

VOD 시스템에서 클라이언트 버퍼를 위한 전송율 제어 알고리즘의 설계 및 분석

김원규* · 박규석**

요 약

본 논문에서는 VOD 서비스 시스템에서 서버의 데이터 전송율과 클라이언트의 데이터 소비율에 기반하여 서버의 전송율을 조정하는 전송율 제어 알고리즘을 설계하고 분석하였다.

제안 알고리즘은 클라이언트 버퍼의 오실레이션 현상을 줄여 버퍼를 안정화시키고 또한 클라이언트 시스템의 스케줄링 부하를 경감시키기 위하여 클라이언트 버퍼의 상태를 일정한 시간마다 검사하는 BCT를 이용한다. 이 BCT를 기반으로 하여 클라이언트의 데이터 소비율을 산출하고 버퍼의 크기를 예측한다. 예측한 버퍼의 크기가 오버런(overrun) 또는 스타베이션(starvation) 경고구간에 속할 경우에는 서버의 데이터 전송율을 적극적으로 조정하여 클라이언트 버퍼의 오버런과 스타베이션이 발생되지 않도록 한다.

네트워크의 부하 변동으로 인하여 전송율 제어 정책의 수행만으로는 클라이언트 버퍼를 안정시키지 못한다. 따라서 본 논문에서는 전송율 제어 정책과 미디어 스케일링 정책을 병행 수행하여 클라이언트 버퍼의 오버런 또는 스타베이션이 발생되지 않고 안정된 상태를 유지하도록 하였으며, 제안한 알고리즘에 대해 시뮬레이션을 통하여 그 성능을 평가하였다.

Design and Analysis of a Transmission Rate Control Algorithm for Client's Buffer on VOD Systems

Wan Kyoo Kim, Kyoo Seok Park

ABSTRACT

In this paper, we present the design and analyze the transmission rate of a control algorithm for the client buffer on the VOD systems. The design is based on the server's data transmission rate and client's data consumption rate.

The proposed algorithm stabilizes the client's buffer by reducing the oscillation phenomena of the buffer. And it uses the BCT (Buffer Check Time) to reduce the scheduling load of the client system. The client's data consumption rate and the buffer size are calculated on the basis of BCT. In case that the predicted buffer size operates in the overrun or starvation warning range, the data transmission rates of the server are regulated adaptively to prevent the overrun or starvation of the buffer.

As the load fluctuation of the network, the transmission rate control policy cannot stabilize the client's buffer alone. The media scaling policy, therefore, also performs its complementary part in stabilizing the client's buffer, and performance evaluation of the proposed algorithm is provided through simulation.

1. 서 론

컴퓨터와 통신망 기술이 급속도로 발전함에 따라

텍스트, 이미지, 오디오, 비디오 등과 같은 다양한 매체들을 하나의 통합된 멀티미디어 응용 서비스 형태로 실현하기 위해 많은 연구가 활발히 진행되고 있다. 대표적인 멀티미디어 응용 서비스들은 주문형 비디오(VOD : Video On Demand), 원격화상 회의, 홈

* 진주산업대학교 전자계산학과 부교수

** 경남대학교 컴퓨터공학과 교수

쇼핑, 원격진료, 전자 도서관 등으로 그 활용 범위가 광범위하다.

높은 전송율과 저장 능력을 지원하는 서버는 하드웨어 기술의 발달로 현저히 향상되었으나 저장 능력의 향상과 더불어 멀티미디어 지원을 위해 저장 장치의 사용 및 제어에 관한 연구가 진행되고 있다[1,2,3, 4,5].

그리고 멀티미디어 데이터를 실시간으로 서비스하기 위해서는 서버의 전송율과 버퍼 공간 및 네트워크의 가용 능력 등이 고려되어야 하며 이러한 고려사항들은 멀티미디어 데이터의 특성인 대용량 처리와 연속성을 보장하여야 한다[2,6,7,8,9].

본 논문에서는 HeiTS[10]에서 발생하는 오실레이션 현상을 줄여 버퍼를 안정화시키고 또한 클라이언트 시스템의 스케줄링 부하를 경감시키기 위하여 클라이언트 버퍼의 상태를 일정한 시간마다 체크하는 BCT(Buffer Check Time)를 이용하였다.

이 BCT를 기반으로 하여 클라이언트의 데이터 소비율을 산출하며, 클라이언트 버퍼의 크기를 예측한다. 그리고 예측되는 버퍼 크기에 따라 서버의 데이터 전송율을 적응적으로 조정한다.

따라서 본 논문에서는 클라이언트 버퍼의 상태에 따라 서버에 대한 전송율을 제어하는 정책과 미디어 스케일링 정책을 병행 수행함으로써 클라이언트 버퍼의 오실레이션 현상을 줄여 버퍼 상태를 안정화하였으며 또한 연속 재생을 보장한다.

본 논문의 제 2장에서는 관련연구, 제 3장에서는 전송율 제어 알고리즘의 설계에 관하여, 제 4장에서는 시뮬레이션 및 분석, 제 5장에서는 결론 및 향후 연구 방향에 대하여 기술한다.

2. 관련 연구

2.1 VOD 서비스 개요

대화형 서비스를 제공하기 위해 VOD 서버는 연속적인 전송을 요구하는 대용량의 비디오 데이터를 실시간에 처리해야 하며 시스템 자원을 효율적으로 활용하여 최대수의 비디오 스트림을 서비스 해야 한다. 다수의 비디오 스트림을 동시에 재생하기 위해 액세스 시간을 최소화하는 저장 장치를 사용하고 비디오 데이터의 특징과 대용량 저장 장치의 특성을 고려하여 미디어 재생을 수행해야 한다. 대용량의 비디

오 데이터를 저장하고 제한된 전송 대역폭에서 실시간 전송을 보장하기 위해서는 MPEG과 같은 비디오 압축 기법을 사용한다. VOD 서버를 위한 저장 매체로 대용량 데이터의 저장이나 고속 데이터 전송을 위해 디스크 어레이 또는 RAID(Redundant Array of Inexpensive Disks)를 주로 사용한다.

2.2 미디어 스케일링

미디어 스케일링(media scaling)은 데이터 스트림 본래의 내용을 부 표본화(subsample)하여 그 데이터 스트림의 일부분만을 나타내는 것을 의미하며, 일반적으로 스케일링은 전송 측이나 또는 수신 측에서 수행될 수 있다[10,11,12,14,16]. 그리고 미디어 스트림에 스케일링을 적용함으로써 네트워크의 상태변화에 따른 수신측 자원의 상태를 동적으로 적응시킬 수 있다. 즉, 시스템 자원이 과부하 상황이 발생할 경우에 미디어 스트림의 서비스를 중단하는 것이 아니라 자원의 부하 상태가 임계상태에 도달할 때 미디어 스트림의 질을 점진적으로 감소시켜 조절함으로써 시스템 자원을 최적의 부하상태로 유지하면서, 의미 있는 미디어 스트림을 사용자에게 제공해 줄 수 있다 [11,12,13,14].

2.2.1 HeiTS의 미디어 스케일링

HeiTS(Heidelberg Transport System)는 IBM 유럽 네트워킹 센터에서 개발한 디지털 오디오 및 비디오의 실시간 전송을 위한 멀티미디어 통신 시스템이다[10]. HeiTS의 미디어 스케일링은 전송수준(transport level)에서 수행되며, 네트워크의 응답성에 중점을 두고 있다. 미디어 스케일링은 자원의 부하 상태에 따라서 scaling up과 scaling down으로 수행하며 Transparent scaling과 Non-transparent scaling 두가지 방법으로 나눌수 있다.

