

# 미디어 스트림 전송을 위한 적응형 멀티미디어 동기화 기법에 관한 연구

## (A Study on an Adaptive Multimedia Synchronization Scheme for Media Stream Transmission)

지 정 규\*  
(Jeong-Gyu Jee)

### 요 약

실시간 응용 프로그램은 미디어 데이터간에 만족되어야 할 동기화 제약조건을 가지고 있다. 클라이언트의 버퍼를 버퍼레벨기법의 가상버퍼와 네트워크 상태를 포함한 피드백기법을 제안한다. 가변적 전송 지연 시간을 흡수하면서 피드백 제어에 의한 동기화를 수행한다. 버퍼의 수위가 정상레벨을 유지하는가에 대한 문제는 재생률과 QoS 서비스에 중요한 영향을 주게 된다. 본 논문에서는 이러한 문제를 해결하기 위해서 필터링함수, 제어함수, 네트워크 평가 함수 등을 적용하여 멀티미디어 서버에서 전송 시작시점을 결정하고 미디어의 재생 시 끊어짐이 없는 유연한 재생을 처리하게 된다. 주 미디어인 오디오 프레임이 상위임계레벨에 버퍼의 수위가 존재 할 경우는 점차적으로 재생시간을 줄이고, 오디오 프레임이 하위임계레벨에 버퍼의 수위가 있을 경우는 점차적으로 재생시간을 늘리는 시스템을 제안한다.

### ABSTRACT

Real-time application programs have synchronization constraints which need to be met between media-data. Synchronization method represents feedback method including virtual client-side buffer. This buffer is used in buffer level method. It is client-leading synchronization that is absorbing variable transmission delay time and that is synchronizing by feedback control. It is the important factor for playback rate and QoS if the buffer level is normal or not. To solve the problems, we can control the start of transmission in multimedia server by applying filtering, control and network evaluation function.

Synchronization method is processing for smooth presentation without cut-off while media is playing out. When audio frame which is master media is in high threshold buffer level we decrease play out time gradually, otherwise we increase it slowly.

---

\* 정회원 : 한국학술진흥재단

## 1. 서론

컴퓨터의 급속한 발전으로 멀티미디어 처리 기술과 광대역 통신망 기술의 비약적 발전으로 분산 멀티미디어 응용이라는 새로운 분야가 창출되었고, 이용자들의 폭발적인 관심에 따라 현재 VOD, 원격 의료, 홈 쇼핑, 원격게임, 멀티미디어 화상회의와 같은 다양한 서비스의 확산이 급속히 진행되고 있다.

멀티미디어 시스템에서는 연속 미디어와 비연속 미디어를 포함하는 여러 형태의 정보가 통합되어 있으며, 이들 미디어 정보들간에는 시간 및 공간적 관계가 존재한다. 예를 들면, 화상 회의에서 사람의 말과 입의 움직임은 일치해야 한다. 이렇게 여러 미디어 정보들간의 관계성을 유지하는 것을 멀티미디어 동기화(multimedia synchronization)라고 한다.

본 논문은 멀티미디어 동기화와 관련하여 연구가 가장 활발하게 이루어지고 있는 멀티미디어 동기화 분야에 대한 새로운 기법을 제안하는 것이다. 동기화 기법은 현재 다양한 형태로 활발히 진행되고 있는데, 본 논문에서는 버퍼레벨기법을 제시한 Little의 버퍼제어 기법[3]과 포워드 피드백 기법을 사용한 Geyer[8]의 기법을 확장한 기법을 제안한다. 동기화 기법은 QoS를 기반으로 재생률을 최대화시켜야 하고, 버퍼 수위를 안정상태로 유지하여야 한다.

기존의 동기화 기법은 하나의 서버에 비디오와 오디오를 저장하여 전송하는 기법을 제안하였다. 그러나 이러한 방법은 네트워크의 부담을 증가시키게 되고, 여러 사용자가 동시에 같은 미디어를 수신 할 때 서버에 많은 부담을 주게 되는 문제를 가지고 있다. 따라서 본 논문에서는 여러 개의 미디어 서버에 관리 서버를 운영하여 오디오와 텍스트를 하나의 서버에 저장하고 나머지 서버에 논리적 시간으로 비디오를 나누어 저장하여 구성한다. 수신자 측에서는 버퍼의 수위를 피드백 기법의 알고리즘에만 의존하는 것이 아니라 실제 동기화 시점을 조정함으로써 버퍼의 수위를 안정상태로 유지 할 수 있도록 하는 알고리즘을 제안한다. 즉, 본 논문의 주된 목적은 통합 네트워크 상에서 분산 멀티미디어 주문형 서비스에 대해 연속적이며, 실시간 접근을 제공하기 위한 기법을 제안하고자 하는 것이다.

본 논문의 구성을 보면, 먼저 2장에서 동기화 기법과 관련된 연구와 기존 연구의 문제점을 기술한다.

