

수신측 버퍼 레벨을 이용한 멀티미디어 동기화 기법

正會員 김 승 천*, 박 기 현*, 이 현 태*, 박 재 성*,
이 재 용*, 이 상 배*

A Multimedia Synchronization Mechanism Using Receiver Buffer-Level

Seung-Cheon Kim*, Kee-Hyun Park*, Hyeun-Tae Lee*, Jae-Sung Park*,
Jai-Yong Lee*, Sang-Bae Lee* *Regular Members*

※본 논문은 국방과학연구소의 지원에 의하여 수행된 연구결과입니다.

요 약

앞으로의 통신은 멀티미디어를 가정하여 복잡하고도 다양한 서비스가 이루어질 전망이다. 따라서 본논문은 멀티미디어 통신의 표현상에 있어 상당히 중요한 부분을 차지하는 멀티미디어 동기화에 대해서 다룬다. 본 논문은 우선 통신 구조상에 동기화가 수행되어야 하는 위치를 수송 계층 이상으로 보고, 그 상태에서 고정 길이의 버퍼를 각 수신단에 할당하고 버퍼의 레벨을 구분하여 이의 적절한 조절을 이용한 동기화 기법을 제안한다. 제안하는 동기화 기법은 미디어내, 미디어간 동기화를 모두 지원하며 끝으로 실험을 통하여 제안하는 고정 길이 버퍼의 크기가 수신단에서 동기화 제공에 적합한지를 검증하고 다른 동기화 기법과의 동기 성능을 비교 실험한다.

ABSTRACT

The future data communications are expected to support the various and complex services with multimedia. So this paper has focused on the multimedia synchronization problem which has important position in multimedia presentation. Firstly, this paper consider the suitable layer for multimedia synchronization in the communication structure as transport layer or upper ones, in which we propose synchronization mechanism using fixed length buffer with buffer-level. The proposed mechanism also supports intra-and inter-media synchronization among media. Through simulation, we prove our analysis of the fixed-length buffer size that the proposed mechanism can provide. Also we show comparisons between our mechanism and other scheme.

*연세대학교 전자공학과
論文番號:96320-1011
接受日字:1996年 10月 11日

I. 서 론

비동기 전송 방식인 ATM(Asynchronous Transfer Mode) 전송 방법을 기반으로 한 광대역 통신망(B-ISDN)의 발전에 따라 고속의 통신 서비스가 되었고, 또한 이러한 고속 통신은 영상, 음성, 문자 등의 데이터를 통합하여 이용하는 멀티미디어 응용 서비스를 가능토록 하였다 [1]. 이러한 서비스들은 자체적 또는 상호적 시간 관계를 갖는 여러 미디어의 데이터로 그 전송 방식이 특성지어 지는데, 송신단의 데이터 생성 특성에 따라 실시간(Live) 또는 저장된 후 이용이 되는(Stored) 응용으로 구분지어 질수 있다[2]. 그러나 고속의 통신망을 이용한 서비스일지라도 여러 미디어들을 함께 표현하는데 있어서는 미디어의 표현시 시간적인 연관성의 해결이라는 선결되어야 하는 점이 남아 있다. 이러한 문제에 대한 해결이 미디어 동기를 이루는 것이다[3].

여러 미디어들을 이용한 통신에 있어서 미디어들간의 동기화는 망내의 종단간 지연과 이러한 지연의 변화인 지터(Jitter)에 의해 상당히 어렵게 된다. 망내의 전송 단위에서의 지터는 결국 멀티미디어 데이터의 표현 단위인 데이터 프레임에 까지 영향을 미치게 되는데, 이러한 지터는 결국, 응용의 입장에서 불연속적인 표현을 야기하고 여러 미디어간의 시간적 연관성도 크게 벗어나는 표현을 하도록 한다. 시간적 연관성의 훼손은 결국 멀티미디어 통신 사용자로 하여금 불편을 느끼게 할뿐만 아니라, 이런 멀티미디어를 이용한 여러 작업들을 불가능하게 만든다. 따라서 멀티미디어 동기화는 고속망을 이용한 멀티미디어 서비스를 위해서는 선결되어야만 한다.

본 논문에서는 이러한 멀티미디어 동기를 해결하기 위한 구조를 제시하고 구체적인 방법으로, 멀티미디어 응용의 입장에서 각 미디어에 일정 크기의 버퍼를 할당하고 이러한 버퍼의 레벨을 관찰하여 버퍼 레벨의 변화에 따른 적절한 조치를 수행하여 멀티미디어의 동기를 해결하는 방법을 제안하고자 한다. 제안하는 방법은 응용의 사용자에게 어느 정도의 감퇴적인 현상(Graceful Degradation)[4]을 허용하면서 사용자가 요구하는 서비스 품질을 만족시키는 동기화의 제공을 목표로 하고 있다.