Transparent scaling 방법은 상위의 프로토콜 및 응용층과는 관계없이 독립적으로 트랜스포트 시스템에 있는 미디어에 대하여 스케일링을 적용한다. 이 방법의 스케일링은 미디어 스트림의 일부분(single frames 또는 substreams)을 드롭(dropping)시켜서 조절한다. 이 스케일링은 비디오 데이터 스트림에 대하여 적용이 적절하며, 오디오 데이터 스트림에 대해서는 사용자가 쉽게 인지할 수 있기 때문에 적용이

적절치가 못하다.

Non-transparent scaling 방법은 상위의 프로토콜 및 응용층과 트랜스포트 시스템과의 상호작용 관계가 있으며, 이 방법에서 스케일링은 미디어 스트림이 트랜스포트층에 표현되기 전에 미디어 스트림을 수정한다. 즉, 코딩 알고리즘의 일부 파라미터들을 수정하거나, 이전에 다른 형태로 엔코딩된 스트림의 레코딩시에 DCT계수를 수정하는 것이다. 특히 미디어 스트림의 엔코딩시에 DCT 계수의 수정에 관한 것은 압축기술에 의존하고 있다.

2.2.2 HeiTS의 전송을 제어

HeiTS는 전송을 제어에 기반 하는(rate-controlled) 전송 시스템이다[10, 14]. 모든 데이터 스트림이 HeiTS 연결을 통하여 전송되기 때문에 그 시스템은 관련된 QoS 파라미터 셋트에 의해서 데이터 스트림의 전송율을 알 수 있으며 이 데이터 스트림의 전송율은 전송 수준의 인터페이스에서 시간 주기마다 논리적 데이터 단위들(예, 비디오 프레임)에 의하여 주어진다. 그리고 HeiTS는 매 패킷(packet)들의 도착을 모니터 하기 위해서 이 정보를 이용한다. 어떤 패킷이 늦게 도착한다는 것은 전송 시스템에서 병목 현상이 있음을 나타낸다. 이 경우에 수신측은 전송측에게 전송 시스템이 과부하 상태임을 알려서 그 스트림을 스케일 다운시키며, 과부하 상태가 일단 끝나면 다시 그 스트림을 스케일 업시킨다. 여기에서 발생하는 문제점은 매 패킷이 도착할 때마다 도착 간격의 계산 및 네트워크의 상태 분석으로 인하여 클라이언트 시스템의 스케줄링에 대한 부하의 가중과 클라이언트 버퍼의 오실레이션(oscillation) 현상이 발생하는 문제점이 있다[14].

2.2.3 미디어 스케일링 메커니즘

미디어 스케일링은 자원의 가용 능력 보다 더 많은 자원사용의 요구에 대하여 적응시키기 위한 것으로서, 자원의 부하 상태에 대한 자원의 가용 능력, 자원의 부족 등을 모니터 하는 메커니즘이 필요하다[11,12,13,14].

자원 모니터(RM: Resource Monitor)는 자원의 부하상태(RLS: Resource Load State)를 관찰하여 그 자원의 부하상태 변화에 따른 지침을 제공하게 되며 자원 부하상태의 유형은 그림 1과 같이 나타낸다.

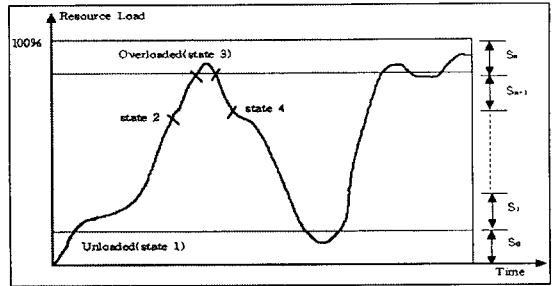


그림 1. 자원의 부하 상태

그림 1의 상태의 분류는 자원의 안정된 상태뿐만 아니라, state 2의 동태에 의해 자원이 부족한 상태가 되는 것을 미리 파악하여 사전에 필요한 조치를 하기 위한 것이다. 즉, 자원의 상태가 state 3에 도달하지 못하도록 사전 조치를 생성한다. 따라서 자원 부하상태(RLS)의 오실레이션 현상을 줄여 자원 부하상태를 안정되게 유지할 수 있다. 부하구역을 S_0, \dots, S_n 으로 세분화하여 각 state들의 동태를 탐지할 수 있다. S_0 는 안정상태로 되기 이전의 언로드 상태(state 1)이며, S_n 은 오버로드 상태(state 3)이다.

3. 전송을 제어 메커니즘

본 논문에서 제시하고 있는 전송을 제어 알고리즘은 클라이언트의 버퍼 상태에 따라 전송 지연 또는 긴급 전송으로 VOD 서버의 데이터 전송율을 적절히 조정하게 함으로써 클라이언트의 버퍼는 오버런 또는 스타베이션이 발생되지 않고 안정된 버퍼 상태를 유지할 수 있다. 그리고 서버의 전송을 조절은 멀티미디어 데이터의 연속성 요구 조건을 만족하는 범위 내에서 이루어지므로 클라이언트의 연속 재생이 보장된다. 이때, 클라이언트 버퍼의 상태에 따라 서버에 대한 전송율을 제어하는 정책과 미디어 스케일링 정책을 병행 수행함으로써 클라이언트 버퍼는 가변적인 네트워크의 상태 변화에 잘 적응하고 오실레이션 현상을 줄여 안정된 버퍼의 상태를 유지할 수 있다.

3.1 VOD 데이터의 전송

본 논문에서는 VOD 서버의 디스크 관리, 버퍼 관리, 승인 제어 등과 같은 부분은 고려하지 않으며 다만, 클라이언트의 전송율 제어 메시지의 요청에 대한

처리하는 서버의 전송 관리자에 의해 지원되는 것을 전제로 한다. 그리고 네트워크의 지연은 자원예약에 의해서 최소 지연 시간과 최대 지연 시간이 제한되는 지연 시간 범위의 네트워크를 가정한다.

VOD 서버의 데이터 블록 전송 주기 시간은 식 3-1과 같이 나타내며, 서버의 전송 관리자에 의해 전송 주기 시간은 조절될 수 있다. 그리고 조절되는 전송 주기 시간은 연속성 요구 조건을 만족하여야 하며 식 3-2와 같다. 즉, 전송 주기 시간이 재생되는 시간보다 적어야 연속성을 만족한다는 의미이다.

전송 주기 시간 : T_{Trans} (식 3-1)

연속성 요구 조건: $T_{trans} \leq \frac{\eta_{vs}}{R_{pl}}$ (식 3-2)

3.2 클라이언트의 버퍼 구조

네트워크의 상태는 수신측 버퍼와 비례 관계가 있다. 따라서 네트워크의 상태가 경부하이면 수신측 버퍼는 오버런이 발생할 수 있고, 네트워크의 상태가 과부하이면 수신측 버퍼는 스타베이션이 발생할 수 있다. 이러한 문제를 해결하기 위하여 그림 2와 같이 임계구역에 대한 개념으로 클라이언트 버퍼에 오버런 경고구간과 스타베이션 경고구간을 설정한다. 이러한 경고구간의 설정은 서버에게 보낼 전송을 제어 메시지의 생성 시점과 미디어 스케일링 정책의 응답성을 빠르게 하기 위한 지표로 활용되며, 또한 클라이언트 자원의 상태를 나타낸다.

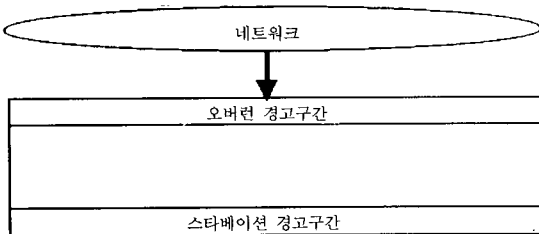


그림 2. 클라이언트 버퍼의 구조

3.2.1 초기 지연시간과 버퍼 체크시간

VOD 서버에서 초기의 스타베이션을 방지하기 위하여 처음 전송된 미디어 스트림을 바로 출력하지 않고 지정된 시간만큼 버퍼링한 다음 출력하게 된다. 이러한 버퍼링을 위한 지정된 시간을 초기 지연시간

이라고 한다. 본 논문에서의 초기 지연시간은 미디어 스트림이 클라이언트의 버퍼에 50%가 될 때까지 누적시키는데 걸리는 시간이다.

