

멀티미디어 통신에서 미디어스트림 전송을 위한 적응형 멀티미디어 동기화 기법

이 기 성[†]

요 약

실시간 응용 프로그램은 미디어 데이터간에 만족되어야 할 동기화 제약조건(synchronization constraints)을 가지고 있다. 멀티미디어 데이터의 가변적 전송 지연 시간을 흡수하여 피드백 제어와 재생정책에 의한 동기화 기법을 수행한다. 버퍼의 수위가 정상레벨을 유지하는가에 대한 문제는 재생률과 QoS 서비스에 중요한 영향을 주게 된다. 본 논문에서는 버퍼의 수위를 안정상태로 유지하기 위해 피드백을 위한 필터링합수를 적용하고, 재생시간을 적응적으로 적용하여 미디어의 재생 시 끊어짐이 없는 유연한 재생을 한다. 또한 버퍼의 주 미디어인 오디오 프레임의 버퍼가 상위임계레벨에 수위가 존재 할 경우는 적응적으로 재생시간을 줄이고, 하위임계레벨에 버퍼의 수위가 있을 경우는 점차적으로 재생시간을 늘리는 시스템이다.

An Adaptive Multimedia Synchronization Scheme for Media Stream Delivery in Multimedia Communication

Gi Sung Lee[†]

ABSTRACT

Real-time application programs have constraints which need to be met between media-data. It is client-leading synchronization that is absorbing variable transmission delay time and that is synchronizing by feedback control and palyout control. It is the important factor for playback rate and QoS if the buffer level is normal or not. This paper, The method of maintenance buffer normal state transmits in multimedia server by appling feedback of filtering function. And synchronization method is processing adaptive playout time for smooth presentation without cut-off while media frame is skip. When audio frame which is master media is in upper threshold buffer level we decrease play out time gradually, low threshold buffer level increase it slowly.

키워드 : 멀티미디어 통신(multimedia communication), 멀티미디어 동기화(multimedia synchronization), 미디어 스트림(media stream), 지터(jitter), 분산 멀티미디어 동기화(distributed multimedia synchronization)

1. 서 론

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[1-3].

멀티미디어 시스템에서 이용되는 비디오나 오디오 정보들은 기존의 텍스트나 그래픽과 같은 정보와는 다른 특성을 가지고 있다. 비디오나 오디오등의 정보는 일정한 시간 간격으로 연속적으로 데이터가 발생되고, 수신측에서는 일정한

시간에 프레임을 연속적으로 재생시켜야 한다. 그러나 연속 미디어를 읽어서 전송하는 서버의 역량이 충분하다고 할지라도 동시에 여러 연속 미디어를 전송하는 경우에는 제한된 통신망의 대역폭으로 인한 병목현상으로 인한 가변적인 트래픽의 문제가 발생할 수 있다. 이러한 문제를 해결하기 위해서는 버퍼의 수위를 정상레벨에 유지시키게 하는 것은 재생률과 QoS 서비스에 중요한 영향을 주게 된다.

기존의 Little와 Gagoor 연구에서는 하나의 서버에 비디오와 오디오를 저장하여 전송하는 기법을 제안하였다[4]. 그러나 이러한 방법은 네트워크의 부담을 증가시키게 되고, 여러 사용자가 동시에 같은 미디어를 수신할때 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. 또한 Geyer

※ 본 논문은 호원대학교 교내학술연구조성지비에 의한 결과임.

† 정 회 원 : 호원대학교 컴퓨터학부 교수

논문접수 : 2002년 7월 4일, 심사완료 : 2002년 11월 22일

멀티미디어 동기화 기법은 버퍼수위를 사용하여 피드백을 이용한 서버의 전송 시작시점을 조정하는 알고리즘을 제안하였다[8, 9]. 그러나 Geyer와 Little의 알고리즘은 재생정책을 적용하지 못하여 멀티미디어 데이터의 특성인 연속적인 시간관계가 네트워크 상태에 의존하게 되는 문제점을 갖게 된다. 본 논문에서는 수신자 측에서는 버퍼의 수위를 피드백 기법의 알고리즘에만 의존하는 것이 아니라 실제 동기화 시점 및 재생시간을 조정함으로써 버퍼의 수위를 안정상태로 유지할 수 있도록 하는 알고리즘을 제안한다. 본 논문의 구성은 2장에서는 동기화 기법과 관련된 연구와 기존 연구의 문제점을 제시하게 된다. 3장에서는 제안된 시스템의 네트워크와 멀티미디어 서버의 형태를 설명하고, 제안된 동기화 시스템의 구조를 설계하게 된다. 또한 분산 멀티미디어 서버를 위한 수신측의 버퍼크기를 정의하게 된다. 재생기법과 피드백기법 등을 설명한다. 4장에서는 서비스 품질을 비교하기 위해 시뮬레이션을 통하여 기존 모델보다 우수함을 보이고 비교·평가한다. 5장에서는 본 연구의 결과를 바탕으로 한 결론과 지속적인 연구를 위한 향후 연구 방향에 대하여 기술한다.

