

VOD 시스템에서 제어 메시지를 이용한 클라이언트 버퍼 안정화 알고리즘의 설계 및 분석

박 규 석[†] · 송 태 섭^{††} · 문 병 철^{†††}

요 약

본 논문에서는 분산 멀티미디어 환경에서 데이터 크기의 가변성과 네트워크의 부하변동으로 인하여 발생하는 미디어 불연속성을 방지하면서 동기화를 보장하는 클라이언트 버퍼 안정화 알고리즘을 제안한다. 제안 알고리즘은 클라이언트의 상황에 맞도록 산출된 미디어스트림의 제어량과 제어시간이 포함된 제어 메시지를 멀티미디어 서버로 전송함으로써 클라이언트 버퍼를 안정화 시킬 수 있다. 클라이언트는 주기단위로 수신측의 자원과 네트워크의 상태를 예측하여 오버런 혹은 스타베이션이 발생할 징후가 보이면 제어 메시지를 생성하는데, 제어값들은 클라이언트의 상황과 멀티미디어 서버의 미디어스트림 전송 메커니즘 및 네트워크의 상태를 고려하여 생성된다. 본 알고리즘의 특징은 미디어스트림의 과도한 드롭으로 인한 질 저하를 방지하며, 특히 네트워크의 부하변동이 클 경우에 빠른 속도로 시스템을 안정화시킨다.

The Design and Analysis of Stabilizing Algorithm for Client Buffer using Control Messages on VOD Systems

Kyoo-Seok Park[†] · Tae-Seob Song^{††} · Beong-Cheoul Moon^{†††}

ABSTRACT

In this paper, we propose client buffer stabilizing algorithm which guarantees synchronization and media continuity against variability of media data and network load in the distributed multimedia environments. The proposed algorithm stabilizes the client buffer by transferring control information messages, which includes the control volume and the control time of media stream calculated on the basis of the current status of a client, to a multimedia server.

The client predicts the receiver's resources and network status along with a pre-determined period, and generates control messages whenever overruns or starvations are expected to occur. The values of control messages are calculated by the status of client, the media stream transfer mechanism of multimedia server, and the status of the network. The characteristics of the proposed algorithm are preventing the low quality of media streams when it is dropped excessively and stabilizing the multimedia system when the variability of network loads is high.

1. 서 론

컴퓨터와 통신기술의 발전으로 네트워크 환경에서

멀티미디어를 이용한 응용 및 서비스에 대한 관심이 높아지고 있다. 멀티미디어 데이터를 서비스하는 서버는 클라이언트의 요구에 의해 멀티미디어 서버에서 미디어스트림 단위의 검색이 이루어지는 client-pull구조와 서버에서 여러 개의 클라이언트에 라운드를 통하여 미디어스트림을 서비스하는 server-push구조를 지원해

† 정 회 원 : 경남대학교 컴퓨터공학과 교수
 †† 정 회 원 : 진주전문대학 전자계산과 부교수
 ††† 정 회 원 : 경남대학교 사회교육원 전임연구원
 논문접수 : 1998년 9월 1일, 심사완료 : 1998년 10월 2일

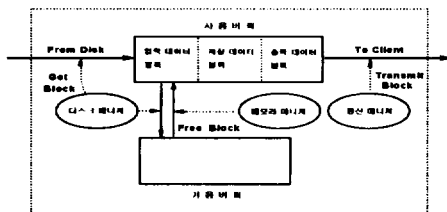
야 한다. 그러나 네트워크를 구성하는 각 요소에서 발생하는 지터(jitter)로 인하여 종단간에 네트워크 전송 지연을 특정 범위내로 보장할 수 없다. 특히 MPEG으로 압축된 멀티미디어 데이터의 크기변화가 매우 큰 폭이어서 네트워크의 처리율도 일정한 범위내로 보장할수는 없으며 멀티미디어 응용에서 요구되는 데이터의 특성인 연속성과 동기성을 보장하는데 장애 요소로 작용하고 있다.

본 논문에서 제안하는 알고리즘은 데이터 크기의 가변성과 네트워크의 부하가 변하더라도 수신측 시스템의 자원이 네트워크의 부하변동을 소화할 수 있을 경우 제어 메시지를 발생하지 않도록 하였으며, 일정한 주기단위로 수신측의 자원과 네트워크의 상태를 예측하여 오버런 또는 스타베이션이 발생할 징후가 보일 경우에 제어 메시지를 발생하도록 설계하였다. 제어 메시지에는 수신측 환경에 적합한 제어시간 및 제어량이 포함되고 서버는 일정한 시간동안 제어를 유지하면서 클라이언트의 상태를 안정화 상태로 제어할 수 있으며 특히, 제안 알고리즘은 네트워크의 변화가 급격할 경우 시스템을 빠른 속도로 안정화시키는 장점이 있다.

2. 관련연구

2.1 VOD 시스템 구조

VOD 시스템은 각 클라이언트에게 미디어스트림의 서비스를 지원하기 위해서는 기본구조인 server-push 또는 client-pull 구조중 하나를 선택해야 한다. server-push구조는 VOD에서 여러 개의 스트림을 여러 클라이언트에 라운드를 통하여 서비스하고, client-pull구조는 클라이언트를 위한 미디어 단위의 검색이 클라이언트의 요구에 의한 응답형식의 구조로, 클라이언트는 주기적인 읽기 요구를 통하여 연속적인 출력을 보장해야 한다. 일반적인 VOD시스템의 구조는 (그림 1)과 같다.



