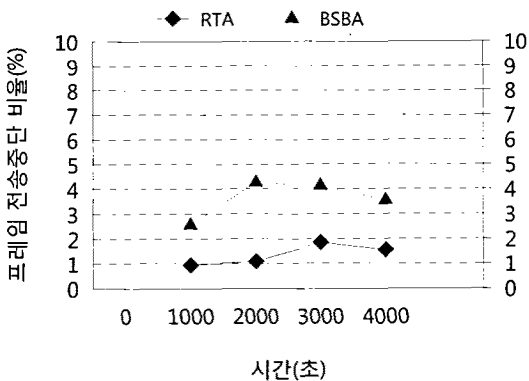


그림 7. 10% 여분의 대역폭이 있는 경우 프레임 전송중단 비율

임 전송중단 비율이 감소되는 것을 의미한다. 그러나 여분의 통신망 대역폭이 존재하는 경우에도 버퍼 수준에 기반한 RTA 기법이 대역폭을 정적으로 할당하는 BSBA 기법보다 프레임 전송중단 비율을 더 감소시킬 수 있음을 알 수 있다.

5. 결 론

모바일 통신망에서 클라이언트가 요청한 MPEG-4 비디오 스트림을 서버가 전송할 때 단말기 버퍼 수준을 활용하여 통신망 대역폭을 동적으로 할당하는 실시간 적응형 스케줄링 프로토콜이 본 논문에서 제시되었다. 프레임의 평균 크기와 재생율에 따라 평균 대역폭이 결정되는 각 비디오 스트림은 VBR 특성상 실제 대역폭의 크기가 가변적이다. 따라서 전송 중인 스트림의 대역폭 합은 이용가능한 통신망 대역폭을 초과하는 기회가 자주 발생하게 된다. 이때 서버는 전송이 중단되는 프레임 수를 최소화할 수 있도록 단말기의 버퍼 수준에 여유가 있을 경우 다음 주기에 전송할 수 있도록 하고 버퍼 수준에 여유가 부족한 스트림의 프레임들을 종료시킨 내에 최대한 전송하도록 대역폭을 스케줄링하게 된다. 본 논문의 스케줄링 기법은 단말기 버퍼 수준을 고려하여 제한된 통신망 대역폭을 효율적으로 사용함으로써 클라이언트 단말기에서 종료시킨 내에 재생되는 프레임 수를 최대화한다. 이 기법은 통신망 대역폭을 정적으로 할당하는 다른 기법과의 성능 비교 결과 비디오 스트림의 서비스 품질과 seamless video streaming 효과를 향상시킬 수 있는 것으로 분석되었다. 제시된 버퍼 수

준에 기반한 실시간 적응형 스케줄링 기법은 통신망 대역폭이 한정된 모바일 환경의 분산 멀티미디어 시스템에서 실시간적 성능을 더욱 향상시킬 수 있는 것으로 기대된다.

참 고 문 헌

- [1] R. Agarwal and A. M. K. Cheng, "Reducing Variation in Bit-Rate Produced by Encoder in MPEG Video," IEEE International Conference on Multimedia Computing and Systems, Florence, Italy, June 1999.
- [2] S. Rao and A. M. K. Cheng, "Scheduling and Routing of Real-Time Multimedia Traffic in Packet-Switched Networks," IEEE International Conference on Multimedia, July-Aug. 2000.
- [3] S. Cen, C. Pu, and R. Staehli, "A Distributed Real-time MPEG Video Audio Player," International Workshop on Network and Operating System Support of Digital Audio and Video, 1995.
- [4] J. Yuen, K. Y. Lam, and E. Chan, "A Fair and Adaptive Scheduling Protocol for Video Stream Transmission in Mobile Environment," IEEE International Conference on Multimedia and Expo, Lausanne, Switzerland, Aug. 2002.
- [5] X. Cheng, P. Mohapatra, S. Lee, and S. Banerjee, "Performance Evaluation of Video Streaming in Multihop Wireless Mesh Networks," Network and Operating Systems Support for Digital Audio and Video, May, 2008.
- [6] K. Y. Lam¹, J. Yuen¹, S. H. Son, and E. Chan, "Scheduling Video Stream Transmissions for Distributed Playback over Mobile Cellular Networks," Int'l Conf. on Parallel and Distributed Systems, Dec. 2002.
- [7] International Organization for Standardization, "Information technology-Coding of audio-visual objects(MPEG-4) Part 2: Video, international standard," ISO/IEC JTC 1/SC 29 14496-2, 2002.

- [8] M. Burza, J. Kang, and P. van der Stok, "Adaptive Streaming of MPEG-based Audio/Video Content over Wireless Networks," *Journal of Multimedia*, Vol.2, No.2, 17-27, 2007.
- [9] X. Hei, Y. Liu, and K. W. Ross, "Inferring Network-Wide Quality in P2P Live Streaming Systems," *IEEE Journal on Selected Areas in Communications* Vol.25, Issue 9, pp. 1640-1654, Dec. 2007.
- [10] H. Espeland, C. H. Lunde, H. K. Stensland, C. Griwodz, and P. Halvorsen, "Transparent Protocol Translation and Load Balancing on a Network Processor in a Media Streaming Scenario," *Network and Operating Systems Support for Digital Audio and Video*, May, 2008.
- [11] M. Krunz, "Bandwidth Allocation Strategies for Transporting Variable-Bit-Rate Video Traffic," *IEEE Communications Magazine*, Jan. 1999.
- [12] F. Yang, Q. Zhang, W. Zhu, and Y. Zhang, "Bit Allocation for Scalable Video Streaming over Mobile Wireless Internet," *IEEE INFOCOM*, 2004.
- [13] 문정모, 이상호, 안지환, "와이브로(WiBro) 이동성 기술," *인터넷 정보학회지* 제6권 제4호, pp. 45-51, 2005.
- [14] S. Mohapatra and N. Venkatasubramanian, "Proactive energy-aware video streaming to mobile handheld devices," *IEEE Conference on Mobile and Wireless Communications Networks*, 2003.
- [15] S. Park, S. Lee, and J. Kim, "Adaptive High Definition MPEG-2 Streaming using Frame-based Prioritization over IEEE 802.11a WLAN," *SPIE*, pp. 1880-1890, 2005.
- [16] <http://trace.eas.asu.edu/cgi-bin/main.cgi>

김진환



1982년 3월 ~ 1986년 2월 서울대학교 컴퓨터공학과 학사
 1986년 3월 ~ 1988년 2월 서울대학교 컴퓨터공학과 석사
 1988년 3월 ~ 1994년 2월 서울대학교 컴퓨터공학과 박사
 1994년 3월 ~ 1996년 2월 서울대학교

컴퓨터신기술공동연구소 특별연구원
 1995년 3월 ~ 현재 한성대학교 멀티미디어공학과 교수
 관심분야 : 멀티미디어시스템, 분산 실시간 시스템