2. 기존 멀티미디어 동기화 기법

2.1 송/수신 주도형

송신측에서 데이터의 전송시간이나 전송률을 조정함으로써 수신측에 비동기(asynchronization) 현상이 발생하지 않도록 하거나 수신측에서 발생한 비동기 현상을 해결하는 방법이다. Little과 Gafoor는 승인(admission) 제어로 서비스 품질을 보장할 수 있는 하부 네트워크와 공통 클럭(common clock) 환경을 가정하고, 다양한 트래픽 형태의 데이터를 재생하기 위해, 패킷의 전파 지연시간 및 손실률 특성을 고려하여 각 미디어의 전송시간을 스케줄 한다[2]. 이 방안에서는 공통 클럭 환경이 유지되고 네트워크의 전파 지연 및 손실률 특성을 미리 알고 있어야 한다는 문제를 가지고 있다. McManus는 VOD(Video-On-Demand) 서버에 저장된 비디오를 ATM망을 통해서 단말기로 전송할때 버퍼 부족 및 범람과 수신 버퍼의 크기를 고려하여 최적의 전송 속도(transmission speed)와 전송시작 시간(start-up)을 결정하는 방안을 제안하였다[5].

Rothermel은 버퍼 수위에 의해 네트워크의 지연 변화를 관찰하여, 버퍼의 상한치와 하한치를 벗어날 경우에 흐름제어를 조절함으로써 버퍼의 크기를 일정하게 유지한다[8].

2.2 혼합형

수신측에서는 전달된 패킷의 도착 시간으로부터 네트워크의 지연 특성을 추정하여 송신측으로 피드백하여 전송 스케줄에 반영하거나[5], 송신측은 수신측에서 활용할 수 있

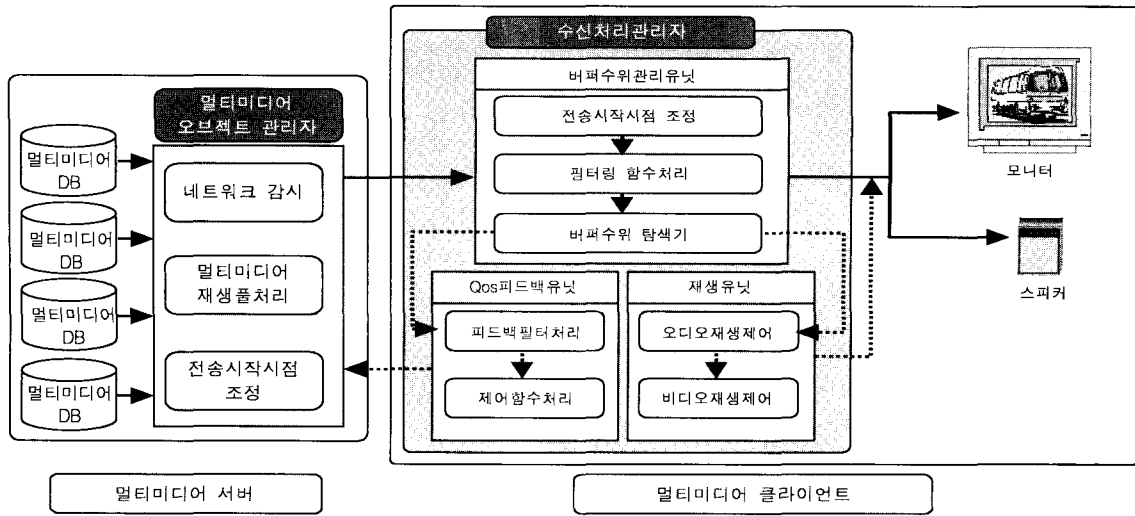
도록 필요한 트래픽 정보를 추가로 전달해 준다. 혼합 주도형은 피드백(feedback)과 피드포워드(feed forward)로 분류한다. 두 방식의 구분은 네트워크와 시스템의 변화되는 환경에 동적으로 적응하면서, 송신측과 수신측 중 어느쪽이 흐름을 조정하는가에 따라서 결정된다. 송신측은 네트워크의 환경에 쉽게 접근할 수 없는 반면, 수신측은 종단간 연결의 상태에 대한 자세한 정보를 갖는다. 혼합형은 송신측 혹은 네트워크로 인한 모든 스트림 폭주의 영향 등이 수신측에 의해 일단 감지되고, 구분된 후에 수신측 혹은 송신측에 의해 흐름제어가 이루어져야 한다[5].

Geyer 기법은 송신측에서 스트림미디어를 전달하기 전에 평가단계에서 전송시작시점을 라운드트립(roundtrip)의 패킷을 전송하여 전송시작시점을 정의하게 된다[12, 13]. 이러한 방법은 지터(jitter)를 정의하여 미디어 내의 지터를 없애려고 하였다. 또한 최대의 지터를 발생한 서버의 번호와 다른 서버의 최초의 전송시작시점을 조정함으로써 버퍼의 크기와 미디어간의 스큐 문제를 해결 하였다. 그리고 버퍼레벨은 버퍼의 임계레벨 및 제어레벨에서 필터링함수를 이용하여 오버셋을 구하고 오버셋 만큼을 서버에 피드백하여 전송시작 시간을 늦추거나 앞당겨 버퍼를 안정상태로 유지하는 기법을 사용하게 된다. 이 기법은 버퍼의 안정상태를 유지하기 위한 방법을 제시하여 분산환경에서 재생률을 높일 수 있었다.