(그림 1) VOD 시스템의 구조
(Fig. 1) Components of VOD system

디스크매니저는 디스크로부터 데이터를 주기억장치 내의 버퍼공간에 저장하며, 이때 필요한 버퍼공간은 메모리매니저의 지원을 받아 버퍼를 할당받는다. 입력 데이터블록은 디스크매니저에 의해 사용되며, 출력 데이터블록은 통신매니저에 의해 소비되고 있는 데이터를 저장하는데 사용한다. 통신매니저는 각 클라이언트에게 출력할 데이터를 네트워크를 통해 해당 클라이언트에게 출력하는 작업을 수행한다.[2]

2.2 검색 구조

멀티미디어 서버의 데이터 검색은 프레임검색과 블록검색이 있다. 프레임검색 방식은 멀티미디어 서버가 독립적으로 각 프레임을 검색하는 방법이며, 블록검색 방식은 서버가 블록단위로 검색하는 방법으로 대역폭의 효율이 좋고, 탐색시간과 회전지연시간이 검색하는 데이터양에 비해 적게 소모되나 scaling과 VCR기능의 사용이 어렵다.[5]

2.3 승인 제어

멀티미디어 데이터의 연속성을 보장하려면 디스크 대역, 프로세서 사이클, 메모리, 네트워크 대역 등과 같은 자원의 적절한 할당이 보장되어야 한다. 따라서 멀티미디어 서버가 여러 파일에 대한 새로운 서비스 요청이 들어오면 현재 서비스되고 있는 작업들에 대한 실시간 서비스 조건을 만족할 수 있는가를 검사해야 한다. 이와같은 것을 승인제어라 하며, 이러한 승인제어는 응용 프로그램마다 각각의 요구 조건이 다르므로 이와같은 정보를 이용하면 보다 융통성 있는 서비스를 제공할 수 있다. 멀티미디어 서버의 서비스 수준은 다음과 같이 정리할 수 있다. 첫째, 확정적인 서비스로 모든 마감시한을 만족하는 것을 보장한다. 이러한 승인제어 방식은 worst-case 또는 결정적(deterministic) 방식이라고 하는데, 최악의 회전시간과 최악의 탐색시간을 가정하여 수용제어를 검사하여 대역폭을 할당한다. 둘째, 통계적인 서비스를 제공하는 것이다. 매 주기마다 요구되는 평균과 편차를 분석하여 특정값의 고정 대역폭을 할당하고, 그에 대한 신뢰도 값 정도로 연속성을 보장한다. 멀티미디어 응용에서 대부분의 승인제어는 worst-case기법을 사용하여 버퍼를 관리하므로 충분한 버퍼 공간을 확보할 수 있으나 재생되는 데이터 크기의 가변성으로 인하여 대체적으로 실제 사용하는 버퍼보다 할당된 버퍼의 양이 크다.[2, 3]

2.4 피드백 메시지를 이용한 비동기성 검사

분산 멀티미디어에서 미디어간의 녹음을 불일치와 네트워크 지연지터가 발생하면 재생시 미디어간의 동기성을 상실하게 되므로 동기화 유지를 위한 메커니즘을 필요로 하게 된다. 클라이언트에서 미디어간의 연속성을 보장하기 위하여 멀티미디어 서버는 클라이언트가 전송해주는 피드백단위인 light-weight 메시지를 이용하여 동기성을 회복시켜 나가는 피드백기법을 사용하고 있다. 피드백 기법에서 사용되는 파라미터로 Δ^m_{min} 은 미디어에 대한 최소 네트워크 지연, Δ^m_{max} 는 최대 네트워크 지연, θ 는 재생지속시간, ρ 는 재생편차, P 는 클라이언트의 안정적인 버퍼링의 양을 나타낸다.

(1) 전송간격

동기성을 회복하기 위한 피드백기법에서 네트워크 지연은 Δ_{min} 와 Δ_{max} 사이의 비결정적 지연이 된다. 클라이언트에서 연속적인 미디어의 재생을 보장하기 위해서 멀티미디어 서버는 클라이언트로 매체 단위를 전송한 후, 미디어스트림 단위 μ 를 식(1)의 범위내에 전송해 주어야 한다.[5.6]

$$(\mu - 1) \times \theta \times (1 - \rho) - (\Delta^m_{max} - \Delta^m_{min}) \dots\dots (1)$$

(2) 버퍼링

멀티미디어 서버에서 두 번째 미디어스트림은 첫 번째 미디어스트림을 전송한 후, $\theta \times (1 - \rho) - (\Delta^m_{max} - \Delta^m_{min})$ 이내에 전송해 주어야 하며, 만약 이 결과가 음이 아니면 식(2)의 조건을 만족해야 한다.

$$\theta \times (1 - \rho) \geq \Delta^m_{max} - \Delta^m_{min} \dots\dots (2)$$

만일 식(2)의 조건이 만족되지 아니할 경우에는, 재생시작 전에 미디어스트림 단위를 prefetch시켜 주어야 한다. 첫번째 및 두번째 미디어스트림 단위의 도착시간 사이의 최대 간격은 $\Delta^m_{max} - \Delta^m_{min}$ 이 된다. 이러한 차가 첫 번째 미디어스트림 단위의 재생 지속시간 (최저 $\theta \times (1 - \rho)$ 까지 가능)을 초과한다면 재생의 종료시점에서 불연속이 발생하게 된다. 이러한 불연속을 방지하기 위해 일정한 수의 미디어스트림 단위를 prefetch하여 재생을 시작하기 전에 버퍼링을 시켜주는 방법을 생각해 볼 수 있으며 prefetch 단위 P 는 식(3)에 의하여 구할 수 있다.

$$P = \lceil \frac{\Delta^m_{max} - \Delta^m_{min}}{\theta \times (1 - \rho)} \rceil \dots\dots (3)$$

여기에서 각 미디어스트림의 재생시구간은 최소한 $\theta \times (1 - \rho)$ 가 된다는 것을 가정하고 있는데 이는 연속성 확보에 필요한 가장 엄격한 요구사항을 나타낸다. 그러나 미디어스트림 단위에 대한 네트워크 지연은 Δ^m_{max} 를 훨씬 하회할 수도 있으며, 재생율에서의 편차는 미디어스트림 단위의 시구간을 $\theta \times (1 - \rho)$ 보다 연장되게 할수도 있다. 이러한 비정상적인 상황 때문에 클라이언트에서 미디어 단위의 누적을 요구하며, 이는 결국 버퍼링의 필요성을 제기해 준다.[5]

(3) Feedback Technique

미디어스트림 단위를 재생하는 시점에서 클라이언트는 자신의 버퍼로부터 디스플레이 모니터로 매체 단위의 데이터 부분을 전송하게 된다. 선택한 미디어스트림의 데이터 부분을 디스플레이 모니터로 전송함과 동시에 클라이언트는 선택한 미디어스트림 단위의 헤더 부분을 피드백 단위로써 멀티미디어 서버로 피드백하여 전송을 요청하게 된다. 그러므로 각 피드백 단위는 피드백 단위의 전송시점에서 현재 재생되고 있는 미디어스트림 단위의 번호만을 담고 있는 light-weight 메시지가 된다. 이러한 전송은 네트워크 부하를 거의 발생시키지 아니한다.

