

버퍼레벨을 이용한 적응형 멀티미디어 동기화 재생 기법

성경상^o, 황민구, 이기성, 이근왕, 오해석

송실대학교 컴퓨터학과

{actofgod, hminkoo, yglee, lkw}@multi.soongsil.ac.kr, oh@computing.soongsil.ac.kr

An Adaptive multimedia Synchronization playback scheme using buffer level

Kyung-Sang Sung^o, Min-Koo Hwang, Gi-Sung Yi, kuen-Wang Lee, Hae-Seok Oh

Dept. of Computer Science, SoongSil University

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.

제안된 논문은 가변적 전송 지연 시간을 흡수하면서 미디어 데이터 간의 동기화를 수행하는 동적 동기화이다. 즉, 최대 지연 지터 시간을 이용한 동기화 구간 조정과 지연시간의 변화에 따른 가변적 대기 시간을 융통성 있게 처리하여 미디어 데이터간의 동기화 요구를 만족시킬 수 있는 기법이다.

본 논문은 동기화 구간 조정을 처리하기 위해 지연 지터를 적용함으로써 트래픽 증가로 인한 미디어 데이터의 손실 시간 및 지연시간의 변화로 인한 데이터 손실을 감소시켰다. 그리고 스무딩 버퍼의 대기 시간을 가변으로 처리함으로써 지연시간의 변화로 인한 불연속을 감소시켰다. 제안된 논문은 고품질 서비스의 보장을 요구하는 시스템에 적합하며, 재생율 증가와 손실을 감소 등 서비스 품질을 향상시켰다.

1. 서 론

멀티미디어 서비스에서 서비스 품질과 관련하여 동기화가 핵심적인 기능으로 요구되는 이유는 ATM(Asynchronous Transfer Mode)망과 같은 고속통신망을 통해 데이터가 수신측에 전달될 때 송신측으로부터 도착시간의 지연이 다르기 때문에 망에서 발생하는 랜덤(random)한 지연이나 송수신 시스템간 클럭(clock)의 불일치 등에 의해 미디어간에 존재하므로 원래의 시간 관계가 파괴되는 원인이 된다. 이와 같은 이유로 시간관계가 훼손된 멀티미디어 데이터에 대해 응용서비스의 요구사항이나 각 미디어의 손실 및 지연에 대한 인간의 인지 한계 등을 이용해서 가능한 한 원래의 시간관계와 유사하게 출력되도록 하기 위하여 인위적인 동기화 기법의 개입을 필요로 한다[1]. 하나의 의미 단위를 이루는 여러 미디어 정보를 전송할 경우, 미디어간의 시간적인 관계가 존재한다. 따라서 수신측에서도 송신측에서의 각 데이터 스트림에 대한 데이터들간의 시간적인 관계가 수신측에서 유지되어야 한다. 동시에 재생하기 위해 동기화가 필요하며 이러한 미디어들 사이의 동기화를 미디어간(inter-media) 동기화라고 한다[1].

멀티미디어 동기화가 멀티미디어 데이터의 표현상의 동기를 말하는 것이기 때문에 기능은 주로 응용의 입장에서 동기를 맞추는 것이 된다[1-4]. 제안된 동기화 기법은 최대 지연 지터 시간을 이용한 동기화 구간 조정과 지연시간 변화로 인한 데이터 손실의 감소 그리고 미디어 내 및 미디어간 동기화를 위한 적용형 동기화 기법을 제안한다.

본 논문의 구성은 다음과 같다. II장에서는 멀티미디어 동기화에 대한 관련 연구를 서술한다. III장에서는 제안하는 동기화 기법을 서술한다. IV장에서는 제안한 멀티미디어 동기화 기법의 시뮬레이션 결과를 서술하고, V장에서는 결론을 내리고 후속 연구 방향에 대하여 논의한다.

