논문 00-25-12A-14 한국통신학회논문지 '00-12 Vol.25 No.12A

멀티미디어 동기화 모델에서 유연한 재생 처리 기법

정회원 이 기 성*, 오 해 석*

A Smoothing Presentation Processing Mechanism for Multimedia Synchronization Model

Gi-seong Lee*, Hae-seok Oh* *Regular Members*

요 약

실시간 응용 프로그램은 미디어 데이터간에 만족되어야 할 동기화 제약조건(synchronization constraints)을 가지고 있다. 이러한 제약조건은 프리젠테이션 되어야 할 미디어 데이터간의 지연시간 및 서비스 품질을 나타낸다. 미디어 데이터간의 지연시간 및 서비스 품질을 효율적으로 표현하기 위해서는 이에 적합한 새로운 동기화 기법이 요구된다. 동기화 기법은 미디어의 재생시 끊어짐이 없는 유연한 재생을 처리하는 기법이다. 제안된 기법은 사용자 측면에서 엄격한 재생을 위하여 프레임을 생략하는 것보다는 부드러운 영화를 볼 수 있도록 유연한 재생 처리를 한다. MMSM(MultiMedia Synchronization Model) 모델의 동기화 기법은 종 미디어 스트림을 부드럽게 재생하는 결과를 나타내었다.

ABSTRACT

Real-time application programs have constraints which need to be met between media-data. These constraints represents the delay time and quality of service between media-data to be presented. In order to efficiently describe the delay time and quality of service, a new synchronization mechanism is needed. multimedia synchronization manages a flexible playout without breaks when playing out media. Thus our proposed scheme handles flexible playout in a point of view of user in order for an audience to view movies more naturally, rather than discarding frames for strict playout. The model results in naturally playing out slave media stream as well.

I. 서 론

멀티미디어 시스템에서 이용되는 비디오나 오디오 정보들은 기존의 텍스트나 그래픽과 같은 정보와는 다른 특성을 가지고 있다. 비디오나 오디오 등의 정보는 일정한 시간 간격으로 연속적으로 데이터가 발생하며, 수신측에서도 똑같은 간격으로 연속적으로 재생되어야 한다[1,2,3].

멀티미디어 서비스에서 서비스 품질과 관련하여 동기화가 핵심적인 기능으로 요구되는 이유는 ATM(Asynchronous Transfer Mode)망과 같은 고속 통신망을 통해 데이터가 수신측에 전달될 때 송신측으로부터 도착시간의 지연이 다르기 때문에 네트워크에서 임의로 발생하는 지연이나 송수신 시스템간 클락(clock)의 불일치 등에 의해 미디어간에 존재하는 원래의 시간 관계가 파괴되기 때문이다. 이와같은 이유로 시간관계가 훼손된 멀티미디어 데이터에 대해 응용 서비스의 요구사항이나 각 미디어의 손실 및 지연에 대한 인간의 인지 한계 등을 이용해서 원래의 시간관계와 유사하게 출력되도록 하기 위하여 인위적인 동기화 기능의 개입을 필요로 한다[4,6,9].

동기화 규격 모델은 객체의 일관성과 동기화 규격의 관리를 지원해야 하고 모든 동기화 관계의 형태를 유연하게 서술해야 한다. 또한 다양한 미디어의 통합과 서비스의 품질 요구사항의 정의가 지원

* 숭실대학교 컴퓨터학과(chlee@multi.soongsil.ac.kr)
논문번호 : 00180-0523, 접수일자 : 2000년 5월 23일

되어야 하며 쉽게 서술되어야 한다. 그러나 OCPN (Object Composition Petri Net), RTSM(Real-Time Synchronization Model)과 같은 기존의 확장된 페트리네트 모델은 멀티미디어 실시간 문제와 패킷 및 셀 네트워크의 랜덤 지연이 고려되었을 때 패킷의 늦은 전송을 다루기가 충분하지 않으며, 또한 미디어내 및 미디어간 서비스 품질 파라미터를 모델링에 표현하지 않았다[5,10].

MMSM(Multi Media Synchronization Model) 모델에서 재생되는 미디어의 끊어짐을 방지하기 위해 유연한 재생 기법을 제안한다.