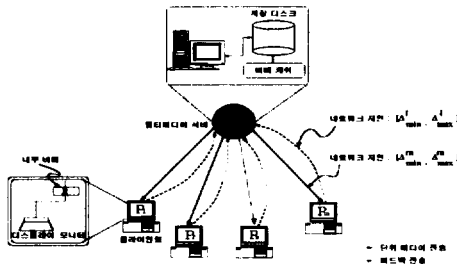
3. 클라이언트 버퍼 안정화 알고리즘

본 논문에서 제안하는 알고리즘은 분산 멀티미디어 환경에서 네트워크의 부하변동으로 인한 미디어 연속성 및 동기화를 보장하기 위하여, 수신측 버퍼의 오버런 혹은 스타베이션 징후를 예측한 후, 수신측 버퍼의 안정화를 위한 제어값 및 제어지속시간을 수신측 환경에 맞게 산출한 후, 이를 피드백 메시지를 이용하여 멀티미디어 서버로 전송한다. 이러한 알고리즘은 네트워크의 상태가 수신측의 미디어스트림의 잔량에 영향을 미치는 현상을 고려하였으며, 제어 메시지를 생성함에 있어 수신측의 상태와 자원의 가용성을 파악하며, 미디어스트림의 연속성을 보장하기 위하여 네트워크의 부하와 수신측의 자원 상태도 고려하고 있다. 그리고 수신측에서 생성한 제어 메시지에에는 제어량과 제어시간을 포함시켜 제어 메시지를 수신한 멀티미디어 서버는 제어값을 제어시간동안 유지함으로써 수신측의 시

시스템 안정된 상태로 유지하여 미디어스트림의 연속성과 동기성을 보장하며, 제어 메시지의 발생 수를 줄인다.

3.1 시스템 구조

클라이언트가 서비스를 요청하면 멀티미디어 서버는 승인제어를 worst-case로 검사하며 클라이언트들이 멀티미디어 서버를 통하여 서비스를 받고 있다면 서비스를 받고 있는 각 클라이언트들은 멀티미디어 서버의 승인제어를 통과했다고 볼 수 있다. 그리고 각 클라이언트들은 네트워크의 부하변동으로 인하여 발생하는 동기화 상실 및 미디어의 불연속성에 대한 완충역할을 수행할 수 있는 각각의 버퍼를 가지고 있다.



(그림 2) 제안 시스템의 환경
(Fig. 2) Environments of proposed system

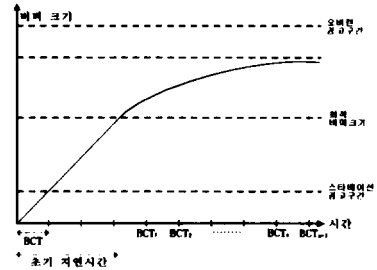
본 논문에서 제안하는 시스템의 환경은 멀티미디어 서버가 수신측으로 전송하는 미디어스트림은 $[D^m_{min}, D^m_{max}]$ 간의 네트워크 지연을 가지며, 수신측에서 멀티미디어 서버로 전송하는 피드백 메시지 역시 $[D^s_{min}, D^s_{max}]$ 간의 네트워크 지연을 가진다.

3.2 클라이언트 버퍼 구조

네트워크의 부하변동으로 인한 클라이언트 버퍼의 오버런과 스타베이션을 방지하고, 미디어의 특성인 연속성을 보장하기 위하여 각 클라이언트들은 완충역할을 수행하는 버퍼를 가진다.

(1) 경고구간

클라이언트에서는 오버런 경고구간과 스타베이션 경고구간을 가진다. 스타베이션 경고구간은 수신측 버퍼의 잔량을 검사할 수 있는 워터마크의 역할을 수행하며, 주기값 산출의 파라미터로 사용된다.



(그림 3) 초기지연시간 및 BCT의 설정
(Fig. 3) Initial delay time & BCT

이러한 역할을 수행하는 스타베이션 경고구간은 신중히 설정해야 하는데, 스타베이션 경고구간이 적을 경우에는 버퍼의 스타베이션 경고 메시지가 멀티미디어 서버에 전달되기 전에 수신측의 버퍼가 고갈되므로 경고구간 설정의 의미가 없어진다. 반대로 경고구간이 클 경우에는 버퍼의 잔존 양이 조금만 감소되더라도 멀티미디어 서버로 제어 메시지가 발생함으로 인해 클라이언트와 멀티미디어 서버에서 빈번한 스케줄러를 호출하게 되는 문제점이 발생한다. 본 논문에서의 스타베이션 경고구간은 식(4)의 최소 prefetch값 이상으로 되도록 설정한다.

$$\text{스타베이션 경고구간} \geq \lceil \frac{D^m_{max} - D^m_{min}}{\theta \times (1 - \rho)} \rceil \dots (4)$$

오버런 경고구간도 역시 버퍼의 이상현상을 감지하는 워터마크의 역할을 수행한다.

(2) 초기지연시간

클라이언트에서는 초기지연시간(그림 3)을 가지며, 초기지연시간은 수신측 스케줄러와 시스템의 가용능력에 융통성을 부여할 수 있어서 미디어스트림의 처리율을 높일 수 있다.

본 논문에서의 초기지연시간은 클라이언트가 멀티미디어 서버로부터 최초의 미디어스트림을 수신한 후 연속적으로 도착하는 미디어스트림을 재생하지 않고 수신측의 버퍼에 누적하여 수신측 버퍼의 잔량이 최적 버퍼 크기를 초과할 때 까지의 시간이다. 아울러 초기지연시간은 멀티미디어 서버로부터 미디어스트림의 입력만 존재하고 출력은 존재하지 않으므로 초기지연시간 동안의 버퍼 증가량은 최대의 증가량으로 가정할 수 있다. 이러한 초기지연시간은 실시간의 성격이 강한 화상회의에서는 최대한 짧게 설정하여야 하나 상호

대화의 응용이 없는 경우에는 초기지연시간을 짧게 설정할 필요는 없다.