2. 기존 멀티미디어 동기화 기법과 지연 지터의 성질

2.1 데이터 전송에 따른 분류

멀티미디어 동기화를 위해 많은 기법이 제안되었다. 이러한 기법을 몇 가지로 분류해 보면 다음과 같다. 첫 번째 방법은 플레이하기 위한 시간

정보를 포함하는 프레임을 전송하는 것이다. 이것은 재생할때 클라이언트측에서 스케줄을 하도록 한다. 두 번째 방법은 송신 주도형 동기화 방법으로 송신측에서 데이터의 전송 시간이나 전송률을 조정함으로써 수신측에 비동기 현상이 발생하지 않도록 하거나 수신측에서 발생한 비동기 현상을 정정하는 방법이다. 실제적인 도착시간과 기대되는 시간을 비교하여 네트워크 로드를 계산함으로써 도착된 프레임을 재생하거나 드롭시키는 것이다. Escobar는 이 기법을 위해 글로벌 클락을 제안하였다[8]. 세 번째 방법은 수신 주도형 동기화 방법으로 수신측에서 방으로 거쳐 도착하는 패킷에 대해 버퍼를 통하여 플레이아웃 시간을 조정하는 방법으로, 여러 미디어 근원지로부터 데이터를 수집하여 저장하거나, 라이브 데이터를 수신하는 데 적합하다. Rothermel, Little의 수신측에서 버퍼를 제공한다. 버퍼의 레벨을 체크함으로써 수신된 프레임은 재생되거나 네트워크 부하와 연관된 것은 폐기시킨다[5]. 마지막 방법은 송신측에서 제어 기법을 제안하였다. Rangan은 송신측이 전송 상태를 체크하여 프레임에 제어하는 피드백 기술을 제안하였다[7]. 수신측은 전송 속도가 재생되는 속도 보다 빠르거나 느릴 때 송신 측에 피드백을 보내게 된다. 송신측은 수신측으로부터 피드백에 관련된 어떤 프레임도 스킵하거나 같은 프레임 을 보낸다. Chakrabarti의 방법은 피드백 기법이 수신지의 버퍼 레벨을 체크함으로써 생긴다는 것을 제외하고는 Rangan의 방법과 같다[9,10]. 이 방법은 압축과 분해에 대한 시간을 고려하지 않는다. 기존연구는 송신측에서 생성된 미디어의 순서와 재생시간을 결정하여 수신측의 버퍼에서 대기하는 시간을 고정시킴으로써 해서 네트워크 지연으로 인한 최대지연 미디어의 재생시간을 감소시키는 단점을 가지고 있다. 이러한 단점은 멀티미디어 객체의 불연속을 해결할 수 있는 방법이 제안하고 있지 않기 때문이다. 본 논문에서는 이러한 단점을 보완한 재생전략을 제안한다.

2.2 지연 지터 기법의 성질

미디어 데이터 스트림을 구성하는 패킷들 간에는 동기화된 프리젠테이션을 위한 동기화 구간이 존재한다. 단위(unit) 시간마다 정상적으로 프리젠테이션 되어야 할 패킷의 개수를 N 이라 할 때, 매 I/N 시간마다 하나의 패킷이 출력되어야 한다. 따라서, 각각의 패킷들은 I/N 마다 동기화 되어 출력되어야 하므로, 이러한 패킷들 간의 출력 간격을 동기화 구간(synchronization interval)이라 한다. 따라서 각 패킷의 동기화 구간의 크기를 T_i 라고 할 때, i 번째 패킷이 T_i 라는 시간에 출력되었으면 $i+1$ 번째 패킷은 $T_i + T_i$ 시간에 출력되어야 미디어간 동기화가 이루어질 것이다. 그러나 연속 미디어 데이터는 그

종류에 따라 어느 정도의 불연속을 허용한다.

[성질 1] 음성 미디어는 최대 지연 지터 시간이 10ms 이하이면 서비스 품질에 영향을 주지 않고 일시적인 불연속을 허용한다.

미디어 데이터의 불연속 허용 특성을 동기화 구간에 적용하면 정상적 동기화 구간보다 더 크게 확장할 수 있어 네트워크 지연 시간 변동으로 인하여 늦게 도착한 패킷들도 프리젠테이션 할 수 있으므로 지연 시간 변동을 어느 정도 흡수할 수 있다. 프리젠테이션 장치로 출력되어야 할 단위 시간당 패킷의 개수를 N 이라고 하면, 단위 시간당 각 패킷들이 $1/N$ 시간 간격으로 출력되어야 하므로, 해당 미디어 데이터 스트림을 구성하는 패킷들 간의 정상적 동기화 구간 Δ 는 $1/N$ 이 된다.

[성질 2] 불연속 허용률을 δ 라고 하면 확장된 동기화 구간 Δ' = $\Delta + \delta$ 이 된다.