본 논문에서는 HeiTS에서 처럼 스트림이 도착할 때마다 도착상태를 분석하지 않고 BCT 단위마다 클라이언트 버퍼의 상태에 따라서 네트워크의 상태를 분석하고 예측하여 전송을 제어 메시지를 서버의 전송 관리자에게 보내며, 또한 미디어 스케일링 정책의 지속 시간도 경험적인 방법이 아닌 BCT 단위를 기준으로 수행한다.

BCT 단위의 산출은 네트워크로부터 전송되는 미디어 스트림이 클라이언트 버퍼의 50% 까지 입력되고, 클라이언트에서는 그 미디어 스트림에 대한 처리를 하지 않으므로 네트워크의 경부하 상태에서 발생할 수 있는 worst-case로 볼 수 있다. 즉, 버퍼링을 위한 초기 지연시간 중에 버퍼의 상태가 미리 설정한 스타베이션 경고구간에 도달할 때의 시간을 BCT로 결정한다.

BCT 단위를 설정하는 이유는 가변적인 네트워크의 상황을 분석하고 클라이언트의 버퍼 상태를 예측하기 위한 시간적 근거를 마련하기 위함이다. 또한 이 BCT는 미디어 스케일링 정책의 지속 시간이 되며, 특히 미디어 스트림이 도착할 때마다 매번 수신측 버퍼를 검사하지 않고 BCT 단위마다 검사함으로써 오버헤드를 줄일 수 있다.

3.2.2 버퍼 구역성

전송을 제어 메시지의 발생 및 미디어 스케일링 정책의 수행을 위해서는 클라이언트 버퍼가 경고구간에 도달하기 전에 자원 부족에 대한 상황을 인식하는 정보가 필요하다. 이러한 정보의 산출 방법으로서 버퍼의 구역성(buffer locality section) 개념을 이용한 지역구간과 추이구간을 산출하여 이용한다. 버퍼 구역성이란 그림 3에서 버퍼의 데이터량이 증가하는 기울기가 $\pm\alpha$ 범위 내에 있다는 것이다.

BCT_i 시점에서 클라이언트의 버퍼 크기는 버퍼에 입력된 미디어 데이터량의 총합에서 미디어 재생기에서 playout된 미디어 데이터량의 총합을 뺀 나머지 양이다. 그리고 버퍼의 크기가 B2(50%)를 초과하는 시점 이후부터 고정 길이의 매 BCT_i(i=1, 2, 3,, n)마다 계속적으로 버퍼 크기인 B_{BCT_i} 를 체크한다. 그리고 한 BCT 동안 네트워크를 통하여 클라이언트

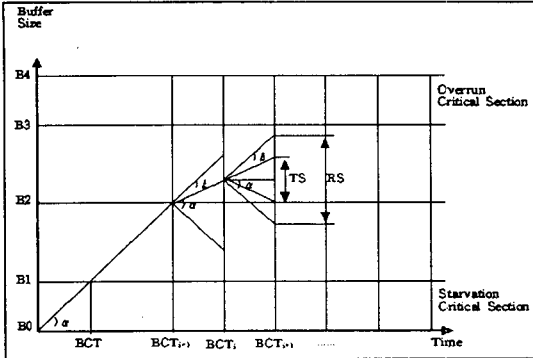


그림 3. 버퍼의 지역구간과 추이구간

버퍼에 도착한 미디어 데이터의 양과 클라이언트의 미디어 재생기에서 처리된 양에 대한 차는 한 BCT 동안에 클라이언트 버퍼의 변화량 ($B_{BCT_i}^A$)으로서 식 3-3과 같이 구할 수 있다.

$$B_{BCT_i}^A = \sum_{BCT_{i-1}}^{BCT_i} Input_Data - \sum_{BCT_{i-1}}^{BCT_i} Playout_Data \quad (식\ 3-3)$$

이 식에서 $\sum_{BCT_{i-1}}^{BCT_i} Input_Data$ 는

한 BCT 동안 네트워크를 통하여 클라이언트 버퍼에 도착한 미디어 데이터의 양으로서 네트워크의 가변적인 상태와 관계가 있으며,

$\sum_{BCT_{i-1}}^{BCT_i} Playout_Data$ 는 한 BCT 동안 클라이언트의 미디어 재생기에서 처리된 양으로서 미디어 재생기의 처리율과 관계가 있다.

식 3-3에서 구한 버퍼의 변화량이 0이라면 클라이언트 버퍼에 도착한 미디어 데이터의 양과 미디어 재생기에서 처리된 양이 같을 때이다. 이 경우는 클라이언트 버퍼의 증감에 대한 기울기가 0으로서 오버런이나 스태이션이 발생되지 않고 안정된 상태로 유지됨을 의미한다. 그리고 버퍼의 변화량이 0보다 큰 경우와 작은 경우에 있어서의 오버런 발생과 스태이션 발생에 관하여 식 3-4, 식 3-5와 같이 각각 나타낼 수 있다.

오버런 발생

($\sum Input_Data > \sum Playout_Data$ 인 경우):

$$B2 + |\sum B_{BCT_i}^A| \geq B3 \quad (식\ 3-4)$$

스타이션 발생

($\sum Input_Data < \sum Playout_Data$ 인 경우):

$$B2 - |\sum B_{BCT_i}^A| \leq B1 \quad (식\ 3-5)$$

식 3-4는 버퍼의 중간 크기 값(B2)에다 BCT_i에서 BCT_i까지 변화량들에 대한 합의 절대 값을 더한 값이 오버런 경고구간(B3) 보다 크거나 같은 경우이므로 버퍼는 오버런이 발생할 수 있으며, 식 3-5는 버퍼의 중간 크기 값(B2)에서 변화량들에 대한 합의 절대 값을 뺀 값이 스태이션 경고구간(B1) 보다 작거나 같은 경우이므로 버퍼는 스태이션이 발생할 수 있다.

버퍼링을 위한 초기 지연시간 동안(B2) 버퍼량의 증가에 대한 기울기 α 는 네트워크가 경부하 상태에서 발생할 수 있는 worst-case경우에 대한 기울기이다.

따라서, BCT_i에서 예측 구간인 BCT_{i+1}에 대한 버퍼의 상태를 버퍼 구역성을 이용하여 예측할 수 있다. 즉, 예측 구간인 BCT_{i+1} 시점에서의 버퍼 크기는 $B_{BCT_i} \pm \Delta\alpha$ 범위 내에 속한다. 이것은 네트워크의 worst-case인 과부하 및 경부하 상태를 모두 포함한 값이다.