논문의 구성은 다음과 같다. Ⅱ장에서는 기존의 페트리네트에 기반한 동기화 모델에 대하여 서술하고 문제점을 제시한다. Ⅲ장에서는 MMSM 모델의 동기화 기법을 서술하고 Ⅳ장에서는 서비스 품질 평가를 기술한다. Ⅴ장에서는 MMSM의 시뮬레이션 결과를 서술하고, Ⅵ장에서는 결론을 내리고 추후 연구 방향에 대하여 논의한다.

Ⅱ. 기존 멀티미디어 동기화 모델

멀티미디어 응용을 위한 동기화 모델을 서술하는 많은 연구들이 이루어져 왔다[10,11]. 기존의 동기화 모델을 살펴보면 다음과 같다.

2.1 OCPN

OCPN(Object Composition Petri Net)은 Little과 *Ghafoor*에 의해 소개된 이후 최근 멀티미디어 데이터의 모델링에 폭넓게 사용되고 있다[7,8]. OCPN은 미디어 데이터 형태와 프리젠테이션 시간을 모델링하기 위해 플레이스에 자원 및 시간주기를 추가한 페트리 네트의 변형된 형태이다.

OCPN은 플레이스에 시간을 할당함으로써 동기화의 정도(granularity)를 자유롭게 선택할 수 있다. 페트리 네트의 정의와 관련해서 OCPN의 점화규칙은 다음과 같다.

1) 전이 ti는 각 입력 플레이스가 모두 열린(unlocked) 토큰을 포함하면 즉시 점화한다.
2) 점화시에, 전이 ti는 각 입력 플레이스에서 하나의 토큰을 제거하고, 각 출력 플레이스에 하나의 토큰을 추가한다.
3) 토큰을 받았을 때, 플레이스 pj는 주어진 시간구간 τ j 동안은 active상태에 있게 된다. 즉, 이 시간 구간 동안에 토큰은 닫힌(locked)상태에 있게 되는 것이다. 토큰은 플레이스에서 정해진 시간구간 τ j 이 지난 후에, 열린(unlocked)상태가 된다.

OCPN은 실시간 응용에서의 이러한 오디오 객체의 지연은 서비스 품질의 심각한 저하를 초래할 수 있다.

2.2 RTSM

OCPN 모델에서 문제점인 미디어간의 지연에 의한 서비스 품질의 저하를 방지하고, 실시간 응용에서 요구되는 서비스 품질을 보장할 수 있도록 하기 위하여 RTSM(Real Time Synchronization Model)이 제시되었다[5].

RTSM에서는 키 매체(key medium)를 정의한다. 다른 미디어에 비해 상대적으로 중요한 미디어이거나, 지연이나 지터에 민감한 미디어가 키 매체로 선택되며, 이것은 두 개의 원으로 표시한 강제(enforced) 플레이스로 나타내어 진다. 각각의 전이 ti는 강제 플레이스중 어느 하나라도 해당 행동이 끝나게 되면, 다른 미디어의 상태에 상관없이 점화가 발생하게 된다. RTSM의 점화 규칙은 다음과 같다.

전이 ti는 각 입력 플레이스중에서 적어도 하나의 강제 플레이스 ei를 포함한다.

1) 어떤 강제 플레이스 ei에 있는 토큰이 열린(unlocked) 상태가 되었을 때 전이 ti는 다른 입력 플레이스의 상태와 상관없이 점화된다.
2) 점화시에, 전이 ti는 각 입력 플레이스에서 하나의 토큰을 제거하고, 각 출력 플레이스에 하나의 토큰을 추가한다.
3) 점화시에 백트래킹 규칙의 집합은 입력 플레이스로부터 토큰을 제거하기 위한 기능을 수행한다.

그러나 RTSM 모델에서는 실시간 응용에 대한 다양한 동기화 관계를 서술하는데 불충분하다.

2.3 기존 멀티미디어 동기화 기법의 문제점

기존의 멀티미디어 동기화 기법의 문제점을 서술하면 다음과 같다. OCPN 모델의 경우 모든 미디어가 도달해야만 점화되기 때문에 늦게 도착하는 미디어의 영향으로 다른 미디어들이 심각한 손상을 입게 된다. 즉, 서비스 품질의 측면을 전혀 고려치 않은 동기화 기법이다.