미디어 데이터 스트림 j 에 대한 j 번째 패킷이 M_{ij} 라는 시간에 프리젠테이션 되었다면 $j+1$ 번째 패킷의 프리젠테이션 시점은 $M_{ij+1} = M_{ij} + \Delta'$ 가 되어야 미디어 내 동기화를 달성할 수 있다.

[성질 3] $j+1$ 번째 패킷의 버퍼 내 도착 시점을 B_{ij+1} 라고 하면 B_{ij+1} 이 M_{ij+1} 보다 작거나 같으면 동기화 조건을 만족한다.

미디어 내의 동기화를 위해서 j 번째 패킷과 $j+1$ 번째 패킷은 동기화 구간 내에서 다음의 [식 1]을 만족해야 한다.

$$\begin{aligned}
 B_{ij} &\leq M_{ij+1} \\
 &\leq M_{ij} + \Delta' \\
 &\leq M_{ij} + \Delta + \delta \\
 &\leq M_{ij} + 1/N + \delta \dots\dots\dots \text{[식 1]}
 \end{aligned}$$

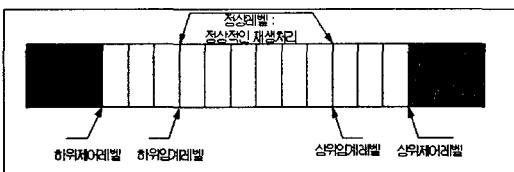
위의 동기화 조건을 기반으로 한 미디어 내 동기화 기법을 제안한다. 네트워크 지연 시간 변동으로 인하여 설정된 초기 지연 시간 보다 더 늦은 지연 시간을 가지고 도착한 패킷으로 인하여 동기화 조건을 항상 만족시킬 수 없게 된다. 따라서, 이러한 경우에는 패킷의 처리 속도 즉, 프리젠테이션 속도를 해당 미디어 데이터에서 허용하는 한도에서 일시적으로 증가시키거나 동기화 조건을 만족할 수 없는 패킷을 스킵 함으로써 서비스 품질에는 영향을 미치지 않으면서 미디어 내 동기화를 수행하여야 한다.

일정한 기간 동안 전송 시간 스케줄링 작업이 반복되면 네트워크의 지연 시간이 계속 상승된다는 것을 의미한다. 이것은 지연 시간 재 설정만으로 지연 시간을 흡수할 수 없다는 것을 의미하므로 프리젠테이션 장치의 출력 속도를 일시적으로 감소시켜 네트워크의 지연 시간을 흡수하여 미디어 내 동기화를 이루어야 하는데, 이런 경우 동기화 구간은 정상보다 더욱 확장된다. 그러나 이 때에도 동기화 구간의 확장이 응용에서 허용하는 범위를 벗어 나서는 안 된다.

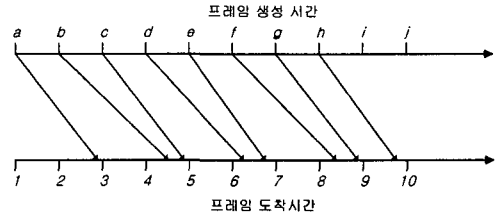
[성질 4] 프리젠테이션 장치의 출력 속도가 변화되면 정상적인 동기화 구간의 크기도 변화된다.

확장된 동기화 구간에 의해 동기화 된 패킷의 수가 많으면 원래의 정상적인 동기화 구간과의 스류가 발생한다. 따라서, 이것은 정상적인 프리젠테이션과 동기화 된 프리젠테이션 사이에 스류가 발생한 것이므로 다시 정상적인 프리젠테이션 시간과 동기화를 맞추어야 한다.

수신지에서는 지연되어 도착하는 프레임들이 일정하게 지연시간을 갖는다면 문제가 되지 않겠지만 대부분 불규칙한 지연시간을 갖게 된다.