기존의 연구 방법은 수신측의 버퍼가 안정상태를 피드백 방법을 사용하여 동기화 문제를 해결하였다. 그러나 피드백 방법만으로는 예상치 못한 지연에 대처할 수 없게 된다. 이러한 문제점을 해결하기 위해서 본 논문에서는 버퍼의 수위를 기준으로 하는 적응용 재생시간 정책을 제안한다.

3. 분산 멀티미디어 동기화 시스템

기존의 연구에서는 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 (그림 1)과 같이 하나의 영화를 네 개의 서버로 나누어 저장함으로써 서버의 전송준비시간, 네트워크 트래픽, 버퍼의 크기 감소 등의 장점을 가질 수 있다. 멀티미디어 서버에서 보내주는 서브 스트림은 하나의 동기화 그룹을 형성하게 된다. 또한 주 미디어인 오디오 미디어를 하나의 서버로 형성시킴으로써 하나의 오디오에 세 개의 비디오 프레임 가지고 동기화 그룹으로 형성하게 하는 시스템을 제안한다. 주 미디어인 오디오가 다른 서버로 구성됨으로서 네트워크와 서버의 전송시작 준비 시간을 절약하게 하는 장점을 갖게 된다. 서버의 데



(그림 1) 제안된 시스템 구성도

이터 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 하기때문에 주 미디어의 논리적 시간이 기준이 되어 각 미디어를 서버 스트림으로 분산하는 구조를 보이게 된다.

본 논문에서는 서버가 전송시작시간을 조정하여 네트워크 지터의 시간을 보상하는 방법을 채택한다. 클라이언트는 네트워크의 상태, 즉 각 서버에서 도착되는 서버 스트림의 도착시간, 서버 스트림의 버퍼 대기시간 그리고 버퍼의 상태에 관한 정보를 관리하는 수신처리 관리자의 버퍼수위관리 유닛이 있다. 버퍼수위관리 유닛에서 발생하는 정보를 분석하여 각각의 서버에 피드백하는 QoS피드백 유닛과 버퍼의 상태를 분석하여 재생시간을 조정하는 재생 유닛이 있다. 본 논문에서 제안된 시스템에서는 멀티미디어 서버의 전송시작시간을 조정하여 미디어 내 동기화하는 구조를 가지고 또한 동기화 그룹으로 미디어간의 동기화를 하게된다.

3.1 분산 멀티미디어를 위한 버퍼크기

본 논문에서 사용되는 지터는 한정적이라는 것을 가정한다. 한정된 시간을 지나는 지터는 재생정책에 의해서 이러한 문제를 해결하게 될 것이다. 본 논문은 제한된 지터를 가정하고 지터를 최대지연 D_m^{max} 과 최소지연 D_m^{min} 의 차로 정의한 Rangan등에 의하여 주어진 지터의 정의를 사용한다[5].

$$J_m = D_m^{max} - D_m^{min} \quad \forall m \quad (1)$$

$$J^{max} = \max\{J_k \mid m \in \{0, \dots, n-1\}\} \quad (2)$$

여기에 평균지연 \bar{D}_k 로 부터 차로 정의된 지터 범위를 정의한다. 지터는 일반적으로 대칭적으로 분포되지 않는다.

따라서 J_k^+ 와 J_k^- 는 같지 않다. 서버 스트림 k의 지터와 서버 스트림 최대 지연, 서버 스트림 최소 지연 그리고 최대 지연 편차는 식 (3), 식 (4), 식 (5) 그리고 식 (6)에서 정의한다.

$$J_m = J_m^+ + J_m^- \quad \forall m \quad (3)$$

$$D_m^{max} = \bar{D}_m + J_m^+ \quad \forall m \quad (4)$$

$$D_m^{min} = \bar{D}_m + J_m^- \quad \forall m \quad (5)$$

$$J^{max+} = \max\{J_m^+ \mid m \in \{0, \dots, n-1\}\} \quad (6)$$

첫 번째 미디어가 네트워크의 최대 지연으로 도착한 서버 스트림 m이고 그 후 모든 스트림이 최소지연으로 도착하게 되는 경우를 가정할 수 있다. 물론 이러한 경우는 최악의 경우가 된다. 최악의 경우는 지터의 범위 J_m 전에 미디어 m+1이 도착되는 경우가 된다. 최대지연과 최소지연으로 도착되는 도착시간의 차이를 t_x 일 때, t_x 의 시간은 빠른 미디어와 늦은 미디어 사이에서 존재한다. 여기서 t_x 는 식 (7)과 같이 얻어지게 된다.

$$t_x = \bar{D}_m + J_m^+ + J_m + J_m^- - \bar{D}_m = 2J_m \quad (7)$$

따라서 최악의 경우에는 클라이언트의 버퍼공간이 $2J_m$ 를 갖게되고 재생시간 또한 $2J_m$ 의 시간이 있어야 한다. 그러므로 단일 스트림을 위한 미디어 내의 동기화를 보장하기 위하여 $\lceil 2J_m \cdot r \rceil$ 개의 미디어 단위의 최소 버퍼 양이 소요된다. 그러므로 단일시스템에서의 버퍼 크기는 $\lceil 2J_m \cdot r \rceil$ 이 된다.