기존 동기화 기법의 공통적인 문제점을 요약하면 다음과 같다.

1) 실시간 응용에서의 지연 시간 및 서비스 품질을 고려하지 않은 불완전한 기법이다.

2) 미디어 내 동기화의 QoS 파라미터를 기술하지 않았다.
3) 지연시간의 변화로 인한 데이터 손실을 해결할 수 있는 방법이 제시되지 않았다.
4) 주 미디어에 의존한 동기화 전략이므로 종 미디어의 재생을 고려하지 않았다.
5) 흐름제어 전략을 표현하지 않았다.

본 논문에서는 다양한 네트워크 및 멀티미디어 시스템 환경에서 여러 가지 종류의 멀티미디어 응용에 적합한 동기화 서비스를 제공하기 위해 다음과 같은 유연한 동기화 표현 기법을 제안한다.

미디어의 재생 시 끊어짐을 방지하기 위해 유연한 재생 처리 기법을 제안하였다. 이 기법은 엄격한 디스플레이의 제어를 가질 수 있다. 그러나 자주 뛰어넘은 프레임이 존재한다면 재생시 부자연스럽게 보인다. 사용자 측면에서 그것은 엄격한 재생을 위하여 건너뛰는 프레임을 가지는 것보다는 부드러운 영화를 보기 위해 모든 프레임을 가지는 것을 원할 것이다. 그러므로 미디어의 끊어짐을 방지하고 부드러운 재생을 위해 제안된 유연한 재생 처리 기법을 적용함으로써 종 미디어 스트림이 부드럽게 재생되도록 하였다.

Ⅲ. MMSM 모델의 동기화 기법

제안된 동기화 기법은 스무딩 버퍼에서 미디어의 효율적인 재생 처리를 하는 재생 처리 기법을 제시하였다. 디스플레이 하기 위한 프레임 주기 시간은 자연스러운 재생을 위해 디스플레이 속도를 증가 또는 감소시키기 위하여 스무딩 버퍼 레벨을 검사함으로써 계산된다.

마지막으로 재생 처리 기법이 기존의 모델에서 제시한 동기화 기법보다 우수함을 입증하기 위해 유연한 재생 처리 알고리즘을 제안하였고 서비스 품질을 평가하였다.

주기 상에서 프레임을 디스플레이 하는 것을 재생 처리라 한다. 이 경우 하나의 프레임은 프레임 버퍼가 유용할 때마다 그 시간에 항상 디스플레이 되지는 않는다는 것을 주목해야 한다. 재생 처리의 기존 작업에서는 경계선(deadline)에 도착하지 않은 프레임은 간단히 스킵 된다. 경계선은 다음 프레임이 디스플레이 되기를 기대하는 일정한 시간을 의미한다.

이 방법은 엄격한 재생 처리를 할 수 있다. 그러나 자주 생략된 프레임이 존재한다면 재생 시 부자연스럽게 보인다. 사용자 측면에서 그것은 엄격한 재생을 위하여 생략된 프레임을 가지는 것보다는 부드러운 영화를 보기 위해 모든 프레임을 가지는 것을 원할 것이다.

본 논문은 프레임의 재생 시간이 주기적인 상태를 엄격하게 유지하지 않는 것을 제안한다. 만약 프레임 버퍼의 레벨이 평균보다 작다면 디스플레이 시간(duration)은 더 길어진다. 그리고 더 많은 프레임을 위하여 충분한 시간을 가지는 버퍼를 허용한다. 또한 프레임 버퍼의 레벨이 평균보다 높다면 디스플레이 시간은 짧아진다. 이 기법은 생략되는 프레임들을 보상할 수 있다.

이 기법은 하나의 동기화 시간에 이루어지며 음성이 도착되었을 때 그 음성의 크기를 기준으로 하여 비디오의 재생시간을 재생하는 시스템이 된다.
제안된 방법에 의해 조정된 시간은 다음 식에 의해 계산된다.

$$\tau_{m-f} := \tau_d / P_n; \quad [1]$$

식 [1]에서 P_n은 오디오 객체를 제외한 모든 미디어가 지속시간에 재생할 수 있는 프레임의 수를 의미하며, τ_d는 최대 지속 시간을 의미한다. 그러므로 τ_{m-f} 는 하나의 프레임이 재생할 수 있는 시간을 나타낸다.