3장에서는 제안된 시스템의 네트워크와 멀티미디어 서버의 형태, 그리고 동기화 시스템의 구조 설계에 대해 기술한다. 아울러 제안된 동기화 시스템의 각 유니트와 함께 재생기법과 피드백기법 등에 대해 기술한다. 그리고 4장에서는 제안 시스템의 성능평가를 위해 실험한 결과를 분석하고, 이를 기존 모델과 비교·평가한다. 마지막으로 제 5장에서는 본 연구의 결과와 향후 연구 방향에 대해 기술한다.

## 2. 동기화 기법 연구

### 2.1 송신주도형

송신측에서 데이터의 전송 시간이나 전송률을 조정함으로써 수신측에 비동기 현상이 발생하지 않도록 하거나 수신측에서 발생한 비동기 현상을 해결하는 방법이다.

ESPRIT OSI 95 프로젝트에서는 하부 네트워크가 전송률 제어에 기초한 트랜스포트 프로토콜과 자원 예약이 가능한 네트워크 계층일 것을 가정하고, 전송률을 감시하거나 조절함으로써 서비스 품질을 재협상한다. 이 방안에서는 트랜스포트 계층으로부터 네트워크의 상태를 측정하여 전송률을 제어하므로 이러한 기능을 갖는 특정한 하부 구조가 제공되어야 하는 문제를 가지고 있다[2].

Little과 Ghafoor는 송인 제어로 서비스 품질을 보장할 수 있는 하부 네트워크와 공통 클럭 환경을 가정하고, 다양한 트래픽 형태의 데이터를 재생하기 위해, 패킷의 전파 지연 시간 및 손실률 특성을 고려하여 각 미디어의 전송 시간을 스케줄 한다[3]. 이 방안에서는 공통 클럭 환경이 유지되고 네트워크의 전파 지연 및 손실률 특성을 미리 알고 있어야 한다는 문제를 가지고 있다. McManus는 VOD 서버에 저장된 비디오를 ATM망을 통해서 단말기로 전송할 때 버퍼 부족 및 범람과 수신 버퍼의 크기를 고려하여 최적의 전송 속도와 전송시작 시간을 결정하는 방안을 제안하였다[3].

## 2.2 수신 주도형

수신 주도형은 수신측에 네트워크를 거쳐 도착하는 패킷에 대해 버퍼 제어를 통하여 상위 계층으로의 플레이어아웃 시간을 조정하는 방법이다.

Escobar의 FSC(Flow Synchronization Control) 방안은 임의의 구성 형식을 가지며 분산된 환경의 하부 통신망에서 동기화 된 네트워크 클럭을 제공할 것을 가정하고 애플리케이션에서 다중 미디어 그룹에 속하는 각 흐름간에 공통 지연 시간을 갖는다. 이 시스템의 장점으로는 그룹 통신에 적합하고, 그룹에 속한 모든 노드는 절대 시간을 갖는다[9]. 단점으로는 하나의 그룹은 지역적으로 모여 있어야 한다. 정기적인 메시지에 포함되는 정보를 쉽게 얻기 힘들기 때문에 물리적인 클럭이 다른 시스템과 함께 적용하기 힘들다.

Rothermel은 버퍼 수위에 의해 네트워크의 지연 변화를 관찰하여, 버퍼의 상한치와 하한치를 벗어날 경우에 흐름제어를 조절함으로써 버퍼의 크기를 일정하게 유지한다[10]. 대부분의 방안들은 버퍼 수위를 관찰하고 조정하거나, 고정 버퍼의 출력률을 제어하는 방법을 사용하고 있기 때문에 패킷의 크기와 전송률을 알고 있어야 하며, 패킷의 크기나 전송률이 가변적인 가변 전송률(VBR: Variable Bit Rate) 데이터를 처리하기에는 부적합하다.

## 2.3 혼합형

수신측에서는 전달된 패킷의 도착 시간으로부터 네트워크의 지연 특성을 추정하여 송신측으로 피드백하여 전송 스케줄에 반영하거나[7][8], 송신측은 수신측에서 활용할 수 있도록 필요한 트래픽 정보를 추가로 전달해 준다. 혼합 주도형은 피드백과 피드포워드로 분류한다. 두 방식의 구분은 네트워크와 시스템의 변화되는 환경에 동적으로 적용하면서, 송신측과 수신측 중 어느 쪽이 흐름을 조정하는가에 따라서 결정된다. 송신측은 네트워크의 환경에 쉽게 접근할 수 없는 반면, 수신측은 종단간 연결의 상태에 대한 자세한 정보를 갖는다. 이를 고려할 때, 송신측(서버) 혹은 네트워크로 인한 모든 스트림 폭주의 영향 등이 수신측에 의해 일단 감지되고, 구분된 후에 수신측 혹은 송신측에 의해 흐름제어가 이루어

져야 한다.