$$B^s = \lceil 2J_m \cdot r \rceil \quad (8)$$

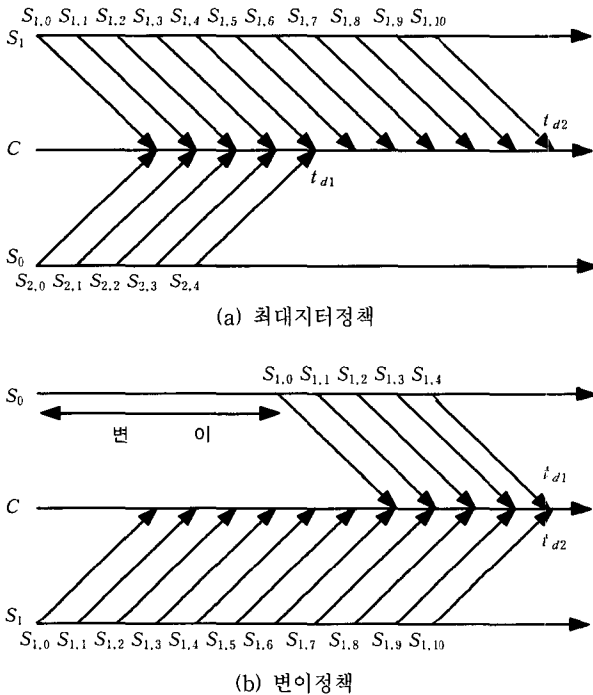
재생은 최대지연과 최소지연으로 도착되는 사이에서 모든 서버 스트림에 대하여 만족할 때 시작될 수 있다. 다른

재생 제한시간과 버퍼 요구를 유도함으로써 대응하는 서브 스트림을 위하여 다른 지터 제한에 의하여 복잡화된다. 작은 지터 한계를 가진 다른 서브 스트림들의 버퍼가 이미 넘치는 동안에 큰 지터 한계를 가진 서브 스트림이 여전히 그들의 제한시간을 기다리는 경우를 피하여야 한다. 이 문제를 해결하기 위하여 Ishibashi 등은 최대지연 한계를 가진 서브 스트림에 따른 버퍼의 할당을 제안한다. 따라서 그룹의 각 서브 스트림을 위한 버퍼요구 b_k^M 와 완전한 그룹을 위한 B^M 은 다음과 같이 주어진다[12, 13]. 결국 분산환경의 버퍼 크기는 식 (10)과 같은 크기를 갖게 된다.

$$b_m^M = \lceil 2J^{\max} \cdot r \rceil \tag{9}$$

$$B^M = \sum_{m=0}^{n-1} b_m^M = n \cdot \lceil 2J^{\max} \cdot r \rceil \tag{10}$$

식 (10)은 서브 스트림의 지터 값에서 차이에 의존한 최대 지터 전략은 버퍼의 낭비를 가져올 수도 있다. 이러한 문제를 해결하기 위해서 가장 큰 지터 범위를 갖는 서브 스트림의 예상재생시점에 따른 모든 서브 스트림의 시작시간을 (그림 2)(b)와 같이 변이하여 수행한다.



(그림 2) (a) 최대지터정책과 (b) 변이정책

변이를 하게 되면, 두 개의 서브 스트림은 같은 시간에 재생될 수 있다. k 번째 서브 스트림은 최대 지터와 자신의 지터 차이만큼을 시간 축으로 변이를 할 수 있을 것이다. 예를 들면 시간에서 $2(J^{\max} - J_m)$ 유닛만큼을 변이 하게 된다. 명백하게 서브 스트림 k는 가장 큰 지터를 가진 서브 스트림보다 $2(J^{\max} - J_m)$ 유닛을 늦게 시작할 수 있다. 동기

화그림은 n 서브 스트림으로 구성되어있다. 각각의 서브 스트림은 재생률 r로 재생한다. 각 서브스트림 k는 지터 J_m 의 범위를 가지고 있고 서브 스트림은 최대지터인 J^{\max} 보다 $J^{\max} - J_m$ 초 늦게 시작한다. 예를 들면 $J^{\max} - J_m$ 는 하나의 변이를 적용한다. 서브 스트림은 그들이 재생되기 위한 평균 지연은 동기화 됐다고 가정해야 한다. 요구되는 버퍼 공간을 적용한 서브 스트림 m를 위한 미디어 유닛의 버퍼공간은 $\lceil J^{\max} - 2(J^{\max} - J_m) \rceil \cdot r$ 의 버퍼공간이 필요하다. 이 수식을 정리하면 $b_k^s = 2J_k$ 의 공간을 갖게 된다. 따라서 변이된 버퍼의 크기는 식 (9)에서 $2(J^{\max} - J_m)$ 을 빼주면 $b_k^s = \lceil J^{\max} - 2(J^{\max} - J_m) \rceil \cdot r$ 의 수식이 된다. 이것을 다시 정리하게 되면 $b_k^s = \lceil 2J_m \rceil \cdot r$ 의 공간이 된다. 변이 전략을 적용한 버퍼의 공간은 다음과 같다.