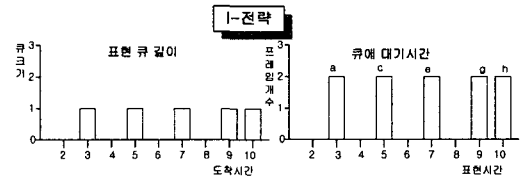


[그림-5] 버퍼 점유율을 이용한 재생처리 전략



[그림 1] 지연지터의 예

[그림 1]은 멀티미디어 데이터의 지연 지터를 나타낸다. 이러한 지연 지터는 불연속의 원인이 된다. 대부분의 회의 시스템에서는 불연속이 너무 자주 일어나지 않는 한, 적은 지연시간으로서 적은 불연속을 갖고서 표현하는 것이 많은 지연시간으로서 불연속 없이 표현하는 것보다 좋은 품질을 나타낸다. 뿐만 아니라, 약간의 불연속을 기꺼이 허용한다면 최악의 end-to-end 지연보다 더 적은 지연으로서 표현하는 것을 선택할 수도 있다. 여기에서 그 프레임은 버릴 것 인지의 I-전략과 표현할 것 인지의 E-전략 등 두 가지의 전략이 있다[5].

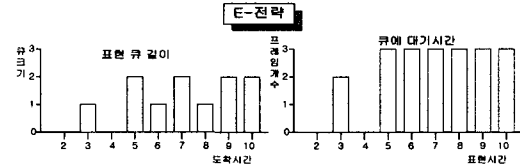


[그림 2] 트래픽이 적을때의 I-전략

[성질 5] I-전략은 논리시간보다 더 큰 end-to-end 지연을 올 갖는 프레임은 버린다.

$$B_{ij} \leq M_{ij} + \Delta \dots\dots\dots \text{[식 2]}$$

I-전략은 프리젠테이션 시점 이전에 패킷의 버퍼내 도착 시점이 이루어져야 한다. 그러므로 [성질 3] 조건에 만족하지만 불연속 허용률 δ 를 고려하지 않았으므로 [성질 2]에 만족하지 못하는 전략이다.



[그림 3] 트래픽이 적을때의 E-전략

[성질 6] E-전략은 늦은 프레임은 버퍼에 대기한 후 다음시점에 표현된다.

$$B_{ij} \leq 2(M_{ij} + \Delta + \delta) \dots\dots\dots \text{[식 3]}$$

이 전략은 하나의 늦은 프레임 이후의 모든 프레임의 지연시간이 증가되는 효과를 갖는다.

[그림 2]와 [그림 3]은 [그림 1]의 I-전략과 E-전략의 방법을 보이고 있다. I-전략의 잠재적 지연 파라미터는 2 프레임 단위이다. [그림 2]와 [그림 3]의 표현 큐 길이는 각 시작 표현시점부터 버퍼 큐의 길이를 보이고 있고, 큐에 대기시간은 각 시작 표현시점에서 표현될 프레임의 지연을 나타낸다. 각 막대기는 시작 표현시점에 표현되는 프레임의 획득시간과 함께 표시되어 있다. 예를 들어, b, d, f는 2 프레임 단위보다 더 긴 end-to-end 지연시간에 도착해서 버려진다. 그래서 I-전략은 시작 표현시점 4, 6, 8에서 표현시에 3 번의 불연속을 포함하게 된다.

[그림 3]의 E-전략은 b 프레임이 단위 4에서 불연속을 발생시키지만, b는 버퍼 큐에 놓이게 되고 결국 3 프레임 단위의 지연을 갖는 단위 5에서 표현된다. 프레임 b 이후에 어떤 프레임도 end-to-end 지연이 3 프레임 단위보다 크지 않으므로 이후에는 불연속이 생기지 않는다. 결국, E-전략이 좋다는 결론이 나온다. E-

전략은 가장 작은 가능한 초기 지연으로서 프레임용 표현하기 시작하여 지연지터를 이용해 지연시간을 길게 조정한다.

E-전략의 결과는 아직 나타나지 않은 어떤 end-to-end 지연보다 더 크게될 지연시간을 동적으로 조정하므로써 공간 없이 프레임용 표현하기에 충분한 지연시간을 발견하는 것이다. E-전략은 I-전략보다 늦게 표현되는 단점은 있지만 많은 프레임용 표현할 수 있을 때는 적합하다.

3. 제안된 동기화 기법

주기상에서 프레임용 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 기대할 때 주어진 주기의 일정한 시간을 의미한다.