Gayer 기법은 송신측에서 스트림 미디어를 전달하기 전에 평가단계에서 전송시작시점을 라운드트립(roundtrip)의 패킷을 전송하여 전송시작시점을 정의하게 된다[8]. 이러한 방법은 지터(jitter)를 정의하여 미디어 내의 지터를 없애려고 하였다. 또한 최대의 지터를 발생한 서버의 번호와 다른 서버의 최초의 전송시점을 조정함으로써 버퍼의 크기와 미디어 간의 스큐 문제를 해결하였다. 그리고 버퍼레벨은 버퍼의 임계레벨 및 제어레벨에서 필터링함수를 이용하여 읍셋을 구하고 읍셋만큼 서버에 피드백하여 전송시작시간을 늦추거나 앞당겨 버퍼를 안정상태로 유지하는 기법을 사용하게 된다. 이 기법은 버퍼의 안정상태를 유지하기 위한 방법을 제시하여 분산환경에서 재생률을 높일 수 있었다.

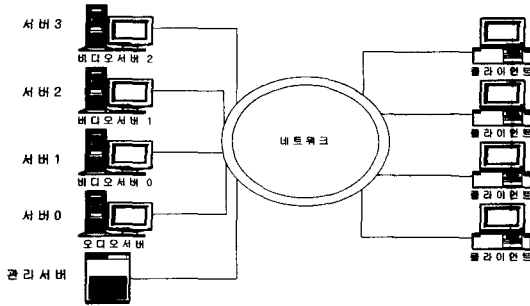
그러나 버퍼의 안정상태를 유지하기 위해서는 얻어지는 필터링함수의 값이 상황에 따라서 변하는 네트워크의 상태를 적용하지 못하므로 기아상태나 오버플로우 상태에서 버퍼의 안정상태로 옮겨가는데 많은 시간을 필요로 하였다. 또한 클라이언트에 재생시점을 고려하지 않았기 때문에 안정상태를 유지하기에는 부족하게 된다.

## 3. 멀티미디어 동기화 시스템

### 3.1 제안 시스템의 구성

멀티미디어 서버의 구조는 하나의 시스템에 여러 개의 영화를 저장하고 전송하게 된다. 이러한 구조에서는 영화의 액세스 시간과 서버의 준비 시간에 많은 부담을 갖게 되고 네트워크 트래픽 증가에 많은 영향을 받으며 성능의 저하를 가져오는 결과를 갖게 된다. 본 논문에서는 이러한 문제점을 해결하기 위해서 하나의 영화를 네 개의 서버로 나누어 저장함으로써 서버의 전송준비시간, 네트워크 트래픽, 버퍼의 크기 감소 등의 장점을 가질 수 있다. [그림 1]과 같이 서로 다른 네트워크에 멀티미디어 서버를 갖는 구조를 구성하게 된다. 멀티미디어 서버에서 보내주는 서브 스트림은 하나의 동기화 그룹을 형성하게 된다. 또한 주 미디어인 오디오 미디어를 하나

의 서버로 형성시킴으로써 하나의 오디오에 세 개의 비디오 프레임 을 가지고 동기화 그룹으로 형성하게 하는 시스템을 제안한다. 주 미디어가 다른 서버로 구성됨으로서 네트워크와 서버의 전송 시작 준비 시간을 절약하게 하는 장점을 갖게 된다.



[그림 1] 멀티미디어 서버 네트워크  
[Fig. 1] Multimedia Server Network

서버의 데이터 형태는 [그림 2]의 구조를 가지고 있다. 즉 비디오의 데이터 형태는 논리적 시간을 가지고 스트림을 분산하여 저장한다. 이것은 주 미디어인 오디오를 기준으로 분산하게 되며, 구성은 하나의 동기화 그룹을 기준으로 하지만 같은 바이트를 기준으로 하지는 않는다. 왜냐하면 압축된 데이터의 논리적 시간을 기준으로 하기 때문에 주 미디어의 논리적 시간이 기준이 되어 각 미디어를 서브 스트림으로 분산하는 구조를 보이게 된다.

	동기화 그룹 $M_1$	동기화 그룹 $M_{j+1}$
비디오서버 2	비디오 프레임 $M_{1,3}$	비디오 프레임 $M_{j+1,3}$
비디오서버 1	비디오 프레임 $M_{1,2}$	비디오 프레임 $M_{j+1,2}$
비디오서버 0	비디오 프레임 $M_{1,1}$	비디오 프레임 $M_{j+1,1}$
오디오서버	오디오 프레임 $M_{1,0}$	오디오 프레임 $M_{j+1,0}$

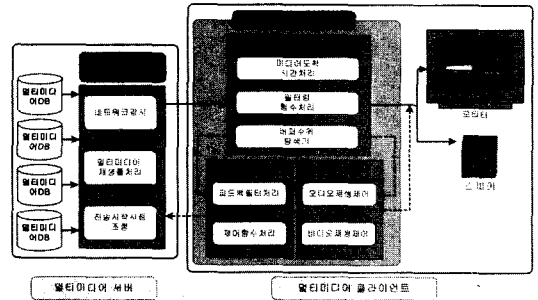
시간

[그림 2] 미디어 데이터의 구조  
[Fig. 2] Structure of Media Data

네트워크의 구조를 기반으로 한 클라이언트의 구조는 [그림 3]과 같다. 버퍼의 크기는 Rangan에 의해서 정의된 지터를 가지고 버퍼의 크기를 결정하였다[5]. 본 논문에서는 서버가 전송시작시간을 조정하

여 네트워크 지터의 시간을 보상하는 방법을 채택한다.