3.3 BCT 설정

BCT(Buffer Check Time)는 수신측 버퍼의 잔존 미디어스트림의 양을 검사하기 위하여 일정 주기로 반복 검사하는 주기 단위값이다. 클라이언트는 매 BCT주기마다 수신측 버퍼에서 미디어스트림의 잔량을 검사함으로써 버퍼의 오버런이나 스타베이션 발생을 방지할 수 있다. BCT값을 산출할 때 초기지연시간의 특성은 미디어스트림의 입력만 존재하고 출력은 존재하지 않으므로 버퍼 증가폭은 최대의 버퍼 증가폭이 된다. 그러므로 BCT단위마다 수신측의 버퍼를 검사하면 수신측 버퍼의 잔량이 오버런 상황이 되기 전에 제어 메시지를 전송할 수 있다.

3.4 예측값(α)의 설정

수신측 버퍼에 저장되어 있는 미디어스트림이 고갈 또는 넘치기 전에 이상현상을 스케줄러에게 통보하여 서버로부터 데이터 전송을 제어해야 한다. 따라서 클라이언트의 버퍼 증가량 또는 감소량을 미리 예측하여 제어 메시지를 생성하여야 하며, 본 논문에서 제어 메시지의 발생은 오버런 경고구간 혹은 스타베이션 경고구간에 예측값이 포함될 때 생성된다. 본 논문에서 예측값을 산출할 때 네트워크의 폭주와 단절 현상은 고려하지 않았으며 현재 BCT_n 에서 검사한 잔존 버퍼량과 BCT_{n-1} 에서 검사한 잔존 버퍼량의 차이를 통해서 BCT_{n+1} 에서의 잔존 버퍼의 크기를 예측한다. 따라서, BCT_{n-1} 과 BCT_n 에서 잔존 미디어스트림의 차이 α 는 식(5)로 정의된다.

$$\alpha = \left(\sum_{n=2}^{n} \text{입력 미디어 스트림} - \sum_{n=2}^{n} \text{출력 미디어 스트림} \right) - \left(\sum_{n=1}^{n-1} \text{입력 미디어 스트림} - \sum_{n=1}^{n-1} \text{출력 미디어 스트림} \right) \dots\dots\dots (5)$$

초기지연시간은 수신측에 할당된 가용버퍼의 50% 이상을 채울때 까지의 시간으로 하고 있다. 따라서 잔존 버퍼량이 식(6)의 범위에 있다면 제어 메시지를 발생하지 않는 안정구역(safe zone)으로 정의할 수 있다.

$$\text{스타베이션 경고구간} + \alpha \leq \text{안정구역} \leq$$

$$\text{오버런 경고구간} - \alpha \dots\dots\dots (6)$$

BCT_{n-1} 에서 잔존 미디어 스트림양 $B(BCT_{n-1})$ 과 BCT_n 에서 잔존 미디어 스트림양 $B(BCT_n)$ 의 차이가 α 이면, 네트워크의 폭주나 단절이 없고 시스템 부하에 대한 변동이 계속 유지될때 BCT_{n+1} 에서의 잔존 버퍼량 $B(BCT_{n+1})$ 은 식(7)와 같이 된다.

$$B(BCT_{n+1}) = B(BCT_n) + \alpha \dots\dots\dots (7)$$

여기에서 α 는 양수 혹은 음수값을 가질 수 있으나 부호에 관계없이 현재 BCT_n 에서 식(7)를 이용하면 BCT_{n+1} 에서의 잔존 버퍼량을 예상할 수 있다. BCT_n 에서 BCT_{n+1} 의 스타베이션 예측은 식(8)을 통하여 예측할 수 있다.

$$B(BCT_n) - \alpha < \text{스타베이션 경고구간} \dots\dots (8)$$

버퍼 증가량 α 를 현재 BCT_n 에서 미디어스트림의 잔량 $B(BCT_n)$ 을 감소시킨 값이 스타베이션 경고구간보다 적을 경우에는 BCT_{n+1} 에서는 스타베이션 경고구간에 포함될 가능성이 있는 것으로 예측한다. 그리고 BCT_n 에서 BCT_{n+1} 의 오버런 예측은 식(9)에 의해 예측할 수 있다.

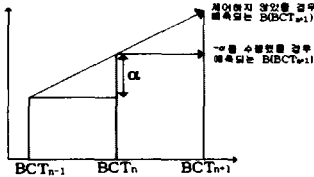
$$B(BCT_n) + \alpha > \text{오버런 경고구간} \dots\dots\dots (9)$$

버퍼 증가량 α 를 현재 BCT_n 에서의 버퍼잔량 $B(BCT_n)$ 에 더한 값이 오버런 경고구간보다 클 경우에 BCT_{n+1} 에서는 오버런 경고구간에 포함될 가능성이 있는 것으로 예측할 수 있다.

3.5 제어값 산출

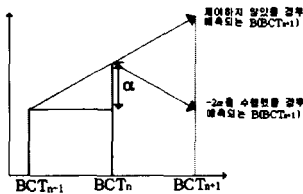
잔존 미디어스트림의 예측값이 오버런 혹은 스타베이션 경고구간에 속할 경우에 생성된 제어 메시지는 제어량과 제어시간을 포함하고 있다. 멀티미디어 서버는 각 클라이언트에 대한 미디어스트림의 서비스 단위를 GOP단위로 전송하며, 클라이언트에서 오버런이 발생하여 미디어스트림에 대한 드롭을 시도할 경우 B프레임에 대한 드롭을 수행 해야한다. BCT_n 과 BCT_{n-1} 사이의 버퍼 잔량 차이가 α 이고 BCT_n 과 BCT_{n+1} 사이에 α 만큼 감소시킬 경우 $B(BCT_{n+1})$ 의 크기는

(그림 4)와 같이 나타낼 수 있다.