$$M_c := \tau_{r-m} / \tau_{m-f}; \quad [2]$$

식 [2]에서 τ_{r-m} 은 각 미디어의 상대 지속시간을 의미한다. 즉, 상대 지속시간에서 하나의 프레임이 재생할 수 있는 시간으로 나누게 되면 실제로 재생할 수 있는 프레임의 수가 실수로 나오게 된다.

$$M_c := Int(M_c); \quad [3]$$

식 [3]은 실수로 나온 실제로 재생할 수 있는 프레임의 수를 정수로 바꾸어 주는 식이다.

$$\tau_p := \tau_{r-mp} / M_c; \quad [4]$$

식 [4]에서 τ_{r-mp} 는 최대 지연 지터 시간을 보상한 오디오의 지속시간을 의미한다. 즉, 오디오의 지속시간을 실제로 재생할 수 있는 프레임 수로 나누면 오디오를 제외한 모든 미디어에 대한 한개의 프레임이 재생할 수 있는 시간이 나오게 된다. τ_p

는 한 개의 비디오 프레임이 재생될 시간을 정의한다. τ_p는 한 개의 비디오 프레임의 재생 시간이 되어 오디오의 재생 시간과 스큐의 차이를 고려해야 한다. 비디오와 오디오의 최대 스큐 허용치는 80ms이다. 이러한 허용치는 본 논문의 동기화 기간이 1초당 8번을 하기 때문에 문제를 발생시키지는 않는다.

$$\tau_{p-t} := \tau_p \times M_c; \quad [5]$$

식 [5]에서는 전체 재생할 수 있는 시간을 나타낸다. 즉, 미디어 한 개의 프레임이 재생할 수 있는 시간에 실제 재생할 수 있는 프레임의 수를 곱하면 전체 재생시간이 나오게 된다.

본 논문에서는 오디오를 키 매체로 정의하여 키 매체의 도착상태를 기준으로 재생시간을 조정하는 기법을 제안한다.

Ⅳ. 서비스 품질 평가

미디어간 동기화에 대한 QoS 파라미터 처리를 위하여 각 플레이스 pi는 키 매체 외에 시간과 관련된 4가지 파라미터인 τ_{diff}, τ_r, τ_d, τ_j을 가지고 있다. τ_d로 표시되는 지속시간은 해당 미디어의 재생 또는 디스플레이가 지속되어야 할 시간 길이, 즉 지속시간을 나타내는 파라미터이다. τ_r은 도착된 미디어를 재생하기 위한 상대적인 지속시간(duration time)으로 계산한 시간을 나타낸다. τ_{diff}는 지속시간과 상대 지속시간의 차이를 의미한다.

τ_j로 표시되는 최대 지연 지터 시간은 각각의 단일 미디어 사이에서 허용되는 지연시간 차이를 나타내는 파라미터이다. 그리고 미디어간의 스큐 시간이 0이라는 것은 두 미디어 스트림이 완전히 동기화가 되어 있는 상태를 의미한다.

본 논문에서 재생 정책에 필요한 매개변수는 [표 1]과 같다. 어플리케이션에서 지연 지터를 피하기 위하여 수신측에 125ms의 스무딩 버퍼를 이용하였다. 스무딩 버퍼로 나타나는 125ms 지연은 비디오폰, 화상회의, 원격강의, 실시간 응용에서 수용될 수 있다.

미디어의 도착된 시간을 스무딩 버퍼 지연시간과 비교하여 스무딩 버퍼 지연시간인 125ms보다 작으면 일찍 도착한 미디어가 된다. 이것은 전체의 재생 데이터가 모두 도착한 경우이기 때문에 최대 지속시간인 125ms를 재생할 수 있는 데이터를 가지게 된다. 그러나 만약 미디어의 도착한 시간을 스무딩 버퍼 지연시간과 비교하여 크다면 늦게 도착한 미디어를 의미한다.