II. 멀티미디어 통신의 동기화 방식

미디어내 동기화(Intra-Media Synchronization)는, 흐름 동기[4](Stream Synchronization)로 표현되기도 하는 것으로 단일 미디어의 연속적인 표현을 위한 동기화가 된다. 송신단에서 전송된 미디어 데이터는 송수신단간의 지연과 이런 지연의 변화인 지터(Jitter)의 영향을 받는다[5]. 따라서 연속적으로 보낸 송신단의 미디어 데이터는 수신단에서 불연속성을 갖게 된다. 이러한 불연속성을 없애고 송신단에서와 같은 순서의 연속성을 갖도록 하는 것이 미디어내 동기화가 된다.

단일 미디어가 아닌 여러 미디어를 이용한 통신이 될 경우, 미디어내 동기화 뿐만 아니라 미디어간의 시간적 연관성은 매우 중요하게 된다. 즉 여러 미디어 데이터 중에서 같은 시간에 표현되어야 할 데이터가 있는 경우 이들의 도착 시간이 다르다고 하면 이들의 동시 표현은 어렵게 된다. 이러한 미디어 데이터들 상호간에 시간적 연관성을 유지시켜 주는 것이 미디어간의 동기화(Inter-Media Synchronization)가 된다.

멀티미디어 동기화에 관한 연구는 다수개의 가상 채널을 하나의 채널에 실어서 미디어의 시간적인 연관성을 유지하려 하였던 Leung의 MVC 방법[6]이 있었고, 미디어 정보를 데이터내에 또는 데이터 밖에 실어서 보내는가에 따른 Sheperd의 SM, SC 방법[7]이 있었다. 그리고 응용과 사용자의 입장에서 멀티미디어의 형태를 시간적인 연관성을 갖도록 구성하는 Little의방법[8] 과 망에서 시간적인 차이를 보상하여 주기 위한 방법으로 Escobar의 흐름 동기화[9] 방법도 제안 되었으며, 주기적인 Feedback 정보에 의존하는 Ramanathan의 동기화 기법[10]등이 제안 되었었다. 현재까지 기존의 방법들은 멀티미디어 동기화를 완벽히 만족 시켜주지 못하는 것이 사실이며, 이를 위해서는 망 내에서의 서비스 품질(QoS)의 보장이 선결되어야 한다. 따라서 본 논문은 가능한 멀티미디어 동기화를 서비스 품질의 변수를 이용하여 규정하고 이를 이용하여 감퇴 현상의 조절이 가능한 멀티미디어 동기화 기법을 제안하고자 한다.

III. 제안하는 동기화 기법

제안하는 동기화 기법은 수신단의 기능을 강화시켰다. 따라서 제안 기법의 세부사항을 설명하는데 앞서서 제안하는 동기화 기법에 사용되는 버퍼의 크기를 먼저 계산하고 이를 바탕으로 제안하는 기법을 설명하기로 한다.

1. 수신단에서의 버퍼의 크기 결정

기본적으로 미디어 간의 동기확보 이전에 단일 미디어에서의 연속성 보장이 선행 되어야 한다. 여기서는 단일 미디어에서의 연속성 보장을 위해서 Playout 이전에 버퍼링 해야 할 프레임의 갯수와 이를 위한 버퍼의 크기를 분석해 보도록 한다. 표 1은 버퍼의 크기 분석에서 사용되는 변수들의 나열이다.

표 1. 버퍼크기 계산에 이용되는 변수들

변수	설명
$\tau(m)$	Transmission time of m'th frame
$p(m)$	Playout time of m'th frame
$a(m)$	Arrival time of m'th frame
Δ	Average delay from source to destination
δ_m	Delay variance of m'th frame
θ	Maximun delay variance
δ	Playout interval

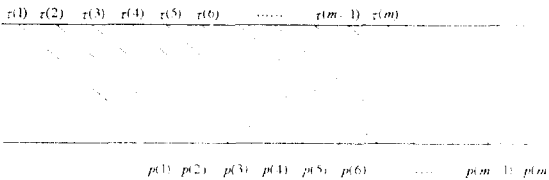


그림 1. 멀티미디어 통신의 데이터 전송

미리 받아야만 하는 데이터의 갯수를 계산하기 위해서는, 먼저 그림 1에서 처럼 미디어 데이터의 시간적인 순서를 고려한 전송을 생각한다. 만약 미디어 데이터가 도착하자 마자 Playout한다고 하면 식(1)과 같은 시간적인 관계를 갖게됨을 알 수 있다.