$$b_m^s = \lceil 2J_m \rceil \cdot r \tag{11}$$

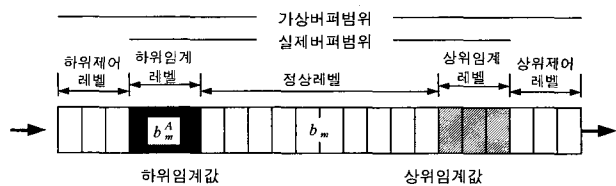
$$B^s = \sum_{m=0}^{n-1} \lceil 2J_m \rceil \cdot r \tag{12}$$

$$B = \sum_{k=1}^n b_k = \sum_{k=1}^n \lceil (2J_m + J^{\max} - J_m^+ + \max\{J_m + J_m^{J^{\max}} \mid m \neq n \wedge m = 0 \dots n-1\}) \cdot r \rceil \tag{13}$$

식 (13)은 Geyer의 연구에서 제시된 버퍼의 크기이다. 그러나 식 (13)은 최악의 상황을 고려하였으나 본 논문에서는 재생정책으로 버퍼의 크기를 줄일 수 있었다.

<표 1> 최대 지연정책과 변이 정책의 인수

기 호	내 용	단 위
m	서브스트림이나 서버의 색인	
r	서브 스트림의 요구되는 재생속도	
D_m^{\max}	서브 스트림 m의 최대 지연	[sec]
D_m^{\min}	서브 스트림 m의 최저 지연	[sec]
\bar{D}_m	서브 스트림 m의 평균 지연	[sec]
J_m	서브 스트림 m의 지터	[sec]
J^{\max}	모든 서브 스트림 최대 지터	[sec]
J_m^+	서브스트림 m의 평균지연에서 상위편차	[sec]
J_m^-	서브스트림 m의 평균지연에서 하위 편차	[sec]
$J^{\max+}$	모든 서브스트림 최대상위편차	[sec]
B^s	단일시스템에서의 버퍼요구	[sec]
b_m^M	최대지터정책의 버퍼요구	[mu]
B_m^M	최대지터정책의 버퍼요구	[mu]
b_m^s	변이정책의 위한 버퍼요구	[mu]
B_m^s	변이정책의 버퍼요구	[mu]



(그림 3) 버퍼상태수위

3.2 버퍼수위관리 유닛

(그림 3)과 같이 버퍼 상태는 정상레벨, 상위임계레벨, 상위제어레벨, 하위임계레벨, 하위제어레벨의 5가지의 상태를 갖게 된다.

3.2.1 정상 레벨

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

3.2.2 상위임계·제어레벨

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 QoS피드백 유닛과 재생 유닛에게 통보하게 된다.

3.2.3 하위임계·제어레벨

버퍼관리 유닛은 버퍼가 하위임계레벨에 이르게 되면 버퍼가 언더플로우를 발생할 가능성이 높게 된다. 버퍼의 레벨이 하위제어레벨에 이르게되면 버퍼가 기아현상이 발생된 경우이다. 버퍼수위 유닛의 버퍼수위 탐색기는 상위임계레벨에 버퍼의 수위가 옮겨가면 네트워크의 상태와 버퍼의 상태를 QoS피드백 유닛과 재생 유닛에게 통보하게 된다.

3.3 QoS피드백 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[4, 8, 9]. 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.

$$\bar{b}_{t,m} = S(q_{t,m}) = \alpha \cdot \bar{b}_{t-1,m} + (1-\alpha) \cdot q_{t,m} \quad (13)$$

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

식 (13)에 의해 시간 t에서 서브스트림 m을 위한 버퍼레벨은 $q_{t,m}$ 에 의해 나타낸다. $q_{t,m}$ 의 값은 네트워크의 짧게 변화되는 지터에 의해서 버퍼의 수위가 연속적으로 변화되는 것을 예방하기 위한 부드러운 버퍼 레벨 $\bar{b}_{t,m}$ 를 계산하도록 필터링 함수로 전달된다. 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^u 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^l 가 충분하게 제공되어야 한다. 필터링 함수에 의해서 구해진 $\bar{b}_{t,m}$ 은 버퍼의 잠정적인 수위이므로 현재의 버퍼수위인 $q_{t,m}$ 의 차이를 구하게 된다. 이 차이는 결국 서버의 시작시점을 조정하는 변수가 된다.

본 논문에서는 α 를 0.6이나 0.7의 값이 버퍼요구와 재동기화에 반영되어 버퍼의 스무딩 버퍼레벨을 구하게 된다. 이 수치는 강하게 반응하지 않으면서 민감하게 재동기화의 수준을 변형하게 한다.

3.4 재생 제어 정책

본 논문은 프레임의 디스플레이 시간이 주기적인 상태를 엄격하게 유지하지 않는것을 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.