지역구간(RS: Region Section)은 BCT_i에서 검사한 클라이언트 버퍼의 크기인 B_{BCT_i} 에 대하여 BCT_{i+1}에서 발생할 수 있는 worst-case의 증감 기울기 $\pm\alpha$ 를 적용한 범위이다. 그리고 $\pm\Delta\alpha$ 는 기울기 $\pm\alpha$ 에 대한 데이터의 증감에 대한 변화량이다. 그림 3의 RS는 지역구간으로서 이것은 네트워크의 경부하 및 과부하 상태에서 전송을 제어 메시지 발생 시점 및 미디어 스케일링 정책의 응답성을 높이는 지표로 사용되며 식 3-6에 의하여 구할 수 있다.

$$\text{지역구간: } RS = B_{BCT_i} \pm \Delta\alpha \quad (식\ 3-6)$$

클라이언트 버퍼의 상태는 네트워크의 상태와 비례 관계가 있으며 또한 클라이언트의 미디어 처리율과 비례 관계가 있으므로 BCT_i에서 버퍼의 크기 B_{BCT_i} 와 BCT_{i-1}에서 버퍼의 크기 $B_{BCT_{i-1}}$ 에 대한 차이로서 증감 폭을 예측 계산할 수 있다. 따라서 이 증감 폭에 대한 기울기가 $\pm\beta$ 이면 다음 번의 버퍼 체크 구간인 BCT_{i+1}에서 예측할 수 있는 버퍼의 크기는 B_{BCT_i} 에 $\pm\beta$ 를 적용한다. 여기서 $\pm\beta$ 는 기울기 $\pm\beta$ 에 대한 데이터의 증감에 대한 변화량이다

다. 이 범위는 BCT_{i+1} 에서의 추이구간(TS: Transition Section)이다. 따라서 BCT_{i+1} 에서의 추이구간은 그림 3의 TS이며, 식 3-7에 의하여 구할 수 있다.

$$\text{추이구간: } TS = B_{BCT_i} \pm \Delta\beta \quad (\text{식 3-7})$$

3.3 미디어 스케일링 정책

미디어 스케일링 정책은 BCT마다 지역구간과 추이구간을 산출하여 클라이언트 버퍼의 현재 상태, 지역구간 그리고 추이구간이 임계구역에 속하는가에 따라 scaling up 또는 scaling down을 수행한다. 그리고 미디어 스케일링의 강도를 3단계로 구분하여 수행한다[15,16].

(1) Scaling up

미디어 스트림의 전송율이 일정 수준 이하로 되면 클라이언트 버퍼는 스타베이션이 발생할 수 있다. 이때 미디어 스트림에 대하여 scaling up을 수행하여 스타베이션 발생 시간을 지연시킨다.

버퍼 체크 구간 BCT_{i+1} 에서의 지역구간이 스타베이션 경고구간에 속하고, 추이구간은 스타베이션 경고구간에 속하지 않는 경우에는 강도 3의 scaling up을 수행하며, BCT_{i+1} 에서의 지역구간과 추이구간 모두 스타베이션 경고구간에 속할 경우에는 강도 5의 scaling up을 수행한다. 그리고 앞의 두 가지 강도(강도 3과 강도 5)에 의해 scaling up을 수행했음에도 불구하고 현재의 버퍼 상태가 스타베이션 경고구간에 속하는 경우에는 강도 7의 scaling up을 수행한다.

(2) Scaling down

미디어 스트림의 전송율이 일정 수준 이상이 되면 클라이언트 버퍼는 오버런이 발생할 수 있다. 이때 미디어 스트림에 대하여 scaling down을 수행하여 오버런 발생 시간을 지연시킨다.

버퍼 체크 구간 BCT_{i+1} 에서의 지역구간이 오버런 경고구간에 속하고, 추이구간은 오버런 경고구간에 속하지 않는 경우에는 강도 3의 scaling down을 수행한다. 그리고 BCT_{i+1} 에서의 지역구간과 추이구간 모두 오버런 경고구간에 속할 경우에는 강도 5의 scaling down을 수행한다. 그리고 앞의 두 가지 강도(강도 3과 강도 5)에 의해 scaling down을 수행했음에도 불구하고 현재의 버퍼 상태가 오버런 경고구간에 속하는 경우에는 강도 7의 scaling down을 수행

한다.

본 논문에서는 미디어 스케일링 정책과 병행하여 서버의 전송율을 제어하는 전송율 제어 정책을 병행 수행함으로써 클라이언트 버퍼의 오버런 또는 스타베이션을 방지하고 비디오 데이터의 질을 보장하도록 하였다.

3.4 전송율 제어 알고리즘

제어할 전송율의 계산은 오버런 방지를 위한 전송율 계산과 스타베이션 방지를 위한 전송율 계산으로 나누어 기술한다.

3.4.1 오버런 방지를 위한 전송율 계산

클라이언트 버퍼의 예측한 상태가 오버런 경고구간을 넘는 경우에는 서버에게 전송 지연을 요청하는 제어 메시지를 보내어 VOD 서버의 전송율을 적절히 조절하게 함으로써 클라이언트의 버퍼는 오버런 발생을 방지 할 수 있다.

그림 3에서 클라이언트 버퍼의 초기 지연 시간 동안에 스타베이션 경고구간(즉, $B1$)을 초과하는 시간을 버퍼 체크 시간인 BCT 로 결정한다.

그리고 이 BCT 에서는 클라이언트 버퍼에 입력되는 미디어 데이터 블록들에 대한 갯수를 카운터하며 그 갯수는 M_{BCT}^{count} 로 나타낸다. 따라서 BCT 동안 입력된 미디어 데이터 블록들의 평균 크기(s_{vj})는 식 3-8에 의하여 구할 수 있다.

$$s_{vj} = \left[\frac{\Delta a}{M_{BCT}^{count}} \right] \quad (\text{식 3-8})$$

이 식에서 Δa 는 클라이언트 버퍼의 초기 지연 시간 동안 스타베이션 경고구간(즉, $B1$)을 초과하는 시간에서 미디어 데이터의 변화량이며, M_{BCT}^{count} 는 앞에서 언급한 바와 같이 클라이언트 버퍼에 입력되는 미디어 데이터 블록들을 카운터한 갯수이다. 따라서 식 3-8은 BCT 동안에 네트워크로부터 클라이언트 버퍼에 도착된 미디어 데이터 블록의 갯수와 평균 크기를 계산해 낼 수 있다.

서버의 전송 관리자는 미디어 데이터 블록 단위(즉, 미디어 프레임 단위)로 네트워크를 통하여 클라이언트 시스템으로 전송하며, 각 미디어 데이터 블록의 전송 주기 시간은 T_{trans} 라고 한다. 이때 클라이언트의 버퍼 상태를 체크하는 시간 단위인 한 BCT

에 대한 서버의 전송 주기 시간도 계산할 수 있다. 즉, 한 BCT 동안에 n 개의 미디어 프레임이 클라이언트의 버퍼에 도착하였다면 서버는 $n * T_{trans}$ 의 전송 주기 시간을 진행한 것이다.

그림 3에서 예측 구간인 BCT_{i+1} 에서의 버퍼 크기 $B_{BCT_{i+1}}$ 는 식 3-7에서 언급한 추이구간(TS)의 정의와 동일하며 식 3-9와 같이 나타낼 수 있다.

$$B_{BCT_{i+1}} = TS = B_{BCT_i} + \Delta\beta \quad (\text{식 3-9})$$

이 식에서 B_{BCT_i} 는 BCT_i 에서의 버퍼 크기이며, $\Delta\beta$ 는 식 3-7의 추이구간에서 언급한 바와 같이 BCT_{i+1} 에서 $\Delta\beta$ 만큼 증감하게 될 데이터의 변화량이다.

클라이언트의 버퍼 크기는 네트워크를 통해 클라이언트의 버퍼에 입력되는 양과 비례 관계가 있으며 또한 클라이언트의 미디어 처리율과 비례 관계가 있다. 따라서 버퍼 체크 구간인 BCT_{i+1} 에서 예측할 수 있는 클라이언트 버퍼 크기($B_{BCT_{i+1}}$)는 식 3-9와 같다.