[그림-5]에서 보는 바와 같이 버퍼가 정상레벨을 유지할 경우에는 미디어의 재생처리 구간을 정확하게 지켜나가게 된다. 만약 1초에 30프레임용 요구하는 시스템일 경우는 33.3ms의 시간으로 재생을 하게 된다. 그러나 예상 못했던 네트워크의 지터에 의해서 미디어의 도착시간이 늦어진다면 버퍼에 미디어 프레임용 많이 확보하고자 할 것이다. 이러한 문제점을 해결하기 위해서는 두 가지 방법이 있다. 첫 번째 방법은 네트워크에서의 흐름제어 및 데이터의 량을 변환시킴으로써 문제를 해결하는 방법이다. 버퍼의 점유율을 가지고 재생 시점을 변화시키는 방법이다. 이러한 방법은 앞 절의 지연 지터 기법의 성질에서 설명한 바와 같이 최대 지연지터인 10ms 이하에서 동기화 시점을 조정할 수 있다는 아이디어를 본 논문에 적용하게 되었다.

버퍼수위가 상위임계레벨에 존재할 때에는 버퍼가 오버플로우 상태가 될 가능성이 높게된다. 버퍼는 미디어가 도착되는 시점이 입력이 되고 재생시점이 출력이 된다. 본 논문은 재생시간을 짧게 조절함으로써 출력의 시간을 빠르게 한다. 즉, 버퍼레벨의 크기에서 현재 버퍼가 가리키고 있는 수위를 마이너스 한 값을 버퍼레벨의 크기로 나누게 되면 재생시간을 위한 가중치를 얻을 수 있다. 변경된 재생시간은 정상레벨에서 사용되는 재생시간에 최대지연지터인 DJ^{max} 을 더한 값에 가중치를 곱하면 새로운 재생시간을 얻을 수 있다. 재생시간은 가중치를 최대지연지터와 곱함으로써 재생시간을 늘려주게 되며 동기화 시점을 변경하는 결과를 가져오게 되는 효과를 갖게 된다. 이것은 언더플로우를 사전에 방지하는 효과를 가지고 있으며 이러한 효과는 상위제어레벨로 버퍼의 구간으로 떨어지는 것은 늦출 수 있는 효과를 갖게 된다.

버퍼의 수위가 하위임계레벨을 가리키고 있을 경우에는 버퍼가 기아 상태로 이동할 확률이 높다. 이 경우 버퍼의 수위를 정상상태로 이동하는 방법은 두 가지가 있다. 프레임이 오버플로우 상태의 프레임용 예상하여 스킵하는 방법과 재생시점을 늘려주는 방법이다. 본 논문에서는 후자의 방법을 기술하게 된다. 버퍼의 전체 크기인 $LEVEL_all_size$ 은 하위임계레벨에 속하는 버퍼의 개수를 갖게되고 $LEVEL_present_size$ 는 상위임계레벨에 차지하고 있는 현재의 개수를 갖게 된다. 버퍼의 수위가 상위임계레벨에 존재한다면 현재의 버퍼 수위를 전체의 버퍼크기로 나누면 가중치를 얻을 수 있다. 이것을 정상레벨에서의 재생시간과 가중치를 곱하여 정상레벨의 재생시간과 더하면 새로운 재생시간을 얻게 된다. 이러한 기술을 멀티미디어 통신에 적용함으로써 예상하지 않은 지터에 의한 버퍼의 오버플로우에 의한 도착되는 미디어를 버리는 경우를 어느 정도 줄일 수 있다.

버퍼의 정상레벨의 알고리즘은 다음과 같다.

```

 $\tau_{expect} = 33$ 

 $\tau_{expect} = 33$ 
IF buffer_point = 2 THEN
 $\tau_t = \tau_{expect}$ ;
END IF
    
```

버퍼의 상위임계레벨의 알고리즘은 다음과 같다.

```

LEVEL_all_size =  $B_{tot}^M$ 
LEVEL_present_size
 $\omega = 0$ ;
 $\tau_{expect} = 33$ 
 $DJ_m^{max} = 10$ 
IF buffer_point = 1 THEN
BEGIN
 $\omega' = LEVEL\_all\_size - LEVEL\_present\_size$ 
 $\omega = \omega' / LEVEL\_all\_size$ 
 $\tau_t = \tau_{expect} + DJ_m^{max} \times \omega$ 
END
END IF
    
```

버퍼의 하위임계레벨의 알고리즘은 다음과 같다.

```

LEVEL_all_size =  $B_{tot}^M$ ;
LEVEL_present_size;
 $\omega = 0$ ;
 $\tau_{expect} = 33$ ;
IF buffer_point = 3 THEN
BEGIN
 $\omega = LEVEL\_present\_size / LEVEL\_all\_size$ ;
 $\tau_t = \tau_{expect} + (\tau_{expect} \times \omega)$ 
END
END IF
    
```

4. 시뮬레이션 결과

4.1 실험 환경

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 파일로 저장된다.