클라이언트는 네트워크의 상태, 즉 각 서버에서 도착되는 서브 스트림의 도착시간, 서브 스트림의 버퍼 대기시간, 그리고 버퍼의 상태에 관한 정보를 관리하는 수신처리 관리자의 버퍼수위관리 유니트가 있다.



[그림 3] 클라이언트 시스템  
[Fig. 3] Client System

버퍼수위관리 유니트에서 발생하는 정보를 분석하여 각각의 서버에 피드백하는 QoS 피드백 유니트와 버퍼의 상태를 분석하여 재생시간을 조정하는 플레이아웃 유니트가 있다. 클라이언트 시스템에서는 멀티미디어 서버의 전송 시작시간을 조정하여 미디어 내 동기화 하는 구조를 가지고 또한 동기화 그룹으로 미디어간의 동기화를 하게된다.

### 3.2 버퍼 수위관리 유니트

서버에서 전송되는 서브 스트림이 네트워크를 통과하는 동안 동기화가 파괴되는 원인은 여러 가지가 있게 된다. 이러한 대표적인 원인은 네트워크 지연이다. 동기화 구조에서 적절한 시간 순서로 연속적인 미디어 스트림을 재생하기 위한 네트워크 지연을 보상해야 한다. 또한 네트워크의 지연 외에도 미디어 단위는 패킷크기나 패킷화를 위하여 통신프로토콜의 물리적 계층에서 일어나는 지연이나 클라이언트에서의 저장 등의 지연을 갖게 된다. 이러한 지연의 변화를 지터라고 한다.

이러한 지터는 동기화 구조에서 해결 할 문제로 본 논문에서는 네트워크의 지연의 정보를 각 서버에

서 전송되어 도착되는 미디어의 도착시간과 미디어의 버퍼 대기시간 그리고 버퍼 점유율 등의 요소를 가지고 함수를 발생하게 한다.

서브 스트림이 도착하게 되면 미디어 번호, 도착시 클라이언트의 로컬시간을 기록하고 동기화 그룹의 번호를 체크한다. 서브 스트림이 도착하였으므로 같은 서버의 이전 서브 스트림의 도착시간을 체크하여 네트워크의 상태가 평균 도착시간 보다 빨라지고 있는가, 늦어지고 있는가를 체크하게 된다. 이러한 요소의 상태는 버퍼수위관리 유닛의 저장공간에 저장되어 QoS 피드백 유닛과 플레이아웃 유닛이 공유하여 네트워크의 상황 및 각 서버의 상황을 체크하게 된다.

[그림 4]와 같이 버퍼 상태는 정상레벨, 상위임계레벨, 상위제어레벨, 하위임계레벨, 하위제어레벨의 5가지의 상태를 갖게 된다. 이러한 상태와 네트워크의 관계는 다음과 같다.

① 정상 레벨

처음에 미디어 데이터가 도착해서 미디어 데이터 버퍼가 정상적인 레벨에 있는 경우로 버퍼수위관리 유닛은 버퍼가 별 이상이 없으므로 계속 미디어 데이터를 전송 받도록 한다. 버퍼수위관리 유닛은 각 서버의 지터범위를 체크하고 정보를 저장한다.

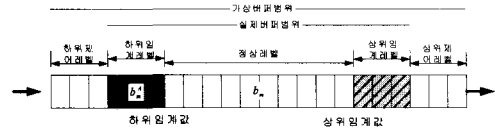
② 상위임계 · 제어레벨

네트워크가 평균 지연시간 보다 빨리 도착되어 네트워크의 상황이 좋아지는 경우이다. 버퍼가 상위임계레벨에 이르게 되면 버퍼가 오버플로우를 발생할 가능성이 높게 된다. 버퍼의 레벨이 상위제어레벨에 이르게 되면 버퍼는 오버플로우가 발생된 경우이다. 버퍼수위관리 유닛의 버퍼수위탐색기는 상위임계레벨에 버퍼의 수위가 옮겨가면 네트워크의 상태와 버퍼의 상태를 QoS 피드백 유닛과 플레이아웃 유닛에게 통보하게 된다.

③ 하위임계 · 제어레벨

네트워크가 평균 지연시간 보다 늦게 도착되어 네트워크의 상황이 나빠지는 경우이다. 버퍼관리 유닛은 버퍼가 하위임계레벨에 이르게 되면 버퍼가 언더플로우를 발생할 가능성이 높게 된다. 버퍼의 레벨이 하위제어레벨에 이르게 되면 버퍼가 기아현상

이 발생된 경우이다. 버퍼수위 유닛의 버퍼수위 탐색기는 상위임계레벨에 버퍼의 수위가 옮겨가면 네트워크의 상태와 버퍼의 상태를 QoS 피드백 유닛과 플레이아웃 유닛에게 통보하게 된다.