그리고 BCT_{i+1} 에서 예측되는 블록의 갯수 $M_{BCT_{i+1}}$ 을 구하는 식은 식 3-10과 같다.

$$M_{BCT_{i+1}} = \left\lceil \frac{|\Delta\beta|}{S_{vf}} \right\rceil \quad (\text{식 3-10})$$

BCT_i 에서 예측한 BCT_{i+1} 에서의 예측 기울기가 오버런 경고구간을 초과 할 경우에는 클라이언트 버퍼의 오버런이 발생할 수 있다. 오버런을 방지하기 위하여 버퍼의 증가 기울기 +b를 $-b/2$ 로 한다. 즉, 예상되는 데이터 증가량인 h 만큼을 전송 지연시킨다. 이 h 는 BCT_{i+1} 에서 예측되는 데이터 블록 갯수인 $M_{BCT_{i+1}}$ 이며, 그 사이즈는 $M_{BCT_{i+1}} * s_{vf}$ 이다. 따라서 클라이언트는 $M_{BCT_{i+1}}$ (BCT_{i+1} 에서 예측되는 데이터 블록 갯수)에 대한 서버의 전송 시간만큼 전송 지연을 요구하는 제어 메시지를 서버에게 보낸다. 한 BCT 동안 증가되는 미디어 데이터 블록의 갯수가 $M_{BCT_{i+1}}$ 이므로 이 갯수에 대한 서버의 전송 지연시간 $M_{BCT_{i+1}}^{time}$ 은 BCT와 동일한 시간이며 식 3-11과 같다.

$$M_{BCT_{i+1}}^{time} = BCT \quad (\text{식 3-11})$$

식 3-11의 $M_{BCT_{i+1}}^{time}$ 는 클라이언트의 버퍼 체크 시간인 BCT 동안(BCT_i 에서 BCT_{i+1} 까지) 서버의 데이터 전송을 지연시킬 시간이다. 그리고 하나의 BCT에 대한 미디어 데이터 블록 갯수는 서버의 전송 관

리자가 전송하는 전송 주기의 수와 같다. 즉, 앞의 식 3-8에서 정의한 바와 같이 BCT에 대한 미디어 데이터 블록 갯수 M_{BCT}^{count} 는 이 BCT 동안에 클라이언트 버퍼에 도착한 미디어 데이터 블록의 갯수이기도 하고 또한 서버의 전송 주기의 수이기도 하다. BCT_{i+1} 에서 예측되는 미디어 데이터 블록의 갯수가 n 이라고 하면 n 은 한 BCT 동안(즉, BCT_i 에서 BCT_{i+1} 까지) 서버의 전송 관리자가 미디어 데이터 블록들을 전송하는 전송 주기의 수가 된다. 그리고 BCT_i 시점에서 서버에게 요청하는 하나의 미디어 데이터 블록에 대한 평균 전송 지연시간은 T_{BCT} 로 나타내며, 식 3-12에 의하여 구할 수 있다.

$$T_{BCT_i} = \frac{M_{BCT_{i+1}}^{time}}{M_{BCT_{i+1}}} \quad (\text{식 3-12})$$

따라서 서버의 지연전송 주기시간(T_{delay})은 식 3-13에 의하여 구할 수 있다.

지연전송 주기시간(T_{delay}):

$$T_{delay} = T_{trans} + T_{BCT_i} \quad (\text{식 3-13})$$

결과적으로 한 BCT 동안(즉, BCT_i 에서 BCT_{i+1} 까지) 서버의 전송을 제어는 한 미디어 데이터 블록에 대한 전송 주기시간인 T_{trans} 에서 T_{BCT_i} 시간만큼 지연된 전송 주기시간(즉, T_{delay})으로 예측되는 데이터 블록의 갯수($M_{BCT_{i+1}}$)와 동일한 횟수만큼 진행함으로써 서버의 전송율을 제어한다.

이와 같이 서버의 전송율을 제어하는 것은 BCT_{i+1} 시점에서의 예측되는 버퍼 증가량에 대한 기울기 +b를 $-b/2$ 로 BCT_i 시점에서 꺾어 내리게 하는 효과이다.

전송 지연은 항상 연속성 요구 조건을 만족하는 범위 내에서 이루어 져야 한다. 클라이언트의 일정한 재생율에 대하여 현 버퍼 체크 시점(BCT_i)에서의 버퍼량에서 다음 번의 버퍼 체크 시점(BCT_{i+1})에서 playout될 양과 전송 지연시킬 블록의 데이터량을 뺀 버퍼의 잔량이 재생되는 미디어 양보다 많아야 한다는 것이다. 즉, 클라이언트 버퍼의 BCT에 상응하는 서버의 각 전송 주기시간 T_{trans} 에서 T_{BCT_i} 만큼의 시간을 지연시켜 전송하더라도 연속성 요구 조건을 만족하여야 한다.

3.4.2 스타베이션 방지를 위한 전송율 계산

클라이언트 버퍼의 예측되는 상태가 스타베이션

경고구간을 넘는 경우에는 서버에게 긴급 전송을 요청하는 제어 메시지를 보내어 VOD 서버의 전송을 적절히 조절하게 함으로써 클라이언트의 버퍼는 스타베이션 발생을 방지 할 수 있다.

그리고 서버의 전송을 제어 즉, 긴급 전송은 멀티미디어 데이터의 연속성 요구 조건을 만족하는 범위 내에서 이루어져야만 클라이언트의 연속 재생이 보장된다.

클라이언트의 버퍼 스타베이션을 방지하기 위한 긴급 전송의 계산 과정은 오버런을 방지하기 위한 전송을 계산 과정과 유사하며 서버에서 미디어 데이터 블록에 대한 긴급 전송을 하게 한다.

한 BCT 동안에 긴급 전송시킬 미디어 데이터 블록의 갯수가 $M_{BCT_{i+1}}$ 이므로 이것에 대한 서버의 긴급 전송시간 $M_{BCT_{i+1}}^{time}$ 은 BCT와 동일한 시간이며 계산 식은 앞 절의 오버런 방지를 위한 전송을 계산에서와 동일하다. 즉, 식 3-11에 의해 계산할 수 있다.

그리고 BCT_i 시점에서 서버에게 요청하는 하나의 미디어 데이터 블록에 대한 평균 긴급 전송시간 T_{BCT_i} 의 계산식은 아래의 식 3-14로 구할 수 있다.

긴급전송 주기시간($T_{urgency}$):

$$T_{urgency} = T_{trans} - T_{BCT_i} \quad (\text{식 3-14})$$

따라서 한 BCT 동안 서버의 전송을 제어는 한 미디어 데이터 블록에 대한 전송 주기시간인 T_{trans} 에서 T_{BCT_i} 시간만큼 앞당긴 긴급전송 주기시간(즉, $T_{urgency}$)으로 예측되는 미디어 데이터 블록의 갯수($M_{BCT_{i+1}}$)와 동일한 횟수만큼 진행함으로써 긴급 전송이 되게 하는 것이다.

이와 같이 서버에게 긴급 전송을 요청하는 것은 BCT_{i+1} 시점에서의 예측되는 버퍼 증가량에 대한 기울기 -b를 +b/2로 BCT_i 시점에서 꺾어 올리게 하는 효과이다.

긴급 전송은 항상 연속성 요구 조건을 만족하는 범위 내에서 이루어 져야 한다. 클라이언트의 일정한 재생율에 대하여 현 버퍼 체크 시점(BCT_i)에서의 버퍼량에서 다음 번의 버퍼 체크 시점(BCT_{i+1})에서 playout될 양을 빼고 긴급 전송시킬 블록의 데이터량을 더한 버퍼의 잔량이 재생되는 미디어 양보다 많아야 한다는 것이다. 즉, 클라이언트의 BCT에 상응하는 서버의 각 전송 주기에 T_{BCT_i} 만큼의 시간을 앞당겨 전송하더라도 연속성 요구 조건을 만족하

여야 한다.

3.4.3 알고리즘 및 VOD 시스템 모델

본 논문에서 제안하는 알고리즘은 단일 스트림을 위한 것으로서, 첫 번째 루틴은 클라이언트 버퍼에 입력되는 미디어 데이터 블록들에 대한 갯수를 카운터하며 버퍼의 크기가 스타베이션 경고구간을 초과하는 시간을 버퍼 체크 시간인 BCT로 결정한다. 그리고 지역구간을 결정하기 위해 $\Delta\alpha$ 도 산출한다.