$$p(1) = \tau(1) + (\Delta + \delta_1)$$

$$p(2) = \tau(2) + (\Delta + \delta_2) = \tau(1) + \Delta + \delta_2 + \theta$$

$$p(3) = \tau(3) + (\Delta + \delta_3) = \tau(1) + \Delta + \delta_3 + 2\theta$$

$$p(4) = \tau(4) + (\Delta + \delta_4) = \tau(1) + \Delta + \delta_4 + 3\theta$$

$$\vdots$$

$$p(m) = \tau(1) + (\Delta + \delta_m) = (m-1)\theta \tag{1}$$

이때 Playout시간의 최소값은 $p^e(m) = \tau(1) + \Delta_{\min} + (m-1)\theta$ 이 된다고 할 수 있다. 이때 미디어 데이터 프레임이 제대로 Playout되기 위해서는 적어도 프레임의 도착 시간보다 Playout시간이 늦어야만 한다. 즉, $a(m) \leq p(m)$ 이어야만 끊김 없는 표현이 가능하다. 수신단에서의 Playout 간격을 생각할 때 Playout 시간이 도착 시간보다 빠른 경우도 있을 것이다. 하지만 원래 끊김없는 표현이 가능하기 위해서는 이러한 경우에도 위에서의 제한이 만족해야만 한다. 만약 Playout 시간이 도착 시간보다 빠르다고 한다면 Playout 시간의 최소값에서 Δ_{\min} 는 $\Delta - \delta$ 가 되고, 도착 시간은 $a(m) = \tau(m) + (\Delta + \delta)$ 이고, $p(m) = \tau(1) + \Delta_{\min} + (m-1)\theta$ 은 $p(m) = \tau(1) + \Delta - \delta + (m-1)\theta$ 로 변환이 될 것이다. 이때도 항상 $a(m) \leq p(m)$ 이어야 하므로 식(2)와 같은 결과를 유도해 낼 수 있을 것이다.

$$\tau(m) + (\Delta + \delta) \leq \tau(1) + (\Delta - \delta) + (m-1)\theta$$

$$\therefore \tau(m) - \tau(1) \leq (m-1)\theta - 2\delta \tag{2}$$

식 (2)가 의미하는 바를 살펴보기전에 m=2를 대입하여 보면 다음의 결과를 얻을수 있다.

$$\tau(2) - \tau(1) \leq \theta - 2\delta \tag{3}$$

이것은 만약에 지연의 변화인 지터(Jitter)가 없다면 모를까 지터가 있을 경우는 이러한 지터를 고려하여 미디어 데이터를 Sampling간격보다 빠르게 보내야 함을 의미한다. 이는 사실적으로 불가능하다. 따라서 이의 해결을 위해서는 얼마의 미디어 데이터를 송신단은 미리 전송해야만 한다. 그리고 수신단은 데이터가 도착된 다음 곧바로 Playout 할수 없으며 Playout 시간의 재조정과 함께 전송된 미디어 데이터의 버퍼링이 필요하다. 즉, 연속적인 Playout을 보장하기 위해서는 미리 얼마의 미디어 데이터를 가져와야만 한다고 할 수 있다.

연속성 보장을 위해 미리 받아야 할 미디어 데이터

의 갯수가 얼마인지 알아보기 위해서 미리 받아야 할 데이터의 갯수를 T 라고 하면, 식(4)와 같이 전개되고 이를 위의 식(2)에 대입하여서 구해보면 식(5)와 같은 결과를 얻는다.

$$\begin{aligned}
 p(1) &= \tau(T) + (\Delta + \delta_T) \\
 p(2) &= \tau(T) + (\Delta + \delta_T) + \theta \\
 &\vdots \\
 p(m) &= \tau(T) + (\Delta + \delta_T) + (m-1)\theta \\
 p^e(m) &= \tau(T) + (\Delta + \delta) + (m-1)\theta \quad (4) \\
 a(m) &= \tau(m) + (\Delta + \delta) \\
 \tau(m) + (\Delta + \delta) &\leq \tau(T) + (\Delta - \delta) + (m-1)\theta \\
 \tau(m) &\leq \tau(T) + 2\delta + (m-1)\theta \quad (5)
 \end{aligned}$$