(그림 4) $-\alpha$ 의 수행시 예측되는 $B(BCT_{n+1})$ 의 크기
(Fig. 4) The size of $B(BCT_{n+1})$ for $-\alpha$

BCT_n 과 BCT_{n+1} 사이에 2α 만큼 드롭시킬 경우 $B(BCT_{n+1})$ 의 크기는 (그림 5)와 같이 나타낼 수 있다.



(그림 5) -2α 의 수행시 예측되는 $B(BCT_{n+1})$ 의 크기
(Fig. 5) The size of $B(BCT_{n+1})$ for -2α

물론, 이것은 BCT_{n-1} 에서 BCT_n 까지 수신된 미디어스트림의 양과 BCT_n 에서 BCT_{n+1} 사이에 수신된 미디어스트림의 양이 같고, BCT_{n-1} 에서 BCT_n 사이의 네트워크의 지연과 BCT_n 과 BCT_{n+1} 사이의 네트워크 지연이 동일하며, 수신되는 미디어스트림의 양이 동일하고 네트워크 폭주나 단절이 없다는 조건에서의 가정이다. BCT_n 에서 BCT_{n+1} 사이에 수신되는 각 SGOP의 합에 대하여 -2α 만큼 드롭시킨다면 BCT_{n+1} 에서의 버퍼잔량을 BCT_{n-1} 에서의 버퍼잔량과 비슷한 범위를 유지함을 알 수 있다.

$$B(BCT_n) \approx \sum_{BCT=n}^{n+1} \text{입력 SGOP}_i - 2\alpha \dots (10)$$

수신측의 오버런을 방지하기 위하여 2α 혹은 α 만큼의 데이터를 드롭시킬 경우 멀티미디어 서버에서의 데이터 전송횟수를 고려해야 하는데 BCT_n 사이에 수신측에서 수신된 SGOP의 갯수를 k 라 할 경우 BCT_{n+1} 의 잔존 미디어스트림이 $B(BCT_n) - 2\alpha$ 를 수행하려면

BCT 단위동안 수신된 SGOP에 대하여 식(11)와 같이 $\frac{2\alpha}{k}$ 를 드롭해야 한다.

$$B(BCT_{n-1}) \approx \sum_{BCT=n}^{n+1} (\text{Input SGOP}_i - \frac{2\alpha}{k}) \dots (11)$$

α 가 적을 경우에는 한 SGOP에 대하여 -2α 를 수행할 수도 있으나, α 가 클 경우에는 한 개의 SGOP를 완전히 드롭시켜야 하는 경우가 발생하게 된다. 이는 멀티미디어 서버에서 LPB의 재배열 의 의도를 퇴색되게 하므로 BCT 단위동안 수신된 k 개의 각 SGOP에 대하여 $\frac{2\alpha}{k}$ 씩 드롭시킴으로써 미디어의 연속성과 멀티미디어 서버에서 LPB의 재배열 의도를 활용할 수 있다. 그러나 이러한 미디어스트림의 드롭이 지속적으로 진행될 경우에는 클라이언트에서 스타베이션을 유도할 수 있으며, 이는 클라이언트로부터 제어 메시지를 발생하게 하는 등의 오실레이션 현상을 유발시킨다. 따라서 본 논문에서는 제어주기를 산출하여 제어주기 동안 각 SGOP에 대하여 $\frac{2\alpha}{k}$ 씩 드롭을 수행하고 제어주기가 끝난후에는 식(12)와 같이 각 SGOP에 대하여 $\frac{\alpha}{k}$ 만큼 드롭을 수행함으로써 오실레이션 현상을 완화시킨다.

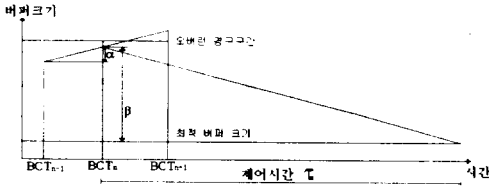
$$B(BCT_n) \approx \sum_{BCT=n}^{n+1} (\text{Input SGOP}_i - \frac{\alpha}{k}) \dots (12)$$

3.6 제어시간(τ) 산출

제어 메시지를 수신한 멀티미디어 서버의 각 SGOP에 대하여 $\frac{2\alpha}{k}$ 씩 계속적으로 드롭을 수행할 수는 없다. 이는 스타베이션으로 유도하게 되고 오실레이션 현상을 유발하므로 일정한 시간동안 각 SGOP에 대하여 드롭을 수행하는데, 본 논문에서는 제어시간을 산출하여 이용함으로써 오실레이션 현상을 완화할 수 있다. BCT_n 에서 예측한 BCT_{n+1} 에서의 수신측 버퍼의 잔량이 오버런 경고구간을 초과했을때 잔존 버퍼량에서 수신측 시스템에서 할당된 버퍼의 50%인 버퍼 크기의 차이를 β 라 할때 식(13)와 같이 나타낼 수 있다.

$$\beta = B(BCT_n) - \text{최적 버퍼 크기} \dots (13)$$

BCT_{n-1} 에서 BCT_n 사이에 입력된 SGOP의 갯수를 k 라고 할 경우, BCT_n 이후 수신되는 각 SGOP에 대하여 $\frac{2\alpha}{k}$ 씩 드롭시키면 식(11)와 같이 $B(BCT_{n+1})$ 와 $B(BCT_{n-1})$ 의 크기는 비슷하게 된다.



(그림 6) 제어시간(τ)의 산출
(Fig. 6) Calculation of control time(τ)

그러므로 현재 수신측 버퍼의 잔량과 최적 버퍼 잔량과의 차이인 β를 BCT_{n-1}와 BCT_n의 버퍼 잔량 차이인 α로 나누어 BCT검사횟수 γ를 구할 수 있으며 식(14)와 같이 나타낼 수 있다.

$$\gamma = \frac{B(BCT_n) - \text{최적 버퍼 크기}}{B(BCT_n) - B(BCT_{n-1})} = \frac{\beta}{\alpha} \dots\dots (14)$$

γ는 BCT의 검사횟수로서, 초기지연시 산출된 BCT 주기시간에 γ를 곱하면 BCT횟수에 대한 시간 τ를 구할 수 있다.