두 번째 루틴은 서버의 전송 관리자로부터 네트워크를 통하여 전송되는 미디어 데이터를 초기 지연 시간동안 버퍼링하며 초기 지연 시간 이후부터는 각 미디어 데이터에 대하여 디바이스 별로 서비스를 시작한다.

```

Label-1:
Input_data_count := Input_data_count + 1;
Current_Buffer_Size := Current_Buffer_Size
                    + Input_data_Size;
If Current_Buffer_Size ≥ Starvation_Caution
    BCT := Current_Time;
    Δα 산출;
    Average_Frame_Size := Current_Buffer_Size
                        / Input_data_Size;
Else
    Go to Label-1;
    
```

(1) BCT, $\Delta\alpha$, 평균 프레임 크기 산출

```

Label-2:
Current_Buffer_Size := Current_Buffer_Size
                    + Input_data_Size;
If Current_Buffer_Size ≥ Optimal_Buffer_Size
    각 디바이스에 대한 서비스 시작
Else
    Go to Label-2;
    
```

(2) 초기 지연 시간동안 버퍼링

세 번째 루틴에서는 클라이언트 버퍼의 스타베이션을 방지하기 위한 전송을 계산을 기술하고 있다. 본 논문에서의 VOD 시스템 모델은 그림 3.3과 같

다. 클라이언트 시스템의 스케줄러는 버퍼 제어기와 미디어 처리기로 나누어진다.

버퍼 제어기는 BCT 단위마다 클라이언트 버퍼의 상태를 검사하여 예측되는 버퍼의 상태가 스타베이션 또는 오버런 경고구간 내에 있는 경우 서버의 전송 관리자에게 전송을 제어 메시지를 보내어 미디어 데이터 블록에 대한 전송 주기 시간을 제어하게 함으로써 예측되는 클라이언트 버퍼의 상태는 스타베이션 또는 오버런 경고구간 내에 들지 않으면서 적정량의 데이터가 유지되도록 한다.

```

RS := BBCTi + Δα; { BCTi+1의 지역구간 계산 }
TS := BBCTi + Δβ; { BCTi+1의 추이구간 계산 }
If(RS ≤ Starvation_Caution) or (TS ≤ Starvation_Caution)
Then
  Begin
    MBCTi+1 := ⌈  $\frac{|\Delta\beta|}{s_{uf}}$  ⌋ ;
    { BCTi+1에서 감소 예측되는 미디어 데이터
      블록의 평균 갯수 계산 }
    MBCTi+1time := BCT;
    { BCTi+1에서 감소 예측되는 미디어 데이터
      블록의 갯수에 대한 전송 시간 계산 }
    TBCTi :=  $\frac{M_{BCT_{i+1}}^{time}}{M_{BCT_{i+1}}}$ ;
    { 한 미디어 데이터 블록에 대한 평균 전송시간 }
    Turgency := Ttrans - TBCTi;
    { 한 미디어 데이터 블록의 긴급전송 주기시간 계산 }
    Control_Message 전송;
    { 긴급전송 주기시간( Turgency)이 한 BCT동안 예측
      되는 미디어 데이터 블록의 갯수( MBCTi+1)와 동
      일한 횟수만큼 진행되도록 하는 제어 메시지를 전송
    }
    Media_Scaling_Up 수행(미디어 처리율 감소)
  End
Else
  정상 서비스 수행;

```

(3) 스타베이션 방지를 위한 전송을 계산

미디어 처리기는 각 미디어 출력 장치와 연결되어 있으며, 버퍼 제어기에서 생성한 전송을 제어 메시지에 의해 서버의 전송 주기 시간을 제어함에도 불구하고 클라이언트 버퍼의 예측되는 상태가 계속 스타베이션 또는 오버런 경고구간 내에 있는 경우에 대하여 미디어 스케일링 강도를 결정하고 그 강도에 따라 미디어 스케일링을 수행한다.

4. 시뮬레이션 및 분석

4.1 시뮬레이션 환경

본 논문에서의 시뮬레이션은 PC-SCHEME으로 작성된 DEVS (Distributed Event System Specification)[17]를 기반으로 수행하였으며, 시뮬레이션 모델은 그림 5와 같다

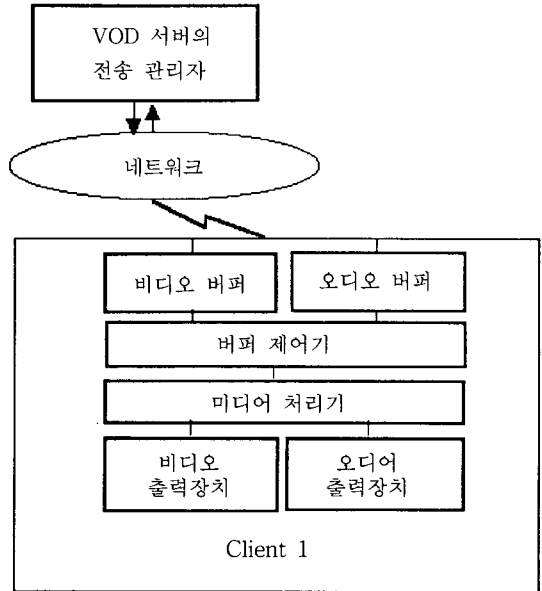


그림 4. VOD 시스템 모델

EF-Ser는 하위 계층의 EF와 Mul-Arch 화일을 로드시키고, EF와 Mul-Arch사이의 포트 연결을 정의한다.

EF는 Genr(Server)와 Transd 사이의 포트 연결과 EF와 상위 계층인 EF-Ser 사이의 포트 연결을 정의한다.

Genr은 지수 분포로 네트워크의 자연 값과 미디어 데이터의 가변적인 크기를 생성하여 클라이언트에 보내며, 클라이언트에서 발생시킨 전송을 제어 메시지를 접수하여 Genr의 데이터 전송율을 조정한다.

Transd는 클라이언트와 각 미디어 별 출력 장치 (Dev1과 Dev2) 사이의 상태 변이를 상위 계층인 EF와 연결된 포트를 통하여 접수하고 기록한다. 그리고 Transd는 전체 처리시간이 90000이 되면 Genr와 연결된 포트를 통하여 STOP이라는 메시지를 보낸다.

Mul-Arch는 클라이언트와 Dev1, Dev2사이의 포트 연결과 상위 모듈인 EF-Ser를 통하여 전달된 미디어 데이터를 클라이언트에게 전달하는 포트를 정의하고 있으며, 또한 클라이언트에서 발생한 전송율 제어 메시지를 Genr(Server)에 전달하기 위하여 EF-Ser로 전달하는 포트를 정의한다.

클라이언트는 Genr에서 발생한 각 미디어 데이터를 완전히 처리하고 Passive 상태가 된다. 클라이언트 시스템은 도착한 미디어 데이터를 초기 지연 시간 동안 각 출력 장치의 버퍼 큐에 저장하는데 이것은 네트워크의 지연이나 미디어 데이터에 대한 지터(jitter) 영향을 흡수 위함이며 또한 초기의 스태이션을 방지하기 위해서 이다. 그리고 버퍼의 크기가 허용치 이상이 되는 시점에서부터 클라이언트 시스템은 미디어 스트림에 대한 서비스를 개시하며, 매 BCT 마다 버퍼의 크기를 검사하여 예측되는 버퍼의 상태가 오버런 또는 스태이션 경고구간에 있게 되면 전송율 제어 메시지를 발생하여 서버의 데이터 블록 전송 간격을 조정하는 것과 미디어 스케일링을 병행 수행한다. 미디어 스케일링은 프레임의 드롭핑 또는 미디어의 처리율 조절에 의한다.

클라이언트는 각 출력 장치를 할당된 처리시간을 기준으로 블록킹하거나 이를 해제하는 메커니즘을 가진다. 즉, 시스템 자원을 각 미디어 별 출력 장치에

게 상호 배제적으로 할당하여 처리한다.