여기서 $m = T + 1$ 인 프레임을 생각하면 $\tau(T + 1) \leq \tau(T) + 2\delta + T\theta$ (6)가 되고 이를 T 에 대해 정리하면 다음과 같다.

$$T \geq \frac{\tau(T + 1) - \tau(T) + 2\delta}{\theta} \quad (7)$$

식(7)에서 $\tau(T + 1) - \tau(T)$ 를 θ 와 같게 놓으면 식(7)은 다음처럼 정리 될 수 있다.

$$T \geq \frac{\theta + 2\delta}{\theta} = 1 + \frac{2\delta}{\theta} \quad (8)$$

즉, 여기서 최소의 T 값은 결국 식(9)와 같이 된다.

$$\therefore T_{\min} = 1 + \frac{2\delta}{\theta} \quad (9)$$

식(9)의 결과가 의미하는 바는 매우 크다고 할 수 있다. 이는 미디어의 연속성을 보장하기 위해서 수신단에서 받아야 하는 최소의 미디어 데이터의 갯수가 식(9)과 같다는 것을 의미한다. 따라서 버퍼의 Overflow와 Underflow를 방지하면서 미디어 데이터의 연속성을 보장하기 위해서는 미리 버퍼링해야 하는 데이터의 갯수에 두배를 버퍼의 크기로 잡는 것이 안정성을 보장할 수 있다. 따라서 버퍼의 크기는 다음과 같다.

$$\therefore B = 2T = 2 \left[1 + \frac{2\delta}{\theta} \right] \quad (10)$$

이렇게 되면 시간만이 아니라 QoS 변수로써 지연

변화량의 최대값과 Sampling 간격, 즉 Playout Ratio에 의해서 버퍼의 크기와 미리 받아야 할 미디어 데이터의 갯수를 결정할 수 있다.

2. 제안하는 동기화 기법 구조

제안하는 동기화 기법의 전체 구성은 그림 2와 같다. 기본적인 재동기 작업은 모니터링 부의 버퍼 레벨의 변화 값에 의해 이루어진다. 또한 미디어의 연속성에 기초하여 멀티미디어 데이터를 마스터 미디어(Master Media)와 슬레이브 미디어(Slave Media)로 나눈 다음, 마스터 미디어에 기준한 미디어간 동기를 획득한다.

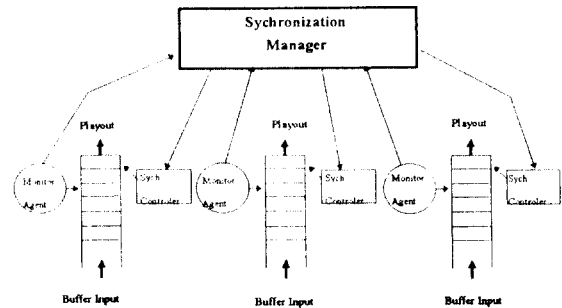


그림 2. 제안하는 동기화 기법의 구조

전체 구성은 그림 2에서처럼 모니터링 부, 동기 관리자부, 그리고 동기 제어부로 나뉘어지며 각 부분의 기능을 살펴 보면 다음과 같다.

2.1. 모니터링 부

제안 동기화 기법에서 동기화 수행의 근거가 되는 정보를 수집하여 알려 주는 부분으로, 그림 3에서 보듯이 전체 버퍼의 레벨은 4개(Under-threshold Level,

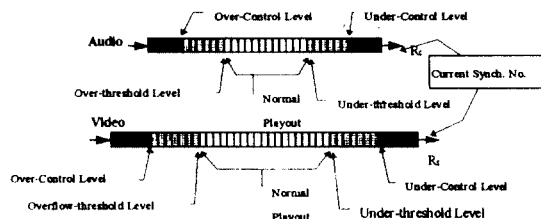


그림 3. 고정 길이 버퍼와 버퍼 레벨

Under-control Level, Over-threshold Level, Over-control Level, Normal 구간)로 나누어지며 이런 버퍼의 상태를 관리자에게 알리는 역할을 한다.

2.2. 동기 관리자부

동기 관리자는 모니터링 부에서 보내주는 정보에 의해서 그림 4와 같은 상태 천이를 보인다. 여기서 동기 관리자를 뒀으로써 모니터링 부와 제어부 사이의 인터페이스를 제공함과 더불어 상황에 맞는 적절한 기능을 추가할 수 있게 된다. 각 상태를 살펴 보면 다음과 같다.