$$\tau = \gamma \times BCT \text{ 값} \dots\dots\dots (15)$$

즉, τ 시간동안 각 SGOP에 대하여 $\frac{2\alpha}{k}$ 를 드롭시킨다면 τ 시간이 지난 후의 수신측 버퍼 잔량은 시스템에서 할당된 버퍼의 50%범위를 유지할 수 있으며, 앞에서의 각 모듈을 고려한 제어 메시지 생성 알고리즘은 (그림 7)과 같다.

```

if in1_port media_stream arrives then
if it is the first
system_clock_on
accumulate the media_stream to buffer
else
accumulate media_stream to buffer
if accumulated media_stream size < starvation_caution_size
wait next media_stream
else
BCT = system_clock;
if accumulated media_stream size < optimal_buffer_size
wait next media_stream
else
while service_time = service_end_time
if system_clock < BCT then
accumulate media_stream to buffer
wait next media_stream
else
accumulate media_stream to buffer
if accumulate media-stream > overrun_buffer_size
    
```

```

or < starvation_buffer_size
= B(BCT_n) size of accumulate media_stream
α = B(BCT_n) - B(BCT_{n-1})
Pl = B(BCT_n) + α
if pl > overrun_buffer_size
β = B(BCT_n) - optimal_buffer_size
γ = β / α
τ = γ × BCT
else
system_clock = 0
while-end
wait next media stream;
    
```

(그림 7) 제어 메시지 생성 알고리즘
(Fig. 7) Generating algorithm of control message

3.7 BCT값 변경

제어시간 τ가 BCT단위보다 적다면 문제는 발생하지 않으나, 제어시간 τ가 BCT단위보다 클 경우에는 현재 멀티미디어 서버가 제어를 하고 있음에도 불구하고, 클라이언트의 스케줄러는 제어 메시지를 발생하게 되며, 이는 수신측과 멀티미디어 서버 양측에 제어에 대한 일관성을 상실하게 만드는 요인이 된다. 본 논문에서는 수신측의 스케줄러가 제어 메시지를 생성하여 멀티미디어 서버를 전송하는 시점에서 수신측 스케줄러의 BCT단위를 일시적으로 제어시간 τ와 동일하게 유지하여 제어시간동안 제어 메시지의 생성을 억제시킨다. α 값은 B(BCT_n)과 B(BCT_{n-1})의 차이로써 α 값이 크다는 것은 BCT단위마다 변하는 수신측 잔존 미디어스트림의 변화 폭이 크다는 것을 의미하며, 이는 네트워크의 부하변동이 크다는 의미가 된다. 따라서 네트워크의 부하변동이 클 경우에는 즉각적이면서 최대한 빠른 시간내에 수신측 버퍼의 미디어 잔량에 대해 안정구역으로 제어를 해야한다.

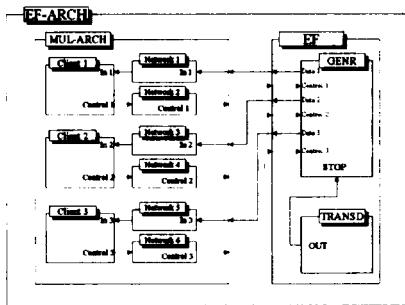
제안 알고리즘은 BCT단위마다 잔존 미디어스트림의 변동폭인 α가 커질수록 제어시간이 짧아지고, α가 적을수록 제어시간은 길어지는 특성을 가지고 있다. 그러므로 BCT단위마다 잔존 미디어스트림의 변동폭이 클수록 짧은 제어시간으로 잔존하는 미디어스트림을 안정구역으로 제어할 수 있다. 물론 α 값이 커지면 SGOP단위마다 제어값인 $\frac{2\alpha}{k}$ 가 커지게 되므로 미디어스트림에 대한 손실이 커질 수 있다. 반대로 α 값이 적으면 $\frac{2\alpha}{k}$ 가 적어져 미디어스트림에 대한 손실이 적어진다. 또한 본 알고리즘은 잔존 비동기 에러요소

를 포함하고 있다. VBR의 특성인 SGOP데이터 양의 변화와 제어 메시지를 생성하는 시간과 멀티미디어 서버에서 제어 메시지를 수신하는 시간차, 그리고 제어 메시지를 생성할 때 이미 네트워크에 실려 있는 SGOP 데이터가 잔존 비동기 에러 요인으로 작용한다.

4. 시뮬레이션 및 평가

4.1 시뮬레이션 모델

제안 알고리즘을 DEVS 기반으로 한 PC-SCHEME 으로 작성하여 시뮬레이션 하였다. 본 논문에서의 시뮬레이션 모델은 클라이언트, 네트워크, 멀티미디어 서버 등으로 구성된 14개의 모듈로 구성되었고 이러한 모듈간의 데이터 전달을 위하여 25개의 포트를 정의하였다. 네트워크는 크게 두 가지의 형태로 구분되는데, 멀티미디어 서버로부터 생성된 미디어스트림을 각 클라이언트에 전달하기 위한 network1, network3과 network5가 정의되었으며 클라이언트에서 생성된 피드백 메시지를 멀티미디어 서버로 전달하기 위하여 network2, network4와 network6이 사용되고 있다. 각 네트워크는 독립적인 네트워크지연과 버퍼를 가지고 있다. 그러므로 멀티미디어 서버에서 생성된 미디어스트림과 각 클라이언트에서 생성된 피드백 메시지는 독립적인 네트워크의 지연을 가지고 시뮬레이션 되도록 설계되어 있다.



(그림 8) 시뮬레이션 모델
(Fig. 8) Simulation model

4.2 시뮬레이션 파라미터

(1) 버퍼

클라이언트들의 버퍼에 설정된 구간 범위는 <표 1>과 같으며, 버퍼 사이즈는 총 16MB로 설정하였다. 초

기지연시간 동안 최적 버퍼 사이즈인 8MB까지 미디어 스트림을 누적시키고, 누적된 미디어스트림의 양이 최적 버퍼 사이즈를 초과할 경우에는 수신측 매체 재생기로 미디어스트림을 재생한다. 본 논문에서 사용하는 각 구간값은 <표 1>과 같다.