본 논문의 시뮬레이션 환경에서 사용한 파라미터들을 정리하면, 클라이언트 버퍼의 크기는 4096 KBytes이며, BCT를 결정하는데 기준이 되는 스태이션 경고구간의 크기는 512 KBytes이다.

본 논문에서 스태이션 경고구간의 설정은 시뮬레이션을 통한 절충안을 선택하였다. 만약, 스태이션 경고구간의 크기를 버퍼의 25%로 설정하고 초기 지연시간을 버퍼의 크기 50%까지 초과하는 경우로 설정하면 데이터 공급율과 데이터 소비율이 정확히 맞지 않는 경우에는 전송율 제어 정책 및 미디어 스케일링 정책이 수행된다. 따라서 초기 지연시간을 버퍼의 크기 50%까지 초과하는 경우로 설정할 경우에는 스태이션 경고구간을 25%이하로 반드시 설정해야 한다. 따라서 본 논문에서는 스태이션 경고구간의 크기를 25%의 절반인 12.5%를 기준으로 16%와 8%로 각각 설정하여 시뮬레이션 하였다.

표 1은 버퍼의 전체 크기(4096 KBytes)에 대하여 스태이션 경고구간의 크기에 따른 BCT 및 $\Delta\alpha$ (스태이션 경고구간을 초과하는 시간에서의 기울기 α 에 대한 데이터 변화량)를 나타낸 것이다.

표 1. 스태이션 경고구간의 크기에 따른 BCT

스타이션 경고구간 (KBytes)	(%)	BCT (ms)	$\Delta\alpha$ (KBytes)
327	8	607	352
512	12.5	878	543
655	16	1191	665

4.2 분석 및 평가

4.2.1 성능 평가

가변적인 네트워크의 부하변동, 서버의 전송율 변화 그리고 클라이언트의 처리율 변화로 인하여 클라이언트 버퍼는 스태이션 또는 오버런이 발생할 수 있다. 본 논문에서는 이러한 현상을 방지하며 안정된 버퍼 상태를 유지하기 위하여 클라이언트 버퍼의 예측 상태 및 현재의 버퍼 상태가 스태이션 또는 오버런 경고구간에 속할 때 전송율 제어 메시지를 서버에게 전송하여 전송 주기 시간을 제어한다. 그리고 미디어 스케일링을 병행 수행함으로써 클라이언트 버퍼의 상태는 스태이션 또는 오버런 경고구간 내에

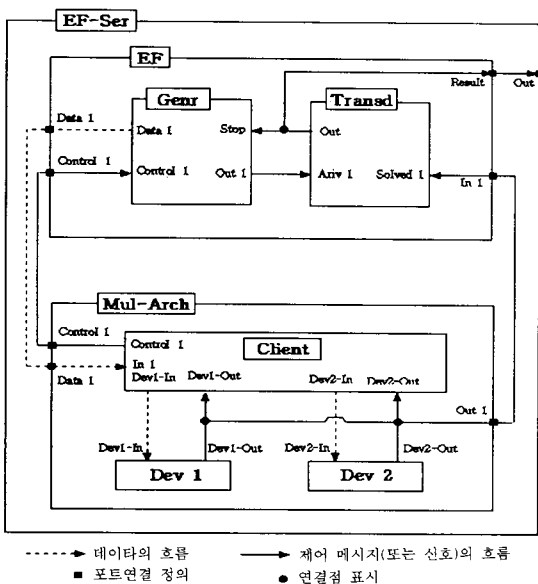


그림 5. 시뮬레이션 모델

들지 않으면서 적정량의 데이터가 유지되도록 한다.

클라이언트의 실제 버퍼 크기는 네트워크의 worst-case인 WU(지역구간의 상한)와 WD(지역구간의 하한)구간에 항상 존재하며, 실제 버퍼 크기는 가변적인 네트워크의 부하 변동으로 인하여 PU(추이구간의 상한)와 PD(추이구간의 하한)구간을 일시적으로 벗어나는 경우가 있음을 확인할 수 있었다. 그리고 클라이언트 버퍼의 실제 크기는 스태이션 경고구간인 512와 오버런 경고구간이 3584를 초과하지 않으며, 오실레이션의 발생이 현저하게 완화되었음을 확인할 수 있다.

4.2.2 스태이션 상황에 대한 성능 평가

시뮬레이션 시간 9683에서 네트워크의 worst-case 예측 구간인 WD가 512이하로 떨어졌고, 클라이언트는 미디어 스케일링을 수행하면서 3.4.3에서 언급한 알고리즘으로 전송 제어를 -2.8을 생성하여 서버로 전송하였다.

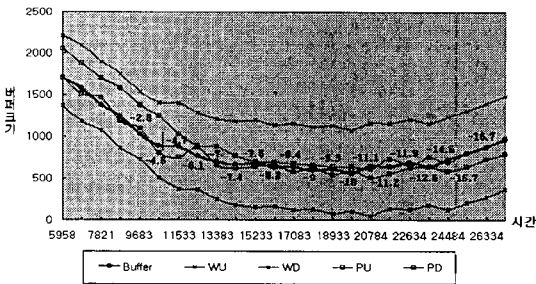


그림 6. 스태이션 상황에 대한 성능평가

다음 BCT에서도 WD가 스태이션 경고구간에 속해 있으므로 클라이언트는 전송을 제어 메시지를 계속 발생하며, 전송을 최대 -16.7까지 생성하여 서버로 전송함으로써 WD가 스태이션 구간에서 벗어나고 있음을 보여 주고 있다. 그림 6의 그래프 상에 나타난 수치들은 클라이언트 버퍼의 스태이션을 방지하기 위해 BCT_i 시점에서 서버에게 요청하는 긴급 전송 시간 T_{BCT_i} 에 대한 누적 값이다.

4.2.3 오버런 상황에 대한 성능 평가

시뮬레이션 시간 45881에서 네트워크의 worst-case 예측 구간인 WU가 3584를 초과하였고, 클라이언트는 미디어 스케일링을 수행하면서 전송 제어율 +1을 생성하여 서버로 전송하였다.

다음 BCT에서도 WU가 오버런 경고구간에 속해 있으므로 클라이언트는 전송을 제어 메시지를 계속 발생하였고 전송을 최대 5.6까지 생성하여 서버로 전송함으로써 WU가 오버런 경고구간에서 벗어나고 있음을 보였다.

그림 7의 그래프 상에 나타난 수치들은 클라이언트 버퍼의 오버런을 방지하기 위해 BCT_i 시점에서 서버에게 요청하는 지연전송 시간 T_{BCT_i} 에 대한 누적 값이다.

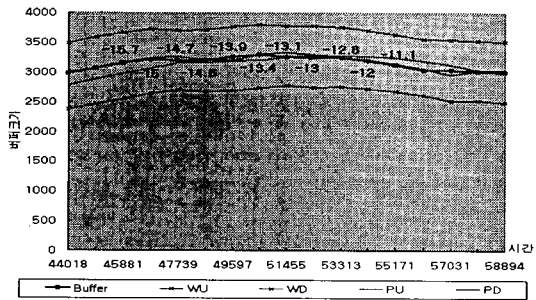


그림 7. 오버런 상황에 대한 성능평가

WU가 3584구간 이하로 떨어지면서 더 이상의 전송을 제어 메시지를 발생하지 않고 있다. 서버는 초기에 비디오 프레임의 전송주기 시간을 33ms로 설정하였으나 한번의 스태이션 상황과 오버런 상황을 겪으면서 데이터 블록의 전송 주기 시간을 21.9ms로 조정되었다.

4.2.4 전송율의 제어 효과

본 논문에서 제안한 전송을 제어 알고리즘의 실행으로 네트워크의 부하변동으로 인하여 발생하는 클라이언트 버퍼의 오버런과 스태이션 현상을 클라이언트 버퍼의 상태에 따라 서버의 데이터 전송 주기 시간을 적응적으로 조절하여 스태이션과 오버런 발생 문제를 해결할 수 있었다.