· Normal 상태

처음에 미디어 데이터가 도착해서 동기 작업을 수행하거나 모든 미디어 데이터 버퍼(Master & Slave)가 정상적인 레벨에 있는 경우로, 이 경우 동기 관리자는 각 미디어 데이터에 대해 미리 정해진 Playout Ratio로 각 미디어 데이터를 Playout한다. 또한 Over_control레벨에서 버려졌던 데이터를 그 이전의 데이터를 복사하여 대체할것을 동기 제어부에게 명령한다.

· Over-threshold, Under-threshold 상태

모니터링 부로부터 현재 버퍼가 이 상태에 속한다는 정보를 받으면 동기 관리자는 동기 제어부에게 동기 획득 작업을 할 것을 요청한다. 즉 실질적인 재동기 작업이 이루어질 수 있는 상태가 된다.

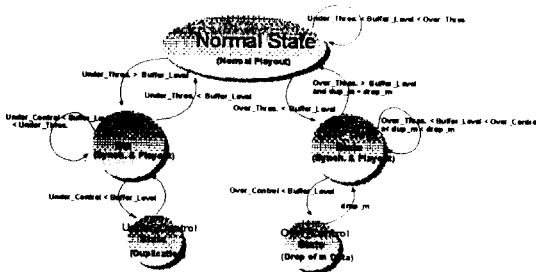


그림 4. 동기관리자의 상태 천이도

· Over-control 상태

모니터링부에서 현재 버퍼가 Over_control 레벨에 속한다는 정보를 받으면 동기 관리자는 동기 제어부에게 버퍼내 데이터를 제거할 것을 요구하게 되고 제

거된 데이터의 정보를 다시 받는다. 제거된 데이터 정보는 Normal 상태에 이르렀을 때 데이터를 복사하는데 이용된다. 데이터의 복사는 제거된 데이터의 이전에 있는 데이터를 복사하여 제거된 데이터에 대체한다.

· Under-control 상태

모니터링 부에서 현재 버퍼의 상태가 Under_control 레벨에 속해 있다는 정보를 받게 되면 동기 관리자는 동기 제어부에게 버퍼내 데이터를 복사하여 버퍼가 Underflow가 되는 것을 막도록 한다.

동기 관리자가 각 상태에 있을 확률은 버퍼내 각 영역이 얼마가 되느냐에 따라서 달라진다. 따라서 버퍼의 레벨을 조정함으로써 각각 수행되어지는 기능의 빈도를 조절할 수 있다.

2.3. 동기 제어부

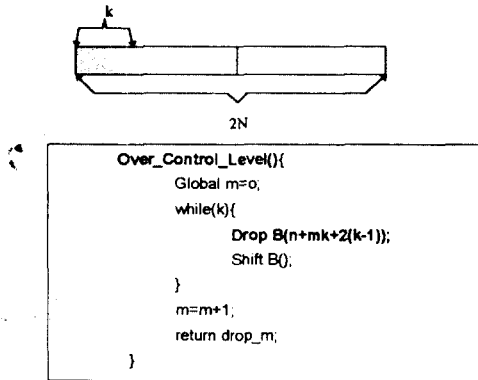
동기 제어부는 입력 버퍼의 레벨 변화에 따라 동기 관리자의 요청에 의해 적절한 제어를 가한다. 다음은 동기 제어부에 의해서 수행되는 일이다.

· 재동기화 (Resynchronization)

재동기는 마스터 미디어의 다음 Playout을 기다리는, 즉 마스터 미디어 버퍼의 맨 처음에 위치해 있는 데이터의 RTS를 검색한후 현재 버퍼에 위치해 있는 동기가 맞아야 하는 데이터를 찾아서 버퍼의 맨 앞에 위치시킨다. 그리고 슬레이브 미디어의 데이터가 앞서 있는 경우는 슬레이브 미디어의 데이터를 한번씩 늦추어 주어서 동기화를 이루도록 한다.