[그림 4] 버퍼상태 수위  
[Fig. 4] Buffer State Level

3.3 QoS 피드백 유닛

버퍼수위관리 유닛이 비동기, 버퍼레벨의 변화를 발견한다면, 그것은 동기화를 회복하기 위한 측정의 기준이 된다. 본 논문에서는 스무딩 버퍼레벨을 측정하는 함수를 이용하여 버퍼의 수위와 버퍼의 레벨을 측정한다. 버퍼의 수위를 이용하는 방법은 동기화 기법에서 많이 사용되는 기법이다[3]. 측정된 버퍼의 수위는 제어 함수에 의해서 제어되어야 할 버퍼수위를 구하여 서버에게 통보하게 된다.

필터링함수  $S(q_{t,m})$ 는 시간에 따라 변하는 버퍼를 측정하는 방법을 제시한다. 시간 t에서 서브 스트림 m을 위한 버퍼 레벨은  $q_{t,m}$ 에 의해 나타난다.  $q_{t,m}$ 의 값은 네트워크의 짧게 변화되는 지터에 의해서 버퍼의 수위가 연속적으로 변화되는 것을 예방하기 위한 부드러운 버퍼 레벨  $b_{t,m}^-$ 를 계산하도록 필터링 함수로 전달된다. 예를 들면 필터링 함수는 기하학적 가중치를 지닌 부드러운 함수이다. 즉, 버퍼의 수위함수가 로컬 피크를 예방하는 방법으로 사용되어진다(부드러운 요소를 갖는 a)[1][6][8].

$$b_{t,m}^- = S(q_{t,m}) = a \cdot b_{t-1,m}^- + (1-a) \cdot q_{t,m} \dots \text{(식 1)}$$

(with  $a \in [0,1]$ )

필터링의 주요 목적은 지터와 긴 간격에 의해 생기는 변화를 버퍼 레벨에서 구분하는 것이다. 만약 필터가 너무 민감하거나 필터가 전혀 사용되지 않는

다면 지터는 예외적인 상황이 발생되지 않을지라도 재동기화를 시도하게 할 것이다. 다른 면에서 필터는 버퍼 오버플로우나 언더플로우의 긴 주기에 대한 결과를 가지고 너무 늦게 행동을 취하는 조건을 바꾸도록 천천히 반응한다. 그러므로 재생의 질은 좋아지게 된다.

비동기 문제의 잠재적 버퍼수위 반응은  $S(q_{t,m})$ 의 수행에 있어 강하게 의존된다. 좀더 느리게  $S(q_{t,m})$  반응되면 마지막 재 동기화 단계는 더 많은 버퍼 공간인 상·하한임계레벨 버퍼 슬롯의  $b_m^A$ 가 가능한 많이 비동기화를 위해서 보상하게 될 것이다. 다른 면에서  $S(q_{t,m})$ 가 민감하게 반응하게 되고, 종종 재 동기화가 불필요하게 행한다면 적은 버퍼 공간인  $b_m^A$ 가 충분하게 제공되어야 한다.

본 논문에서는  $\alpha$ 를 0.6 이나 0.7의 값이 버퍼 요구와 재동기화에 반영되어 버퍼의 스무딩 버퍼레벨을 구하게 된다[7][8]. 이 수치는 강하게 반응하지 않으면서 민감하게 재동기화의 수준을 변형하게 한다.

### 3.4 플레이아웃 제어

본 논문은 프레임의 디스플레이 시간이 주기적인 상태를 엄격하게 유지하지 않는 것을 제안한다. 만약 버퍼의 레벨이 정상레벨보다 상위 제어 레벨이 높다면 디스플레이 시간(duration)은 더 길어진다. 그리고 더 많은 프레임을 위하여 충분한 시간을 가지는 버퍼대기시간을 허용한다. 또한 버퍼의 레벨이 정상레벨보다 적다면 디스플레이 시간은 짧아진다.

이 기법은 하나의 동기화 시간에 이루어지며 주 미디어가 도착된 시간의 주 미디어의 크기를 기준으로 하여 종 미디어의 재생시간을 가변시간으로 재생되는 기법이다.

주(master) 미디어란 동기화의 기준이 되는 미디어를 주 미디어라 하고 주 미디어에 동기화 되는 미디어를 종(slave) 미디어라 한다. 주 미디어인 오디오를 재생 버퍼 레벨 기법에 적용하고, 이 절에서는 재생 버퍼 레벨 기법에 대하여 설명한다.

<표 1>은 재생정책에서 이용되는 매개변수에 대한 설명이다.

각 패킷의 동기화 구간 크기를  $\Delta_S$ 라 할 때,  $i$  번째 패킷이  $T_i$ 라는 시간에 출력되었으면  $i+1$ 번째 패킷은  $T_i + \Delta_S$  시간에 출력되어야 미디어간 동기화가 이루어질 것이다. 그러나 연속 미디어 데이터는 그 종류에 따라 어느 정도의 불연속을 허용한다. 예를 들어, 음성 미디어인 경우 종단간 최대 지터 지연이 10ms 이하이면 서비스 품질에 영향을 주지 않고 일시적인 불연속을 허용할 수 있다. 또한 비디오 데이터인 경우에는 지정된 동기화 구간에 어느 정도 분량의 패킷을 출력하지 못해도 비디오의 서비스 품질에는 큰 영향을 주지 않는다.