주(master) 미디어란 동기화의 기준이 되는 미디어를 주 미디어라 하고 주 미디어에 동기화되는 미디어를 종(slave) 미디어라 한다. 주 미디어인 오디오를 재생버퍼 레벨기법에 적용하고, 이 절에서는 재생버퍼 레벨기법에 대하여 설명한다. 각 패킷의 동기화 구간 크기를 Δ_s 라 할때, i번째 패킷이 T_i 라는 시간에 출력되었으면 i+1번째 패킷은 $T_i + \Delta_s$ 시간에 출력되어야 미디어간 동기화가 이루어질 것이다. 그러나 연속 미디어 데이터는 그 종류에 따라 어느 정도의 불연속을 허용한다.

$$\rho_i = 1 / r \quad (14)$$

$$\gamma_i = \rho_i + \lambda \quad (15)$$

식 (14)는 한 개의 서브 스트림의 재생시간을 구한 식이다. 이것은 미디어 시간을 재생률로 나누어 한 개의 서브 스트림의 재생 시간을 얻을 수 있다. 또한 식 (15)는 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 프레임을 많이 확보하고자 할 것이다. 이러한 문제점을 해결하기 위해서는 두가지 방법이 있다. 첫 번째 방법은 네트워크에서의 흐름 제어 및 데이터의 양을 변환시키므로 문제를 해결하는 방법

이다. 두 번째 방법으로는 버퍼의 점유율을 가지고 재생 시점을 변화시키는 방법이다. 이러한 방법은 최대지연 지터인 10ms 이하에서 동기화 시점을 조정할 수 있다는 아이디어를 본 논문에서 적용하게 되었다.

버퍼의 점유율이 상위임계레벨에 존재한다면 식 (16), 식 (17)을 적용하여 재생시간에 대한 가중치를 변경하게 된다.

$$\omega' = BL_{allsize} - BL_{psize}; \quad (16)$$

$$\omega = \omega' / BL_{allsize}; \quad (17)$$

식 (16)에서 $BL_{allsize}$ 은 상위임계레벨에 속하는 버퍼의 개수를 갖게되고 BL_{psize} 는 상위임계레벨에 차지하고 있는 현재의 개수를 갖게 된다. 식 (17)에서는 가중치를 최대지연 지터와 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 (18)을 적용하여 재생시간에 대한 가중치를 변경하게 된다.

$$\omega = BL_{allsize} / BL_{psize}; \quad (18)$$

$$\tau_i = \rho_i - (\rho_i \times \omega) \quad (19)$$

식 (19)에서 ρ_i 는 예상되어졌던 재생시간이다. 즉, 1초에 24프레임을 요구하는 시스템일 경우 41.6ms가 된다. 또한, 실제로 계산되어진 재생시간이 된다. 식 (19)를 적용함으로써 재생시간은 점차적으로 줄게되며 재생이 빨라지므로 버퍼의 수위는 안정상태로 옮겨가는 효과를 가질 수 있다. 이러한 기술을 멀티미디어 통신에 적용함으로써 예상하지 않은 지터에 의한 버퍼의 오버플로우 상태에 도착되는 미디어를 삭제하는 경우를 어느 정도 줄일 수 있다.

식 (20)은 주 미디어의 재생시간을 동기화 그룹 내 종 미디어의 수로 나누어 종 미디어의 서버 프레임의 재생시간을 구하는 식이다.

주 미디어가 하위 제어레벨에 있을 경우에는 재생시간 만큼 기다리게 함으로서 주 미디어가 기아상태에서 탈출할 수 있도록 한다. 버퍼의 수위는 도착시간과 재생시간에 비례하기 때문에 재생시간을 늘리게 되면 미디어의 도착되는 개수를 늘릴 수 있는 장점을 갖게 된다.

$$WAIT_{Audio_i} = 125 \quad (20)$$

$$VLTU(i) = 125 / p_n \quad (21)$$

식 (20)은 주 미디어의 재생시간을 기다리게 하는 식이고 식 (21)는 기다리는 동안 종 미디어의 재생시간을 얻게 되는 식이다.

주 미디어가 상위 제어레벨에 있을 경우는 오버플로우의

상태를 경험하게 된다. 이때에는 주 미디어와 종 미디어의 재생시간을 모두 스킵하게 된다. 본 논문에서 제안된 시스템의 재생 기법은 미디어의 재생을 유연하게 하는 결과를 가질 수 있었다.

4. 성능 평가

4.1 시뮬레이션 환경

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

1Kbyte 오디오 데이터는 PCM 인코딩 기법에 의해서 인코딩되고 비디오 프레임의 해상도는 120×120을 사용했다. 초당 24프레임의 인코딩 작업을 하여 사용되어진 프레임이 된다. 비디오 프레임의 수는 하나의 오디오 프레임과 비교되는데 이유는 다음과 같다. 클라이언트에서 어플리케이션은 125ms마다 오디오 디바이스로부터 오디오 패킷을 얻고 125ms동안에 어플리케이션은 운영체제의 런타임 프로세싱 오버헤드에 의해서 결정된 세개 이하의 비디오 프레임을 비디오 그래픽으로부터 얻는다.

본 논문은 시뮬레이션 환경을 Ethernet상의 WAN 환경으로 가정하였다. 각각의 패킷에 대한 적절한 작업을 수행하기 위해서 실제 시뮬레이션에 사용된 정보는 포하송 분포로 산출하여 네트워크 지연 시간을 두 가지 미디어에 똑같이 적용하였다. 성능 측정 실험에서 사용되는 프레임의 개수는 1000개이고, 실험에서 최대지연 지터시간 100ms를 적용하였다.