<표 1> 클라이언트 버퍼의 각 구간 값
<Table 1> Range of client buffer

스타베이션 경고구간	4.096 KB	오버런 경고구간	14,288 KB
최적 버퍼 크기	8.192 KB	버퍼 크기	16,384 KB

(2) 미디어스트림

본 논문에서 사용하는 GOP는 I프레임 1개, P프레임 3개, B프레임 6개로 구성되어 있으며, SGOP은 15개의 GOP를 묶어 사용한다. MPEG으로 압축된 미디어스트림은 VBR의 특성을 가지므로 매 프레임마다 I,P,B 프레임은 데이터 양이 가변적이며, 데이터의 크기는 <표 2>와 같은 크기로 변한다.

<표 2> MPEG에서 I,P,B 데이터 크기의 변화량
<Table 2> Variance of I,P,B data size in MPEG

프레임	μ (KB)	σ (KB)	Max(KB)	Min(KB)
I	12.28	3.13	25.99	7.58
P	7.86	4.08	27.20	0.52
B	2.24	1.68	17.48	0.32

SGOP의 데이터 크기는 최소 165.9KB, 최대 3187.05 KB, 평균 739.5KB로 한다.

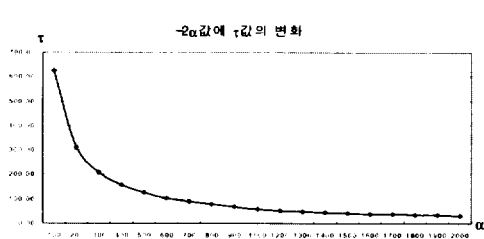
<표 3> SGOP의 데이터 크기
<Table 3> Date size of SGOP

	평균	최대	최소
I	184.2 KB	389.85 KB	113.7 KB
P	353.7 KB	1224 KB	23.4 KB
B	201.6 KB	1573.2 KB	28.8 KB
SGOP의 크기	739.5 KB	3187.05 KB	165.9 KB

스타베이션 경고구간이 4096KB이므로 BCT단위에 수신되는 미디어스트림의 갯수는 최소 1.28개, 최대 24.68개, 평균 5.53개가 된다.

4.3 시뮬레이션 분석

제안된 알고리즘은 네트워크의 부하변동이 급격히 변할 경우에는 α 값이 커지게 된다. 네트워크의 부하 변동 폭과 α 값은 비례 관계가 있으며, α 값의 변동폭은 제어시간 산출 알고리즘에 영향을 미친다. 네트워크의 부하변동이 클 경우에는 제어시간이 적어지고, 네트워크의 부하변동이 적을 경우에는 제어시간이 커진다. 또한, 오버런이 예측되는 상황에서 생성된 α 값이 일정한 범위 이상 커지게 될 경우 제어시간은 (그림 9)에서와 같이 일정 시간내로 수렴하고 있음을 알 수 있다.



(그림 9) -2α 값에 따른 τ 값의 변화
(Fig. 9) Variance of τ value for -2α

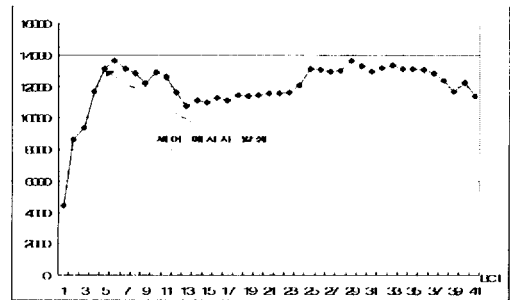
본 논문에서 제안하는 미디어스트림의 감소량인 $-\frac{2\alpha}{k}$ 는 B프레임에 대하여 적용되어야 미디어스트림의 화질을 보장할 수 있다. 따라서 제어량인 $-\frac{2\alpha}{k}$ 값이 <표 3>의 SGOP내의 B프레임 평균 데이터크기 2016KB를 초과해서는 안된다. 따라서 BCT단위에 들어오는 SGOP는 평균 5.53개 이므로 α 가 500KB 이상이 되지 않으면 B프레임에 대해서만 드롭이 발생한다<표 4>.

<표 4> -2α 값에 대한 드롭의 양
(Table 4) Drop quantity for -2α

-2α \ k	3	4	5	6	7	8	9	10	11	12
200KB	66.67	50.00	40.00	33.33	28.57	25.00	22.22	20.00	18.18	16.67
400KB	133.33	100.00	80.00	66.67	57.14	50.00	44.44	40.00	36.36	33.33
600KB	200.00	150.00	120.00	100.00	85.71	75.00	66.67	60.00	54.55	50.00
800KB	266.67	200.00	160.00	133.33	114.29	100.00	88.89	80.00	72.73	66.67
1000KB	333.33	250.00	200.00	166.67	142.86	125.00	111.11	100.00	90.91	83.33
1200KB	400.00	300.00	240.00	200.00	171.43	150.00	133.33	120.00	109.09	100.00
1400KB	466.67	350.00	280.00	233.33	200.00	175.00	155.56	140.00	127.27	116.67
1600KB	533.33	400.00	320.00	266.67	228.57	200.00	177.78	160.00	145.45	133.33
1800KB	600.00	450.00	360.00	300.00	257.14	225.00	200.00	180.00	163.64	150.00
2200KB	733.33	550.00	440.00	366.67	314.29	275.00	244.44	220.00	200.00	183.33
2400KB	800.00	600.00	480.00	400.00	342.86	300.00	266.67	240.00	218.18	200.00
2600KB	866.67	650.00	520.00	433.33	371.43	325.00	288.89	260.00	236.36	216.67

그러나, 수신측 버퍼에서 오버런이 발생했다는 것은 BCT단위에 수신되는 k의 수가 크다는 것을 의미한다. 그리고 BCT단위동안 SGOP이 최대 24.68개가 도착할 수 있는 것을 감안하면, 오버런이 발생하여 SGOP에 대한 $\frac{2\alpha}{k}$ 를 드롭시키면 B프레임에 대하여 드롭이 된다고 볼 수 있다.

(그림 10)은 제안 알고리즘을 평가한 결과를 나타낸 것으로, 오버런이 발생하면 $-\frac{2\alpha}{k}$ 를 각 SGOP에 대하여 τ 시간동안 수행하여 최적 버퍼 크기인 잔존 미디어스트림 양을 확보하기 위하여 멀티미디어 서버에서 제어시간동안 각 SGOP에 대해 $-\frac{2\alpha}{k}$ 를 수행하고, 제어시간이 끝난후에는 $-\frac{\alpha}{k}$ 를 수행함을 보여준다.