· 데이터 제거(Data Drop)

버퍼의 Overflow가 일어나고 있으므로 이를 방지하는 것이 제일 시급하다. 따라서 버퍼의 데이터를 제거하여 버퍼의 Overflow를 방지하는데, 기본적인 데이터 제거 방법은 사이 사이 제거하는 방법이다. 즉 연속적인 데이터 제거로 인한 연속성의 단절을 막으면서 이후 보상이 가능한 상태로 되기 위해서는 하나씩 건너서 데이터를 제거하는 방법을 사용해야 한다. 그림 5는 이러한 데이터 제거 방법의 일반적인 경우를 설명한다.



n: 미리 전송 받은 데이터의 갯수
k: Over-control 레벨 영역의 넓이

그림 5. 데이터 제거 방식

· 데이터 복사(Data Duplication)

현재 버퍼의 Underflow가 일어나고 있으므로 이를 먼저 해결하는 것이 제일 중요하다. 따라서 Under-control Level 이하의 데이터에 대한 복사를 실시한다. 복사한 데이터가 버퍼내에 추가 되므로 이로 인해 버퍼의 Underflow를 방지 할 수 있으며 또한 Playout 이 끊기는 것을 막을 수 있다.

IV. 실험 및 결과 고찰

제안하는 동기화 기법의 실험을 위해서 기본적으로는 점대점 Interactive 서비스를 가정하였다. 서비스는 평균 지연인 δ 를 300ms로 가정하고 샘플간격인 R_s 는 50ms로 이루어지며 최대 지터 지연인 Δ 는 150ms로 이루어진 서비스가 20분 수행되어지는 경우로 하여서 실험을 하였다. 또한 사용된 미디어는 연속성을 갖는 데이터로 가정하였고 마스터와 슬레이브 두개의 미디어를 대상으로 실험하였다.

먼저 본 논문에서 제안하였던 버퍼 크기를 실험을 통하여 검증하기로 한다. 그림 6, 그림 7에서는 버퍼 길이의 변화에 따른 동기화 정도를 비교하였다. 본 논문에서는 미디어 데이터의 연속성 보장을 위해서 수신단이 미리 받아야 할 최소 필요 데이터의 수를 식 (9)과 같이 정의했으며 수신단의 버퍼의 크기는 식 (10)처럼 정의 했다.

$$\therefore T_{min} = 1 + \frac{2\delta}{\theta} \quad (9)$$

$$\therefore B = 2T = 2 \left[1 + \frac{2\delta}{\theta} \right] \quad (10)$$

이번 실험은 본 논문에서 제안한 버퍼 크기에 대한 검증을 위한 실험이다. 실험 가정은 앞서와 동일하고, 여기서는 동기의 허용치인 Coarse Skew 허용치를 300 ms[20]로 잡았다.

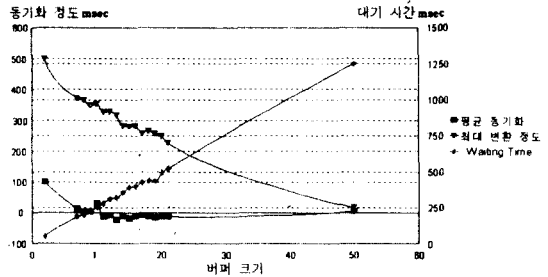


그림 6. 버퍼 크기 변화에 따른 미디어내 동기정도 및 대기 시간(1)

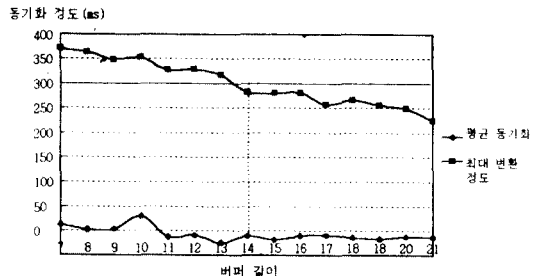


그림 7. 버퍼 크기 변화에 따른 미디어내 동기화 정도(2)

그림 7은 그림 6에서 동기화 정도에 대한 부분을 버퍼 크기 7~21까지 발췌한 그래프이다. Coarse 동기화를 위한 Skew의 값을 300ms로 할 경우, 그림 7에서 보는 바와 같이 이를 만족시키는 최소의 버퍼의 크기가 14임을 알 수 있다. 본 논문에서 제안하는 버퍼의 크기는 식 (9), (10)에 가정한 값을 대입하여 보면

$$T = 1 + \frac{2\delta}{\theta} = 1 + \frac{2 \times 0.15}{0.05} = 1 + 6 = 7$$

$$B = 2 \times T = 14$$

가 되어서 이를 만족함을 알 수 있다. 물론 이보다 더 큰 버퍼를 수신단이 준비한다면 좀더 안정된 동기화

얻을 수 있겠으나 평균 버퍼의 대기 시간이 증가하여 실시간 서비스를 방해한다는 점을 심각히 고려해야 한다.