<표 1> 재생정책에 사용되는 매개변수  
<Table 1> Argument in Playback Policy

매개변수	설 명
$\rho_i$	한 서브스트림의 재생시간(ms)
$\gamma$	미디어 재생률
$\lambda$	미디어내 최대 지터시간(ms)
$BL_{allsize}$	버퍼레벨에서의 전체크기
$BL_{psize}$	현재 버퍼 수위
$\theta$	가중치
$\gamma_i$	한 서브스트림의 확장된 재생시간(ms)
$\tau_i$	버퍼레벨에 의한 오디오의 재생시간(ms)
buffer_pointi	1가 0이면 하위제어레벨 1가 10이면 하위임계레벨 1가 20이면 정상레벨 1가 30이면 상위임계레벨 1가 40이면 상위제어레벨
VLTU(i)	비디오프레임 한 개의 논리적 재생시간(ms)
$p_n$	하나의 동기화그룹에 재생되어야 할 종미디어 수
$AUDIO_i$	하나의 동기화그룹에서 재생되는 오디오 재생시간(ms)

$$\rho_i = 1 / r \dots\dots\dots (식 2)$$

$$\gamma_i = \rho_i + \lambda \dots\dots\dots (식 3)$$

(식 2)는 한 개의 서브 스트림의 재생시간을 구한 식이다. 이것은 미디어 시간을 재생틀로 나누어 한 개의 서브 스트림의 재생 시간을 얻을 수 있다. 또한 (식 3)은 최대 지터 시간을 정상적인 재생시간에 더하여 확장시킨 시간이다.

버퍼가 정상레벨을 유지할 경우에는 미디어의 재생처리 구간을 정확하게 지켜나가게 된다. 만약 1초에 24프레임을 요구하는 시스템일 경우는 41.6ms의 시간으로 재생을 하게 된다. 그러나 예상 못했던 네트워크의 지터에 의해서 미디어의 도착시간이 늦어진다면 버퍼에 미디어 프레임이 많이 확보하고자 할 것이다. 이러한 문제점을 해결하기 위해서는 두 가지 방법이 있다. 첫 번째 방법은 네트워크에서의 흐름제어 및 데이터의 양을 변환시키므로 문제를 해결하는 방법이다. 이 방법에서는 피드백 전략에서 이미 설명하였다. 두 번째 방법으로는 버퍼의 점유율을 가지고 재생 시점을 변화시키는 방법이다. 이러한 방법은 최대지연지터인 10ms 이하에서 동기화 시점을 조정할 수 있다는 아이디어를 본 논문에 적용하게 되었다.

버퍼의 점유율이 상위임계레벨에 존재할 때, 가중치에 의해서 재생시간을 변경하는 알고리즘이다.

```

PROCEDURE Lower_level
BEGIN
     $BL_{allsize}$ ;
     $BL_{psize}$ 
     $\omega = 0$ ;
     $\rho_i = 125$ 
     $\lambda = 10$ 
    IF buffer_point = 1 THEN
    BEGIN
         $\omega' = BL_{allsize} - BL_{psize}$  ; ....(식 4)
         $\omega = \omega' / BL_{allsize}$  ; ..... ..(식 5)
         $\tau_t = \rho_i + \lambda \times \omega$ 
    END
    END IF
END
    
```

(식 4)에서  $BL_{allsize}$ 은 상위임계레벨에 속하는 버퍼의 개수를 갖게되고  $BL_{psize}$ 는 상위임계레벨에 차지하고 있는 현재의 개수를 갖게 된다. (식 5)에서는 가중치를 최대지연지터와 곱함으로써 재생시간을 늘려주게 되며, 동기화 시점을 변경하는 결과를 가져오는 효과를 갖게 된다. 이것은 기아 상태를 사전에 방지하는 효과를 갖게 되며, 이러한 효과는 상위제어레벨로 버퍼수위를 떨어지는 것을 늦출 수 있는 효과를 갖게 된다.

버퍼의 수위가 하위임계레벨에 존재한다면 (식 6)을 적용하여 재생시간에 대한 가중치를 변경하게 된다.

```

PROCEDURE Upper_level
BEGIN
     $BL_{allsize} = BL_{allsize} + 1$ ;
     $BL_{psize}$ 
     $\omega = 0$  ;
     $\rho_i = 125$  ;
    IF buffer_point = 3 THEN
    BEGIN
         $\omega = BL_{allsize} / BL_{psize}$  ; .... (식 6)
         $\tau_t = \rho_i - (\rho_i \times \omega)$  ; ..... (식 7)
    END
    END IF
END
    
```

(식 7)에서  $\rho_i$ 는 예상되어졌던 재생시간이다. 즉, 1초에 24프레임을 요구하는 시스템일 경우 41.6ms가 된다.

또한, 실제로 계산되어진 재생시간이 된다. 이러한 기술을 멀티미디어 통신에 적용함으로써 예상하지 않은 지터에 의한 버퍼의 오버플로우에 의해 도착되는 미디어를 버리는 경우를 어느 정도 줄일 수 있다.