(그림 10) 시뮬레이션 결과
(Fig. 10) Result of simulation

제어 메시지가 멀티미디어 서버로 전송되는 네트워크의 지연, 제어 메시지를 생성하는 시점에 멀티미디어 서버에서 이미 서비스한 미디어스트림의 양과 VBR 특성인 미디어스트림의 데이터 크기의 가변성 등 잔존 비동기 요소로 인하여 제어 메시지가 발생한 시점에서부터 잔존 미디어스트림의 제어시간이 끝난후 잔존 미디어스트림 양의 차이는 1506KB이며, 본 논문에서 예상한 최적 버퍼 크기와는 3452KB의 차이를 보였다. 그리고 제어 메시지를 전송한 후, 곧바로 수신측 잔존 미디어스트림이 감소하지는 않고 최소 한개의 BCT가 지난 후 잔존 미디어스트림의 양이 감소되고 있음을 보여 주고 있다. 그리고 제어시간이 끝난후 $-\alpha$ 를 각 SGOP에 대하여 감소시킴으로써 오실레이션 현상이 줄어들면서 잔존 미디어스트림의 안정구역 시간이 연장되고 있음을 알 수 있다.

5. 결 론

본 논문에서는 네트워크의 부하변동과 미디어스트림 크기의 가변성으로 인하여 발생하는 수신측 미디어 스트림의 불연속성을 방지하며 동기화를 보장하기 위한 제어 메시지 생성 알고리즘을 제안하였다. 멀티미디어 서버는 수신측 시스템의 자원 상태와 네트워크의 부하변동이 미디어스트림의 연속성을 보장할 경우에는 server-push구조로 운영되며, 수신측 미디어스트림의 연속성을 보장할 수 없는 상태가 예측되면 멀티미디어 서버는 클라이언트에서 생성한 제어 메시지를 기반으로 client-pull구조로 동작한다.

멀티미디어 서버가 client-pull구조로 동작하기 위하여 필요한 정보는 수신측의 미디어 연속성을 보장하기 위하여 클라이언트에서 생성한 제어 메시지에 포함시켰으며, 제어 메시지에는 제어량과 제어시간을 포함하고 있다.

제어 메시지는 미디어스트림의 연속성을 보장하기 위하여 네트워크의 상태와 클라이언트의 자원상태까지 고려하였으며, 특히 제어량이 B프레임에 적용 받도록 고려함으로써 드롭을 수행하더라도 미디어스트림의 화질에 큰 영향을 미치지 않도록 하였고, 네트워크의 부하변동이 급격하게 발생할 경우에도 빠른 시간내에 수신측 시스템을 안정화 상태로 유지할 수 있다. 그러나 미디어스트림의 가변성과 멀티미디어 서버에 제어 메시지가 도착할 때까지의 시간 및 제어 메시지가 생성될 시점에 이미 네트워크에 존재하는 미디어스트림의 잔존 비동기성 예외요인에 대한 처리에서의 안정화 상태는 미흡했다.

향후, 잔존 비동기성 예외 요인에 대한 처리를 함으로써 제어가 이루어진 후 목표로 설정했던 안정구역으로의 시스템 제어에 대한 연구와 server-push구조에서 client-push구조로 전환할 때의 승인제어 문제에 대한 지속적인 연구가 요구된다.

참 고 문 헌

[1] E.T.Chang, "Storage and Retrieval of Compressed Video," Ph.D. Thesis, California University, Berkeley, 1996.

[2] P.Venkat Rangan & Harrick M.Vin. "Efficient Techniques for Digital Continuous Multimedia". IEEE Transactions Knowledge and Data Engi-

neering Special Issues and Multimedia Information Systems, August, 1993.

[3] D.James Gemmel, Harrick M.Vin, Dilip D.Kandlur, P.Venkat Rangan, "Multimedia Storgae : A Tutorial and Survey," IEEE Computer, Vol.28, Issue 5, pp.40-49, May, 1995.

[4] P.Venkat Rangan, Harrick M.Vin, Srinivas Ramanathan, "Design an On-Demand Mutimedia Service," IEEE Communication Magazine, Vol.30, No.7, pp.56-65, July, 1992.

[5] Vin,Srinivas Ramanathan, P.Venkat Rangan, "Feedback Techniques for Intra-Media Continuity and Inter-Media Synchroniaztion in Distributed Mutimedia System," The Computer Journal 36(1), pp.19-31, April, 1993.

[6] Vin,Srinivas Ramanathan, P.Venkat Rangan, "Adaptive Feedback Techniques for Synchroniazied in Mutimedia Retrieval over Integrated Networks," IEEE/ACM Transaction on Networking, Vol.1, No.2, April, 1993.

박 규 석

e-mail : kspark@hanma.kyungnam.ac.kr
 1988년 중앙대학교 전자계산학과
 (이학 석/박사)
 1990년~1991년 미 UCLA 객원교수
 1982년~현재 경남대학교 컴퓨터
 공학과 교수

1998년~현재 행정자치부 정책자문위원(전자정부분과)
 관심분야 : 운영체제, 멀티미디어시스템, 정보통신시스템

송 태 섭

e-mail : tssong@namgang.chinju-c.ac.kr
 1986년 경남대학교 전자계산학과
 졸업(학사)
 1989년 경남대학교 산업대학원 전
 자계산학과(공학석사)
 1996년 경남대학교 대학원 컴퓨터
 학과(박사과정수료)

1992년~현재 전주전문대학 전자계산과 부교수
 관심분야 : 멀티미디어시스템, 정보통신시스템, 분산처
 리시스템

문 병 철

e-mail : gomdoli@kyungnam.ac.kr

1993년 경남대학교 전자계산학과
졸업(학사)

1995년 경남대학교 대학원 전자
계산학과(공학석사)

1998년 경남대학교 대학원 컴퓨터
공학과(박사과정)

1996년~현재 경남대학교 사회교육원 전임연구원

관심분야 : 멀티미디어시스템, 시뮬레이션, 분산처리시스템