표 1. 재생 정책에 요구되는 매개변수

매개변수	설 명	단위
m	모든 미디어	
P_{ms}	재생할 수 있는 최대 크기	Byte
τ_b	최대 지연 시간	[ms]
τ_{ma}	미디어들의 도착한 시간	[ms]
P_{ma}	스무딩버퍼지연동안에 도착된크기	Byte
P_r	P_{ms}와 P_{ma}의 재생 비율	
τ_r	상대 지속시간	[ms]
τ_{r-a}	오디오의 상대 지속시간 변수	[ms]
τ_{r-m}	오디오를 제외한 미디어의 상대 지속시간 변수	[ms]
τ_{r-mf}	각 미디어들의 재생시간	[ms]
τ_d	최대 지속시간	[ms]
τ_j	최대 지연 지터 시간	[ms]
τ_{diff}	최대 지속시간과 상대 지속시간의 차이	[ms]
P_n	최대 지속시간에 재생되는 각각의 미디어 프레임 수	frame
τ_{m-f}	각 미디어들의 재생되는 하나의 프레임 시간	[ms]
M_c	현재 미디어가 도착된 프레임 수	frame
τ_p	한 개의 비디오 프레임이 재생될 시간	[ms]
τ_{p-t}	미디어의 유연한 재생시간	[ms]

스무딩 버퍼에서 미디어 데이터를 측정하여 상대적인 재생시간을 구할 수 있다. 좀 더 자세히 살펴보면 스무딩 버퍼 지연시간인 125ms에 도착된 데이터의 크기를 계산한다. P_r은 지속시간에 미디어가 재생할 수 있는 최대 크기와 스무딩 버퍼 지연동안에 도착된 크기의 재생 비율을 의미한다. τ_r은 도착된 미디어를 재생하기 위한 상대적 지속시간을 나타낸다.

지속시간 동안 프레임을 디스플레이하는 것을 재생 제어라 한다. 이 경우 하나의 프레임은 프레임 버퍼가 유용할때마다 그 시간에 항상 재생이 되지는 않는 다는 것을 기억해야 한다. 재생 제어의 기존 작업에서는 스무딩 버퍼의 최대지연시간에 도착하지 않은 프레임은 간단히 스킵되었다. 스무딩 버퍼의 최대지연시간은 다음 프레임이 재생되기를 기

대할 때 주어진 주기의 일정한 시간을 의미한다. 그러나 자주 스킵된 프레임이 존재한다면 어떤 것은 부자연스럽게 보이고, 사용자 측면에서는 그것은 엄격한 재생을 위하여 생략된 프레임을 가지는 것보다는 부드러운 영화를 위해 생략되는 프레임을 적게 하는 것을 선호할 것이다.

본 논문에서는 지속 시간 동안에 네트웍 상태의 변화 또는 평균 지연의 변화에 따른 오디오의 손실에 따른 비디오의 손실을 비디오의 재생시간을 변화시킴으로 해서 손실될 수 있는 비디오의 손실을 보상시키고자 한다. 만약 오디오의 경우 90ms를 재생하는 데이터를 가지고 있고 비디오는 125ms를 재생할 수 있는 경우라 할 때, 비디오는 키 미디어인 오디오에 의해서 90ms에 지터 10ms를 더한 100ms만 재생하게 된다. 이것은 비디오 데이터의 재생시간이 25ms 손실되는 것을 의미한다.

그래서 오디오의 재생시간을 비디오 프레임의 개수로 나누게 되면 30ms를 얻게 된다. 이때 최대 지터를 보상하여 100ms를 비디오 프레임 개수로 나누게 되면 33ms를 얻게 된다. 그러므로 100ms를 재생시킴으로써 손실될 수 있는 비디오 프레임을 어느 정도 재생시킬 수 있는 방안이 될 것이다. 이러한 재생 처리 알고리즘은 [그림 1]과 같다.

```
begin
    while media then
        τ_{m-f} := τ_d / P_n;
        M_c := τ_{r-m} / τ_{m-f};
        M_c := Int(M_c);
        τ_p := τ_{r-mp} / M_c;
        τ_{p-t} := τ_p × M_c;
end
```

그림 1. 유연한 재생 처리 알고리즘

부드러운 재생을 위해 오디오가 정상적으로 도착하였을 때를 살펴보면 다음과 같다. 125ms동안에 비디오는 세개의 프레임을 재생할 수 있다. 그러므로 125 / 3 = 41.66ms는 하나의 프레임이 재생할 수 있는 시간이다.