〈표 2〉 시뮬레이션을 위한 파라미터

(단위 : ms)

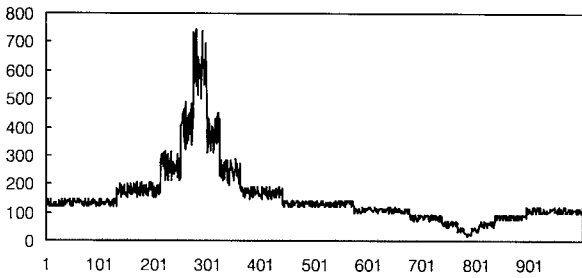
미디어 형태	최대지연	최소지연	평균지연	편차	최대지연지터
오디오	600	20	100	20	10
비디오	800	20	120	100	10

4.2 실험 결과

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.

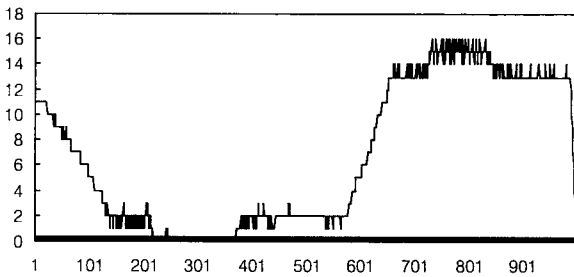
(그림 4)는 멀티미디어 서버에서 네트워크를 통한 미디어의 도착시간을 보여주고 있다. 본 논문에서 실험되는 도착시간은 포하송 분포를 기본으로 한 랜덤발생기를 이용하여 다음과 같은 그림을 얻을 수 있었다. 최대지연이 평가 단계보다 더욱 빠르거나 훨씬 느리게 발생하였으며 이러한 방법은 실제 네트워크에서 실제적으로 발생하는 시간을 측정

함으로서 얻어질 수 있었다.

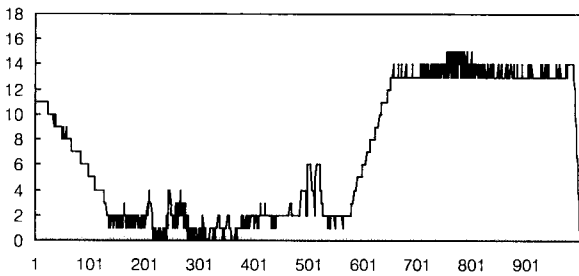


(그림 4) 네트워크의 도착시간

(그림 5(a)와 (그림 5(b)는 패킷이 도착되어 버퍼의 수위를 보여주고 있다. x축은 패킷의 번호이고 y축은 버퍼의 수위를 의미한다. (그림 5(a)의 경우 네트워크의 피드백 기법을 이용한 Geyer정책을 적용했을 경우이고 (그림 5(b)는 재생정책을 동시에 적용했을 때의 버퍼 수위의 변화를 보이고 있다. (그림 5(a)의 200번부터 350번의 프레임 도착시간이 느려짐으로서 버퍼에서는 정상상태(y축의 4와 13사이)를 벗어나는 경우를 보이고 있다. (그림 5(b)는 200번부터 350번의 프레임을 식 (16), 식 (17)에 의한 재생시간 가중치인 w 를 조정함으로써 계속해서 안정상태로 수위를 옮기고 있는 것을 보여주고 있다. 또한 (그림 5(a)는 670에서 850 사이의 오버플로우 상태를 재생시간 변경에 의해서 버퍼가 변화되는 것을 보이고 있다. (그림 5(b)의 670에서 850사이 는 식 (18)과 식 (19)의 재생시간을 변경함으로써 오버플로우를 안정상태로 수위가 옮겨가는 것을 볼 수 있다.