제한한 동기화 방법이 지터 지연 변화에 어떻게 대처하는가에 대해 알아 보기 위해서 기존의 방법중 Shephard의 SM(Synchronization Marker)방식[7]과 비교하는 실험을 실시하였다.

제한 기법은 버퍼 크기를 식(10)에 의하여 14개의 미디어 데이터 크기로 하였고, 비교의 대상이 되는 SM 기법은 두가지의 경우로 나누어서 SM 기법1은 수신단에서의 버퍼 크기를 20개의 미디어 데이터 크기로 하였다. 그리고 SM 기법2는 40개의 미디어 데이터 크기로 하여 성능을 비교하였다.

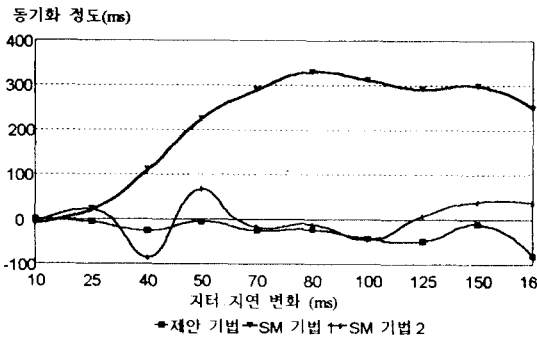


그림 8. 지터 변화에 따른 평균 동기화 정도의 비교

그림 8의 결과에서 알 수 있듯이 SM 기법1은 지터 지연의 변화에 대해 동기가 크게 어긋나고 있음을 알 수 있다. 즉, SM 기법1은 지터 지연이 매우 작은 경우 외에는 그 동기 기능을 제대로 수행할수 없음을 알 수 있다. 반면에 SM 기법2와 제안 기법은 훨씬 동기화 기능을 잘 수행하고 있음을 알 수 있다.

그림 9에서는 평균 대기 시간을 비교했는데 제안 기법이 대략 400ms의 대기 시간을 갖는 반면에 SM 기법은 500~600ms의 대기 시간을 갖는 것을 보였다.

그림 10에서는 버퍼의 Underflow로 인한 Playout 연속성의 장애 정도를 나타낸다. 제안 기법은 하나도 일어나지 않았고 SM 기법은 지터 지연의 증가에 따라서 증가하는 형태를 보였다. 결과에서 SM 기법2가 SM 기법1보다 연속성 보장이 훨씬 잘 되고 있음을 알 수 있다.

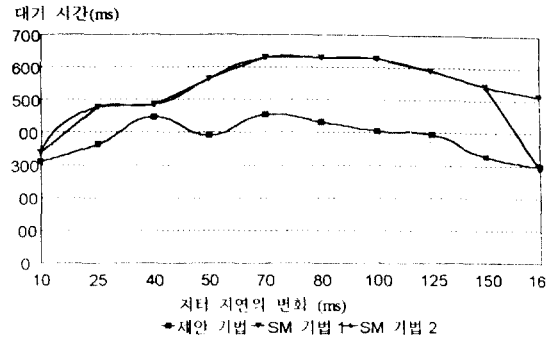


그림 9. 평균 대기 시간

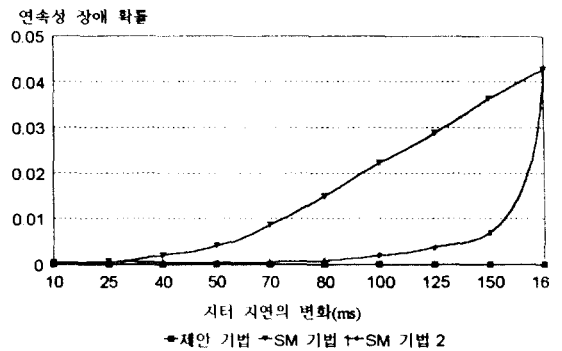


그림 10. 연속성 장애도

V. 결 론

본 논문에서는 고속망을 기반으로 한 멀티미디어 서비스에 적합한 고정 길이 버퍼의 레벨 변화에 기초한 실시간 멀티미디어 동기화 기법을 제시하였다. 먼저 미디어내 연속성의 보장을 위해 수신단이 미리 전송 받아야 하는 최소의 미디어 데이터의 수를 제시하였으며 이를 바탕으로 수신단에서는 연결 설정시에 QoS 변수 값으로 고정 길이 버퍼를 준비 할 수 있도록 하여 전체적인 서비스의 설정과 연관되게 그 기능을 수행할 수 있도록 하는 동시에, 동기화 수행을 위한 준비를 간략화 하였다. 그리고 실험을 통해서 제안한 고정 길이 버퍼의 크기가 동기 수행에 있어서 적합함을 보였다. 그다음으로 본 논문에서는 미디어내 연속성을 보장하면서 마스터 미디어에 대한 슬레이브 미디어의 상대적 동기화를 위한 버퍼의 레벨을 정의하였다. 이러한 버퍼 레벨은 지속적인 동기 작업