지터를 적용하지 않은 첫 번째 경우와 지연지터를 적용한 두 번째 경우, 그리고 본 논문에서 제안한 유연하게 지속시간을 늘이고 줄인 세 번째 경우 등 세가지 경우를 비교한다.

오디오 객체가 정상적으로 도착하고, 비디오 객체가 늦게 도착한 경우를 살펴보면 [표 2]와 같다.

표 2. 오디오가 정상이고 비디오가 비정상인 경우

	80ms	140ms(2 프레임)
첫번째	125	41.66 X 2 = 83.33
두번째	125	83.33
세번째	125	62.5 X 2 = 125

[표 2]에서 오디오 객체의 지연시간이 80ms이고 비디오 객체의 지연시간이 140ms라고 가정하자. 오디오 객체는 일찍 도착하였기 때문에 125ms의 지속시간을 갖고, 비디오 객체는 늦게 도착하였기 때문에 지속시간은 111ms가 된다. 그러나 [그림 2](a)(b)에서 처럼 실제로 비디오 객체는 두 개의 프레임밖에 재생할 수 없다. 즉, 두 프레임의 재생시간인 83.33ms만 재생하고, 41.66ms 동안은 멈춘다. 비디오는 111 / 41.66 = 2 프레임 즉, 27.68ms는 손실이 된다. 왜냐하면 비디오는 프레임 단위로 재생되기 때문이다.

[그림 2](c)에서 보는바와 같이 실제로는 41.66 X 2 = 83.32ms 밖에 재생할 수 없지만 제안한 기법에 적용하면 비디오는 2 프레임을 재생할 수 있기 때문에 125 / 2 = 62.5ms가 된다. 즉, 하나의 프레임당 62.5ms씩 늘려서 재생하면 125ms가 된다.

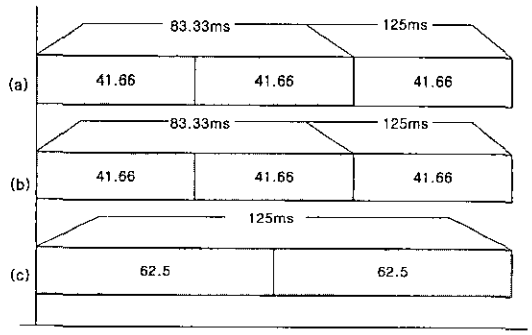

그림 2. 비디오가 비정상적일때의 경우

오디오가 비정상적으로 도착하였을 때의 상태를 살펴보면 다음과 같다.

오디오 객체가 늦게 도착하고 비디오 객체가 정상적으로 도착한 경우는 [표 3]과 같다.

[표 3]에서 오디오 객체의 지연시간이 140ms이고 비디오 객체의 지연시간이 80ms라고 가정하자. 오디오 객체는 늦게 도착하였기 때문에 111ms의 지

표 3. 오디오가 비정상이고 비디오가 정상인 경우

	140ms	80ms(3 프레임)
첫번째	111	111
두번째	121	121
세번째	121	121

속시간을 갖는다. 그리고 비디오 객체는 일찍 도착하였기 때문에 세 개의 프레임을 재생할 수 있다. 하지만 지속시간은 111ms가 된다. [그림 3](a)는 두 개의 프레임은 정상적으로 재생되지만 세 번째 프레임은 27.68ms 밖에 재생할 수가 없다. 그러므로 13.98ms동안 비디오는 멈추게 된다.

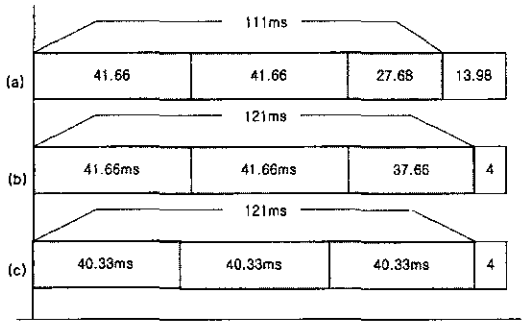

그림 3. 오디오가 비정상적일때의 경우

[그림 3](b)는 지연지터만큼 보상하였기 때문에 121ms가 재생이 된다. 그러나 세 번째 프레임은 37.66ms만 재생이 된다. 즉, 4ms동안 멈추게 된다. 첫 번째 경우와 비교할 때 두 번째 경우는 지연지터 만큼 더 재생되었다.