(a) Geyer정책에서의 버퍼 수위변화

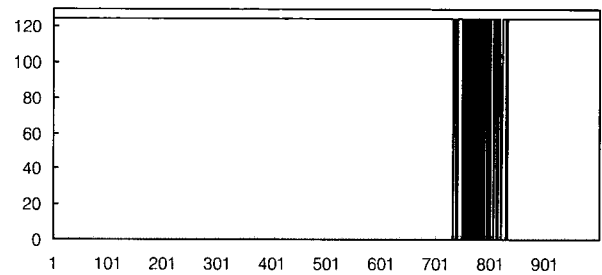


(b) 제안 기법에 의한 버퍼 수위변화

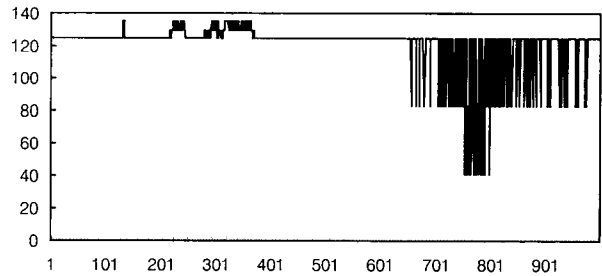
(그림 5) 버퍼 수위 변화

(그림 6(a)와 (그림 6(b)는 제안된 시스템의 재생 유닛

프로세서가 버퍼수위에 의한 재생정책을 한 결과를 보이고 있다. (그림 6(a)와 (그림 6(b)의 x축은 패킷번호이고 y축은 재생시간을 나타낸다. (그림 6(a)의 프레임 번호 200에서 350사이에서는 wait되는 경우이며 이것은 실제적으로 앞 프레임을 복사하여 wait됨으로 멀티미디어 데이터가 진행되지 못하는 경우가 된다. (그림 6(b)의 200번과 350사이 는 재생시간을 단계적으로 10ms만큼 늘려 줌으로서 네트워크의 지연을 흡수하고 있는 상황을 보이고 있다. 또한 (그림 6(a)의 경우 670부터 820까지는 skip되는 경우를 보이고 있다. 그러나 (그림 6(b)의 경우 670에서 820의 경우 재생 시간을 가중치인 w 를 적용하여 재생시간을 변경함으로써 네트워크의 상황이 빨라지는 경우에 안정적인 재생을 보여 주고 있다.



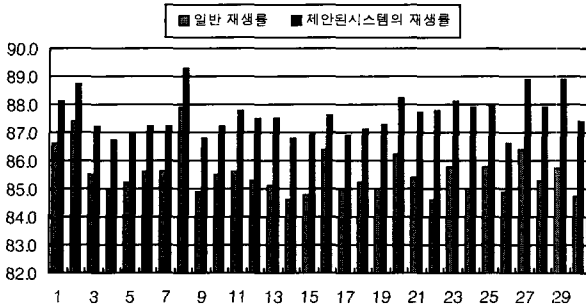
(a) 일반재생정책에 의한 재생시간변화



(b) 제안된 정책에 의한 재생시간변화

(그림 6) 재생시간 변화

(그림 7)은 실험을 31번하여 얻어진 결과를 보이고 있다. 도착시간을 변경하면서 31번의 실험에 의해서 (그림 7)와 같은 결과를 얻을 수 있었다. (그림 7)의 실험번호 2, 실험번호 8에서는 최소지연을 50ms에서 최대지연을 600ms로 조정하여 재생률이 일반 기법보다 제안된 시스템의 경우는 약 10%를 향상시키는 결과를 얻을 수 있었다. 또한 (그림 7)의 4, 9, 26번째의 실험에서는 최소지연을 20ms에서 최대지연을 800ms로 조정함으로써 오버플로우와 기아현상을 발생시킨 결과를 보여주고 있다. 이 경우에는 네트워크 상태의 트래픽을 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.



(그림 7) 재생률 비교결과

5. 결 론

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 그리고 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.

본 논문에서는 서버에서의 미디어 동기화방법과 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 동기화하여 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 미들웨어인 관리서버에 대한 모든 플랫폼 및 기능에 관하여 계속적으로 연구하여야 할 것이다. 동영상 이동을 통신에서 사용되게 함으로써 보다 자유로운 멀티미디어 영상을 감상하게 하고자하는 것이다. 이러한 이동통신 내에서의 멀티미디어 통신은 최소 버퍼를 이용한 최적의 동기화 기법을 연구해야 할 것이다.

참 고 문 헌

[1] G. Blakowski and R. Steinmetz, "A Media Synchronization Survey : Reference Model, Specification, and Case Stud

ies," IEEE Journal on selected Areas in Communications, Vol.14, No.1, Jan., 1996.

[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] C.-C. Yang and J.-H. Huang, "A Multimedia Synchronization Model and Its Implementation in Transport Protocols," IEEE Journal on selected Areas in Communications, Vol.14, No.1, Jan., 1996.

[4] 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.

[5] P. V. Rangan, H. M. Vin and S. Ramanathan, "Designing an On-Demand Multimedia Service," IEEE Communications Magazine, Vol.30, No.7, pp.56-65, July, 1992.

[6] N. U. Qazi, M. Woo and A. Grafoor, "A Synchronization and communication model for distributed multimedia objects," in Proc. ACM Multimedia, 1993.

[7] R. Steinmetz, "Human Perception of Jitter and Media Synchronization," IEEE Journal on selected Areas in Communications, Vol.14, No.1, Jan., 1996.

[8] W. Geyer, "Stream synchronization in a scalable video server array," Master's thesis, Institut Eurecom, Sophia Antipolis, France, Sep., 1995.

[9] Ernst Biersack, Werner Geyer, "Synchronization delivery and playout of distributed stored multimedia streams," Multimedia Systems, Vol.7, No.1, pp.70-90, 1999.

[10] Azzedine Boukerche, Sungbum Hong and Tom Jacob, "MoSync : A Synchronization Scheme for Cellular Wireless and Mobile Multimedia System," Proceedings of the Ninth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems IEEE, 2001.



이 기 성

e-mail : ygslee@sunny.howon.ac.kr
 1993년 숭실대학교 전자계산학과 졸업 (학사)
 1996년 숭실대학교 대학원 컴퓨터학과 졸업(공학석사)
 2001년 숭실대학교 대학원 컴퓨터학과 졸업(공학박사)

2001년~현재 호원대학교 컴퓨터학부 전임강사
 관심분야 : 멀티미디어통신, 이동통신, 정보보호