#### 4. 실험 및 성능평가

##### 4.1 실험 환경

본 논문의 실험을 위한 환경으로는 IBM 호환 기종의 펜티엄 PC를 이용하였으며, 인터페이스 및 알고리즘은 Java 개발 킷 JDK 1.3으로 구현하였고, 마이크로 소프트 MDB에 simulation.mdb 화일로 저장된다.

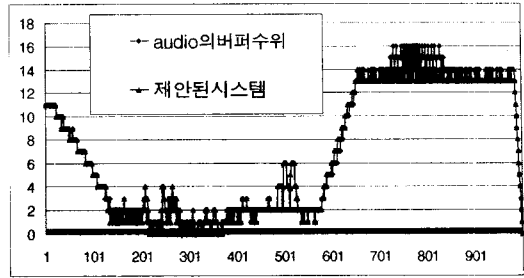
1K byte 오디오 데이터는 PCM 인코딩 기법에 의해서 인코딩 되고 비디오 프레임의 해상도는 120 X 120을 사용했다. 초당 24프레임의 인코딩 작업을 하여 사용되어진 프레임이 된다. 비디오 플레이스의 수는 하나의 오디오 플레이스와 비교되는데 이유는 다음과 같다. 클라이언트에서 어플리케이션은 125ms 마다 오디오 장치로부터 오디오 패킷을 얻고 125ms 동안에 어플리케이션은 운영체제의 런타임 프로세싱 오버헤드에 의해서 결정된 세 개 이하의 비디오 프레임을 비디오 그래픽으로부터 얻는다.

본 논문은 실험 환경을 인터넷상의 WAN 환경으로 가정하였다. 각각의 패킷에 대한 적절한 작업을 수행하기 위해서 실제 시뮬레이션에 사용된 정보는 포아송 분포로 산출하여 네트워크 지연 시간을 두 가지 미디어에 똑같이 적용하였다.

성능 측정 실험에서 사용되는 프레임의 개수는 1000개이고, 실험에서 최대 지연 지터 시간은 100ms를 적용하였다.

##### 4.2 실험 결과 및 분석

본 절에서는 버퍼레벨을 정상상태로 유지하기 위한 정책의 네트워크 정책과 버퍼레벨을 이용한 재생 시간을 이용한 재생시간과 손실 시간을 기존의 기법과 비교·분석한다. 먼저 본 논문에서 버퍼레벨을 제어하여 버퍼의 상태가 기아 상태나 오버플로우 상태를 보이고, 버퍼의 수위의 변화를 비교하여 타 방법과의 비교를 통해 본 논문에서 제안한 방법의 우수성을 보인다.

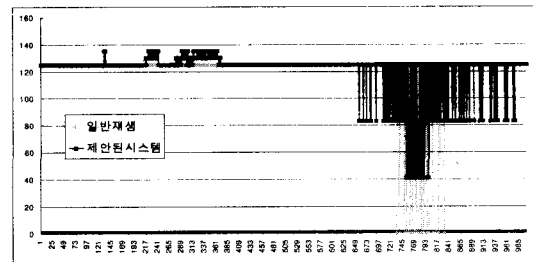


[그림 5] 버퍼수위 변화

[Fig. 5] Variation of Buffer Level

[그림 5]는 버퍼의 수위가 변화되는 그래프이다. 200번부터 300번의 프레임 도착시간이 느려짐으로서 버퍼에서는 정상상태를 벗어나는 경우를 보이고 있다. QoS 피드백 유니트에 의하여 피드백 정책에 의해서 버퍼의 수위가 안정상태로 변하는 그림을 보이고 있다. 또한 일반 정책보다 늦게 기아상태에 도착됨을 보이고 있다.

[그림 6]은 제안된 시스템의 플레이아웃 유니트 프로세서가 버퍼수위에 의한 재생정책을 한 결과를 보이고 있다. 프레임 번호 200에서 300사이에서는 무변화의 경우를 보이고 있고, 720부터 820까지는 통과되는 경우를 보이고 있다.



[그림 6] 재생률 변화

[Fig. 6] Variation of Playback Rate

[그림 7]은 실험의 결과를 나타내는데, 도착시간을 변경하면서 31번의 실험에 의한 결과이다.



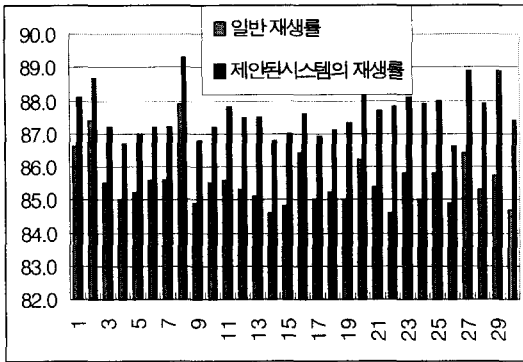


그림 7 재생률 비교

[Fig. 7] Comparison of Playback Rate

실험번호 2, 8에서는 최소지연을 50ms에서 최대 지연을 600ms로 조정함으로써 재생률이 일반 기법보다 제안된 시스템의 경우 약 10%를 향상시키는 결과를 얻을 수 있었다. 또한 4, 9, 26번째의 실험에서는 최소지연을 20ms에서 최대지연을 800ms로 조정함으로써 오버플로우와 기아현상을 발생시킨 결과를 보여주고 있다. 이 경우에는 네트워크 상태의 트랙픽을 나쁘게 만들면서도 제안된 시스템의 경우 약 8-9%를 향상시킬 수 있는 결과를 얻을 수 있었다. 평균적으로 일반기법의 경우 약 79%를 나타내고, 제안된 시스템의 경우는 91%의 재생률을 얻을 수 있었다.