[그림 3](c)에서 보는바와 같이 제안한 기법에 적용하여 세 개의 프레임을 유연하게 재생하게 한다. 즉, 121ms의 재생시간을 세 개의 프레임에 적용함으로써 부드러운 재생이 될 수 있다.

오디오 객체와 비디오 객체가 늦게 도착한 경우는 [표 4]와 같다.

표 4. 오디오와 비디오가 비정상일 경우

	140ms	140ms(2 프레임)
첫번째	111	83.33
두번째	121	83.33
세번째	121	121

[표 4]에서 오디오 객체의 지연시간이 140ms이고 비디오 객체의 지연시간이 140ms라고 가정하자. 오디오 객체와 비디오 객체는 늦게 도착하였기 때문에 지속시간은 111ms가 된다.

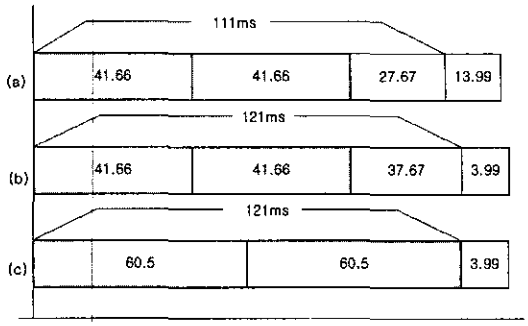

그림 4. 두 미디어가 비정상일때의 경우

그러나 [그림 4](a)의 경우 비디오 객체의 재생시간은 111ms 이지만 실제로는 두 개의 프레임만 재생할 수 있기 때문에 83.33ms만 재생이 되고 27.67ms동안 멈추게 된다. [그림 4](b)는 지연 지터만큼 보상이 되어 121ms동안 재생이 되지만 (a)과 마찬가지로 83.33ms만 재생이 되고 37.67ms동안 멈추게 된다.

[그림 4](c)에서 보는바와 같이 121ms 동안에 두 개의 프레임이 재생되어야 하기 때문에 하나의 프레임을 늘여서 재생함으로써 부드러운 재생효과를 가진다. 즉, 41.66 X 2 = 83.32ms 밖에 재생할 수 없지만 제안한 기법에 적용하면 비디오는 2 프레임을 재생할 수 있기 때문에 121 / 2 = 60.5ms가 된다. 즉, 하나의 프레임당 60.5ms씩 늘려서 재생하면 121ms가 된다.

V. 실험 및 평가

본 논문에서 제안한 기법의 실험을 위한 환경으로는 IBM 호환 기종의 펜티엄 PC를 이용하였으며, 인터페이스 및 알고리즘은 Java 개발 킷 1.2.2로 구현하였고, 마이크로 소프트 MDB에 petrinet.mdb 화일로 저장된다.

1Kbyte 오디오 데이터는 PCM 인코딩 기법에 의해서 인코딩되고 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. 송신측에서 어플리케이션은 125ms마다 오디오 디바이스로부터 오디오 패킷을 얻고

125ms동안에 어플리케이션은 운영체제의 런타임 프로세싱 오버헤드에 의해서 결정된 세 개 이하의 비디오 프레임을 비디오 그래버로부터 얻는다. 성능 측정 실험에서 사용되는 트랜지션 단위체의 개수는 200개이다.

[그림 5]는 제안한 모델의 유연한 재생시간을 적용하였을 경우와 적용되지 않은 경우를 비교한 것이다.