을 피하고 그때 그때 동기가 어긋났을 가능성이 가장 큰 경우에만 동기 확인을 하여서 재동기 작업을 수행하도록 함으로써 수신단에서의 처리 효율을 높일 수 있도록 하는 근거가 되었다. 그리고 끝으로 기존의 방법 중에서 Shepherd의 SM기법과의 비교 실험을 통해서 제안 기법이 동기화 기능이 월등함을 알 수 있었고, 버퍼의 이용 효율도 높음을 확인하였다. 또한 제안하는 동기화 기법은 큰 부하 없이 동기 정보를 데이터 채널에 삽입하여 이를 바탕으로 동기를 확보하므로 앞으로의 고속망을 기반으로 한 복잡하고 다양한 복합적 고속 멀티미디어 서비스에 적합할 것으로 기대된다.

참 고 문 헌

1. P. B. Berra, C-Y. R. Chen "Issues in Networking and Data Management of Distributed Multimedia Systems", Symposium on High-Performance Distributed Computing, September. 1992.
2. T. D. C. Little and F. Kao, An Intermedia Skew Control System for Multimedia Data Presentation, Proc. 3rd Intl. Workshop on Network and Operating System Support for Digital Audio and Video, Dec. 1992, pp. 121-132.
3. Doug Shepherd and Michael Salmony, "Extending OSI to support Synchronization required by Multimedia Applications", Computer Communications, Vol. 13, No. 7, September 1990, pp399-406.
4. Geoff Coulson, "Meeting the Real-Time Synchronisation Requirements of Multimedia in Oen Distributed Processing", Distributed Systems Engineering Journal, 1994.
5. Warren A. Montgomery Techniques for Packet Voice Synchronization, IEEE JSAC, Vol. 1, No. 6, Dec., 1983, pp 1022-1028.
6. Leung, W.F., et al., A Software Architecture for Workstations Supporting Multimedia Conferencing in Packet Switching Networks, IEEE JSAC, Vol. 8, No. 3, Apr. 1990, pp 380-390.
7. Shepherd, D., and M. Salmony, Extending OSI to Support Synchronization Required by Multimedia

Applications, Computer Comm., Vol. 15, No. 10, Dec. 1992, pp 399-406.

8. T.D.C. Little and A. Ghafoor, "Multimedia Synchronization Protocols for Broadband Integrated Service", IEEE JSAC, Vol. 9, No. 9, December. 1991.
9. Escobar, J., D. Deutch, and C. Partridge, Flow Synchronization Protocol, Proc. 1992, IEEE Globecom, Vol. 3, pp 1381-1387.
10. S. Ramanathan and P. Venkat Rangan Feedback Techniques for Intra-Media Continuity and Inter-Media Synchronization in Distributed Multimedia Systems The Computer Journal, Vol. 36, No. 1, 1993, pp 19-31.



김 승 천(Seung-Cheon Kim)정회원
 1994년 2월:연세대학교 전자공학과 학사 졸업
 1996년 2월:연세대학교 전자공학과 석사 졸업
 1996년 2월~현재:연세대학교 전자공학과 박사과정 재학중

※주관심분야: BISDN/ATM, Multimedia communication, Satellite Communication

박 기 현(Kee-Hyun Park) 정회원
 1990년 2월:연세대학교 전자공학과 학사 졸업
 1992년 2월:연세대학교 전자공학과 석사 졸업
 1992년 2월~현재:연세대학교 전자공학과 박사과정 재학중

※주관심분야: BISDN/ATM, ATM traffic control, ABR

이 현 태(Hyeun-Tae Lee) 정회원
 1983년 2월:경북대학교 전자공학과 학사 졸업
 1986년 2월:연세대학교 전자공학과 석사 졸업
 1986년 2월~현재:한국전자통신연구소 통신시스템연구단 선임연구원
 1993년 2월~현재:연세대학교 전자공학과 박사 과정 재학중

※주관심분야: B-ISDN/ATM, Switching, Gigabit networks, Multimedia application