### 5. 결론

본 논문은 멀티미디어 시스템 및 서비스 제공에 있어 핵심적인 기술로 부각되는 동기화에 대한 동기화 기법을 제시하였다. 단일 서버를 이용한 멀티미디어 전송에서 사용된 버퍼의 크기를 분산 환경으로 확장함으로써 버퍼의 크기를 줄일 수 있었다. 또한 오디오와 텍스트, 비디오 등의 데이터를 여러 서버에 저장함으로써 인기가 높은 미디어를 적절히 서비스 할 수 있었다. 또한 버퍼레벨을 사용한 기존의 기법에 재생제어 기법을 포함한 버퍼레벨 제어를 함으로써 버퍼의 기아 상태나 오버플로우 상태를 막을 수 있었다.

그리고 서버에서의 미디어 동기화 방법과 수신측 동기화 방법을 사용하여 혼합형 동기화 기법을 이용하여 재생률을 높일 수 있었다.

본 논문에서는 수신측에서 버퍼레벨을 이용함으로써 미디어 내 동기화 및 미디어 간 동기화를 제안하였다. 주 미디어인 오디오 미디어의 최대 지연 지터 시간을 이용하여 미디어 내의 동기화 구간을 확장하였고, 그 시간을 미디어간 동기화에 적용함으로써 확장된 구간 내에서 불규칙적으로 도착하는 패킷들을 서로 동기화 하여 서비스 품질을 저하시키지 않고서도 최적의 동기화를 수행할 수 있었다.

본 논문에서 제안한 프레임에 대한 조절된 디스플레이 재생시간 기법은 부드럽고 자연스러운 성능을 나타내었고, 전체 재생 시간은 원래의 재생시간을 벗어나지 않았다.

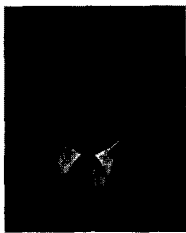
향후 연구 방향은 사용자와의 상호 작용을 고려한 정형화된 멀티미디어 모델을 만들고 이를 트랜스포트 프로토콜로 구현하여 시뮬레이션 하는 것이다. 또한 동영상상을 이동통신에서 사용되게 하기 위한 최적의 동기화 기법을 연구해야 할 것이다.

### ※ 참고 문헌

- [1] R. Steinmetz and K. Nahrstedt, *Multimedia: Computing, Communication and Applications*, Prentice Hall, 1995.
- [2] M. J. Perez-Luque and T. D. C. Little, "A Temporal Reference Framework for Multimedia Synchronization," *IEEE Journal on selected Areas in Communications*, Vol.14, No.1, Jan. 1996.
- [3] T. D. C. Little, and Arif Ghafoor, "Multimedia Synchronization Protocols for Broadband Integrated Services," *IEEE Journal on selected Areas in Communications*, Vol. 9, No.9, Dec. 1991.
- [4] N. U. Qazi, M. Woo, and A. Grafoor, " A Synchronization and communication model for distributed multimedia objects," in *Proc. ACM Multimedia*, 1993.

- [5] S. Ramanathan and P. V. Rangan, "Feedback Techniques for Intra-Media Continuity and Inter-Media Synchronization in Distributed Multimedia Systems," The Computer Journal, Vol. 36, No.1, Oct. 1993.
- [6] R. Steinmetz, "Human Perception of Jitter and Media Synchronization," IEEE Journal on selected Areas in Communications, Vol.14, No.1, Jan. 1996.
- [7] W. Geyer, "Stream synchronization in a scalable video server array," Master's thesis, Institut Eurecom, Sophia Antipolis, France, Sep. 1995.
- [8] Ernst Biersack, Werner Geyer, "Synchronization delivery and playout of distributed stored multimedia streams", Multimedia Systems ,Vol. 7, No. 1 , pp. 70-90, 1999.
- [9] J. Escobar, C. Patridge, and D. Deutsch, "Flow Synchronization Protocol," IEEE/ACM Transaction on Networking, Vol 2, No.2, pp. 111-121, Apr. 1994.
- [10] 이기성, 오해석, "멀티미디어 동기화 모델에서 유연한 재생 처리 기법", 한국통신학회논문지, 제 25권, 제 12A호, 2000.

지 정 규



1987년 서울산업대학교 전자계산학과(공학사)  
1989년 숭실대학교 전자계산학과(공학석사)  
1998년 숭실대학교 전자계산학과(공학박사)  
1978년~1996년 (주)삼호, 서울 시설관리공단 전산실  
1996년~현재 한국학술진흥재단 학술정보팀  
관심분야 : 멀티미디어, 데이터베이스, 영상처리