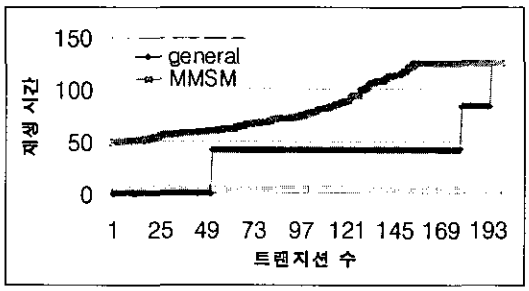

그림 5. 유연한 재생시간 비교 결과

동기화 모델링 기법의 가장 중요한 점은 표현의 유연성과 확장성이라 할 수 있으며, 객체간의 시간 관계의 예상 가능한 경우를 모두 표현할 수 있어야 한다. 제안된 동기화 모델은 재생 처리의 기존 작업에서는 스무딩 버퍼의 최대지연시간에 도착하지 않은 프레임은 간단히 스킵되었다. 스무딩 버퍼의 최대지연시간은 다음 프레임이 재생되기를 기대할 때 주어진 주기의 일정한 시간을 의미한다. 그러나 자주 스킵된 프레임이 존재한다면 어떤 것은 부자연스럽게 보이고, 사용자 측면에서는 그것은 엄격한 재생을 위한 스킵된 프레임을 가지는 것보다는 부드러운 영화를 위해 스킵되는 프레임을 적게 하는 것을 선호할 것이다. 본 논문에서는 주 미디어인 오디오를 기준으로 비디오 데이터를 늘리고 줄임으로써 부드러운 재생을 할 수 있다.

Ⅵ. 결 론

본 논문은 멀티미디어 시스템 및 서비스 제공에 있어 핵심적인 기술로 부각되는 동기화에 대한 동기화 기법을 제시하였다. 본 논문에서는 OCPN이나 RTSM 모델에서의 문제점을 해결하였다. 그러므로 서비스 품질의 향상을 도모할 수 있도록 하였다. 그리고 효율적인 서비스 품질을 제공하는 멀티미디어 동기화 기법을 제안하였다.

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

제안된 미디어내 및 미디어간 동기화 기법은 네트워크 로드의 일시적인 증가에 적합하며 예측할 수 없는 단절에도 적합하다. 또한 실시간 응용에서 주문형 응용에까지 널리 이용할 수 있다.

본 논문에서 제안한 동기화 기법은 다음과 같은 결과를 향상시켰다.

1) 미디어의 끊어짐을 방지하고 부드러운 재생을 위해 제안된 유연한 재생 처리 기법을 적용함으로써 종 미디어 스트림이 부드럽게 재생되는 결과를 나타내었다.
2) 응용할 수 있는 범위는 실시간 시스템에서 주문형 시스템까지 모두 적용이 가능하다.

향후 연구 방향은 사용자와의 상호 작용을 고려한 정형화된 멀티미디어 모델을 만들고 이를 트랜스포트 프로토콜로 구현하여 시뮬레이션 하는 것이다. 그럼으로써 모든 멀티미디어 응용 프로그램에 적용 가능한 동기화 모델을 확립하는 것이다. 또한 최소 버퍼를 이용한 최적의 동기화 기법을 연구해야 할 것이며, 나아가 이동 통신에서의 동기화 모델과 기법을 연구해야 한다.

참 고 문 헌

[1] F. Fluckiger, Understanding Networked Multimedia, Prentice Hall, 1995.

[2] R. Steinmetz and K. Nahrstedt, Multimedia: Computing, Communication and Applications, Prentice Hall, 1995.

[3] G. Blakowski and R. Steinmetz, "A Media Synchronization Survey: Reference Model, Specification, and Case Studies," IEEE Journal on selected Areas in Communications, Vol.14, No.1, Jan. 1996.

[4] 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.

[5] 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.

[6] 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.

[7] T. Murata, "Petri Nets: Properties, Analysis and Applications," The Proc. IEEE, 77-4, 1989.

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

[9] 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.

[10] P. W. Jardetzky, and C. J. Sreenan, and R. M. Needham, "Storage and synchronization for distributed continuous media," Multimedia Systems/Springer-Verlag, 1995.

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

이 기 성(Gi-Sung Lee) 정회원

1993년 2월 : 숭실대학교 전자계산학과 졸업
1996년 8월 : 숭실대학교 컴퓨터학과 공학석사
1996년~현재 : 숭실대학교 컴퓨터학과 박사수료
<주관심 분야> 멀티미디어 통신, 멀티미디어 응용, 무선 이동 통신

오 해 석(Hae-seok Oh) 정회원

1975년 : 서울대학교 공과대학 졸업
1981년 : 서울대학교 대학원 전자계산학 석사
1989년 : 서울대학교 대학원 전자계산학 박사
1997년~1999년 : 숭실대학교 부총장
1982년~현재 : 숭실대학교 정보과학대학 교수
<주관심 분야> 멀티미디어 통신, 멀티미디어 응용, 무선 이동 통신