그림 8에서 서버가 21.9ms의 전송 주기 시간으로 전송함에 따라 클라이언트의 실제 버퍼는 2500으로 안정화되고 있음을 알 수 있다.

5. 결론

멀티미디어 데이터를 실시간으로 서비스하기 위해서는 서버의 전송율과 버퍼공간 및 네트워크의 가

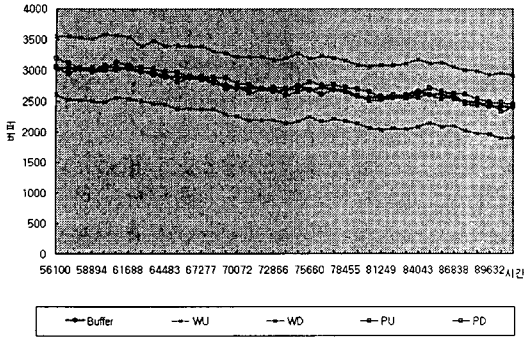


그림 8. 전송율의 제어 효과

용 능력 등이 서로 많은 영향을 미치기 때문에 동시에 고려되어야 한다.

본 논문은 VOD 서비스 시스템에서 서버의 데이터 전송율과 클라이언트의 데이터 소비율 및 클라이언트 버퍼의 상태 변화를 기반으로 하여 서버의 전송율을 조정하는 전송율 제어 알고리즘을 설계하고 분석하였다.

제안 알고리즘은 클라이언트 시스템의 스케줄링 부하를 줄이기 위하여 버퍼의 상태를 일정 시간마다 검사하는 BCT를 이용하며, 이 BCT를 기반으로 클라이언트 버퍼의 상태를 예측한다. 예측한 버퍼의 상태가 오버런 혹은 스타베이션 경고구간에 속할 경우에는 버퍼의 증감 크기에 따라 제어할 서버의 전송율을 산출하여 서버의 전송율을 적응적으로 조정하게 함으로써 클라이언트 버퍼의 오버런과 스타베이션이 발생되지 않도록 설계하였다.

서버에 대한 전송율 제어 정책을 수행하는 것만으로는 네트워크의 과부하 또는 경부하와 같은 가변적인 부하 변동에 클라이언트의 버퍼가 안정되지 못한다. 따라서 본 논문에서는 클라이언트 버퍼의 상태에 따라 서버에 대한 전송율을 제어하는 정책과 미디어 스케일링 정책을 병행 수행함으로써 클라이언트 버퍼의 상태를 안정화하며 또한 연속 재생을 보장하였으며, 오실레이션 발생 빈도가 현저히 줄어들음을 확인할 수 있었다.

향후, 전송율 제어 알고리즘에 관한 연구로서 다수의 클라이언트를 위한 VOD 서버의 전송 스케줄 및 승인제어 등에 관한 지속적인 연구가 요구되며, 또한 MPEG 데이터를 이용한 구현과 성능분석 및 그 효율성에 대한 지속적인 연구가 요구된다.

참 고 문 헌

- [1] D. J. Gemmell, H. M. Vin, D. D. Kandlur, and P. V. Rangan, "Multimedia Storage Servers: A Tutorial and Survey," *IEEE Computer*, Vol. 28, pp.40-49, 1995.
- [2] 이시진, 양택훈, 김영찬, "데이터 전송율과 소비율에 기반한 제한블록할당에서의 효율적 버퍼 크기 결정," *한국정보과학회 논문지(A) 제 23권 제 10호*, pp. 1028-1040, 1996년 10월.
- [3] P. V. Rangan, H. M. Vin, "Designing File System for Digital Video and Audio," *Proc. of the 13th ACM Symp. on Operating System Prin.*, pp. 81-94, Oct. 1991.
- [4] H. M. Vin, P. V. Rangan, "Designing a Multi-User HDTV Storage Server," *IEEE Journal on Selected Areas in Communicatio*, Vol. 11, No. 1, Jan. 1993.
- [5] P. V. Rangan, H. M. Vin, "Efficient Storage Techniques for Digital Continuous Multimedia," *IEEE Transactions on Knowledge and Data Engineering*, Aug. 1993.
- [6] D. J. Gemmell, "Multimedia Network File Server: Multi-Channel Delay Sensitive Data Retrieval," *ACM Multimedia '93*, PP. 243-250. 1993.
- [7]. D. J. Gemmell, "Delay Sensitive Multimedia in Disk-Based Environment," *IEEE Multimedia* PP. 56-66. 1994.
- [8] 이경호, 염현영, "디스크 전송율과 버퍼 크기를 동시에 고려한 효율적 수용제어 기법," *한국정보과학회 논문지(A) 제 24권 제 8호*, pp. 731-740, 1997.
- [9] D. R. Kenchamma-Hosekote and J. Srivastava, "Scheduling Continuous Media in a Video-On-Demand Server," *Proc. of the International Conference on Multimedia Computing and Systems*, IEEE Computer Society Press, pp. 19-28, May 1994.
- [10] L. Delgrossi, C. Halstrick, D. Hehmann, R. Guido Herrtwich, O. Krone, J. Sandvoss, and C. Vogt, "Media Scaling for Audiovisual Com-

munication with the Heidelberg Transport System," Proceedings ACM Multimedia 1993.

- [11] T. Kappner and L. C. Wolf, "Media Scaling in Distributed Multimedia Object Services," Second International workshop, IWACA '94, Springer- Verlag, pp.34-43, 1994.
- [12] L. Delgrossi, C. Halstrick, D. Hehmann, R.G. Herrtwich, O. Krone, J. Sandvoss, and C. Vogt, "Media Scaling for Audiovisual Communication with the Heidelberg Transport System," Proceedings ACM Multimedia, pp.99-104, 1993.
- [13] L. C. Wolf, R. G. Herrtwich, "The System Architecture of the Heidelberg Transport System," ACM Operating Systems Review, Vol. 28, No. 2, pp.51-64, April 1994.
- [14] J. Sandvoss, J. Winkler, H. Witting, "Network Layer Scaling: Congestion Control in Multimedia Communication with Heterogeneous Networks," IBM European Networking Center Heidelberg, 1994
- [15] Soheui Park, Beongcheol Moon, Wankyoo Kim, Zhixian Jin, and Kyooseok Park, "Design and Analysis of a Buffer Controller Using Buffer Locality," ICOIN-10, pp.85-93, Jan. 1996.
- [16] 박소희, "Buffer Locality와 Control Rate를 이용한 멀티미디어 시스템용 버퍼제어기의 설계 및 분석," 경남대학교 석사학위 논문, 1995.

- 17. B. P. Zeigler, "Object-Oriented Simulation with Hierarchical Modular Model," Academic Press, 1990.



김 완 규

경남대학교 컴퓨터공학과 (공학사·석사·박사)
 1988년~1989년 한국기계연구원 메카트로닉스과 연구원
 1995년~현재 한국정보과학회 영남지부 운영위원
 1998년~현재 한국멀티미디어학

회 편집위원

1998년~현재 진주산업대학교 산학협력처장
 1989년~현재 진주산업대학교 전자계산학과 부교수
 관심분야 : 운영체제, 멀티미디어시스템, 분산처리시스템, 정보통신시스템



박 규 석

중앙대학교 전자계산학과(이학석·박사)
 1990년~1991년 미 UCLA/객원교수
 1992년~1997년 경남대학교 전산정보원/원장

1995년~1996년 한국정보과학회/이사/영남지부장
 1998년~현재 행정자치부/정책자문위원(전자정부분과)
 1982년~현재 경남대학교 컴퓨터공학과/교수
 관심분야 : 운영체제, 분산처리시스템, 멀티미디어시스템, 정보통신시스템