1Kbyte 오디오 데이터는 PCM 인코딩 기법에 의해서 인코딩되고 비디오 프레임의 해상도는 120 X 120을 사용했다. 초당 24프레임의 인코딩 작업을 하여 사용되어진 프레임이 된다. 송신측에서 어플리케이션은 125ms마다 오디오 디바이스로부터 오디오 패킷을 얻고 125ms동안에 어플리케이션은 운영체제의 런타임 프로세싱 오버헤드에 의해서 결정된 세 개 이하의 비디오 프레임용 비디오 그레버로부터 얻는다.

본 논문은 시뮬레이션 환경을 Ethernet상의 WAN 환경으로 가정하였다. 각각의 패킷에 대한 적절한 작업을 수행하기 위해서 실제 시뮬레이션에 사용된 정보는 포하송분포로 산출하여 네트워크 지연 시간을 두 가지 미디어에 똑같이 적용하였다.

4.2 실험 결과

본 절에서는 본 논문에서 제안하고 있는 최대 지연 지터 시간을 이용한 버퍼전략 및 피드백기법과 재생 기법의 재생시간과 손실 시간을 기존의 피드백기법과 비교 분석한다. 먼저 본 논문에서 버퍼전략적인 측면에서 버퍼는 언더플로우나 오버플로우를 극복하는데 타 방법과의 비교를 통해 본 논문에서 제안한 방법의 우수성을 보였다. 각각의 미디어가 주미디어가 정상적으로 도착하였을 경우, 평균 지연은 100ms이고 편차는 20ms라고 가정할 후 실험하였다.

[표 1]은 피드백에 의한 버퍼 점유율정책을 적용한 경우와 재생 처리정책 만 적용되었을 경우와 피드백과 재생정책을 모두 사용하여 적용하였을 경우를 비교 평가한 결과이다. 본 논문에서 제안한 피드백과 재생처리를 모두 적용했을때, 제안된 전략의 재생율을 10번의 실험을 통해 얻어진 결과이다. 제안된 전략이 약 8% 이상의 재생율을 향상시켰다.

[표 1] 제안 기법의 재생율 비교
단위(%)

실험횟수	일반버퍼전략	제안전략
1	79.06	86.86
2	79.98	87.75
3	80.11	87.92
4	77.76	85.65
5	80.34	88.20
6	79.80	87.55
7	80.46	88.19
8	80.88	88.70
9	80.00	87.75
10	80.14	87.97
평균	79.85	87.65

Communications, Vol. 8, No.3, Apr. 1990.

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

[4] E. Biersack, W. Geyer, and C. Bernhardt, "Intra- and Inter-Stream Synchronization for Stored Multimedia Streams," *IEEE Proc. of Multimedia'96*, pp. 372-381, 1996.

[5] D. L. Stone, and K. Jeffay, "An empirical study of delay jitter management policies," *Multimedia Systems/Springer-Verlag*, 1995.

[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] 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.

[8] J. Escobar, C. Patridge, and D. Deutsch, "Flow Synchronization Protocol," *IEEE/ACM Transaction on Networking*, vol 2, no.2, pp. 111-121, Apr. 1994.

[9] P.V.Rangan, S.Ramanathan, et al., "Techniques for Multimedia Synchronization in Network File Systems," *Computer Communications*, Vol.16, No.3, Mar, 1993.

[10] S.Chakrabarti and R.Wang, "Adaptive Control for Packet Video," *Proc. of Int. Conference on Multimedia Computing*, pp. 56-62, May 1994.

5. 결론

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

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) 동기화 구간 조정을 처리하기 위해 지연 지터를 적용함으로써 트래픽 증가로 인한 미디어 데이터의 손실 시간 및 지연 시간의 변화로 인한 데이터 손실을 감소시켰다.
- 2) 스무딩 버퍼의 대기 시간을 가변으로 처리함으로써 지연시간의 변화로 인한 불연속을 감소시킬 수 있었다.
- 3) 응용할 수 있는 범위는 실시간 시스템에서 주문형 시스템까지 모두 적용이 가능하다.

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

참고문헌

[1] 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.

[2] R. Steinmetz, "Synchronization Properties in Multimedia Systems," *IEEE Journal on selected Areas in*