

상호대화형 객체 삽입과 가변 버퍼 정책을 이용한 스트림 동기화 기법의 성능 평가

이병문^o, 이양민, 이재기
동아대학교 컴퓨터공학과

Performance Estimation of Stream Synchronization Mechanism using Insertion Interactive Object and Variable Buffer

Byung Moon Lee^o, Yang Min Lee, Jae Kee Lee
Dept. of Computer Engineering, Dong-A Univ.

요 약

현재 시간성에 민감한 서비스가 인터넷상에서 중요한 위치를 차지하고 있다. 시간성이 민감한 서비스는 사용자 상호대화를 가능케 하는 미디어 전송 방법 및 연속적 재생을 보장하기 위한 버퍼 정책이 요구된다. 관련 연구에서는 여러 방법을 통해 동기화를 달성하고 있으나 상호대화라는 측면에서는 만족할 만한 해결책을 제시하지 못하고 있다. 본 논문에서는 상호대화형 객체(Interactive Object)를 각 미디어 파일에 삽입하고 객체들이 서로의 정보를 이용할 수 있는 함수를 설계하여 실시간에 원하는 미디어 프레임의 재생위치를 찾아냄으로써 동기화와 상호대화성이라는 문제를 해결하였다. 또한 네트워크에 대한 의존성 때문에 발생하는 불연속적인 재생은 크기를 변화시킬 수 있는 가변 버퍼를 이용함으로써 해결하였다. 그리고 두 가지 방법을 적용한 기법의 우수성을 시뮬레이션 실험을 통하여 확인하였다.

1. 서 론

네트워크 환경의 고속화에 의해 인터넷을 기반으로 하는 다양한 응용 서비스들이 학술, 군사, 의료, 상업 등의 목적으로 등장하고 있으며 사용자의 요구도 매우 다양해지고 있다. 이러한 서비스의 대표적인 예로는 실시간 인터넷 방송, 화상회의, 원격강의 등이 있으며 또한 가상 공동 협업 시스템과 같은 인터넷 기반의 종합적인 컴퓨팅 시스템도 개발 되어 있다. 언급한 응용들 중 실시간 방송과 같은 서비스를 보면 지금까지의 기법들은 단일 파일 내에 비디오, 오디오, 기타 미디어들을 묶어서 전송함으로써 실시간에 하나의 미디어만 조작해내야 하는 응용에는 한계를 가진다. 요즘의 미디어 서비스는 각 미디어를 따로 분리한 후 제공하는 추세이며 미디어 파일을 분리해서 제공할 경우 이 미디어들 간의 동기화가 핵심 문제로 부각 된다.^[1]

본 논문에서는 상호대화성과 동기화 문제를 해결하는 방법으로 미디어 간의 시간 명세 기능이 뛰어난 패트리 넷(Petri Net)을 기반^{[2][3]}으로 하고 정밀한 동기화를 이루기 위해 상호대화형 객체라는 개념을 도입하여 스트림에 삽입함으로써 두 가지 문제점을 해결하고자 한다. 또한 전송된 스트림들을 저장할 버퍼를 가변적으로 운영함으로써 불연속적인 재생을 최소화 하고자 한다.

본 논문에서는 2장에서 기존의 동기화 모델에 대한 고찰과 더불어 구성하고자 하는 시스템의 개요에 관해 설명하고 3장에서 시스템 내부의 동작 원리와 가변 버퍼의 역할에 대해서 설명한다. 그리고 4장에서 제안한 시스템의 효과 및 장점을 설명하고 향후 과제를 제시함으로써 결론을 맺는다.

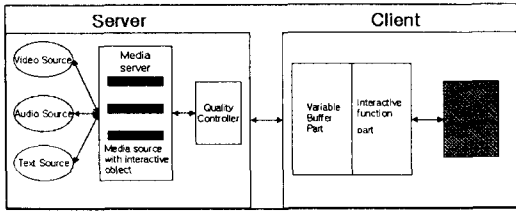
2. 가변형 버퍼를 적용한 상호대화형 패트리 넷

2.1 기존의 동기화 모델

미디어 동기화 기법들을 살펴보면 크게 두 가지 범주로 나눌 수 있다. 하나는 Allen이 제시한 13개의 시간 관계성^[2]을 만족하게 하는 방법이며 다른 하나는 송, 수신측의 버퍼 정책^{[5][6]}을 이용하는 방법이다. 시간 관계성을 명세하는 모델로는 OCPN(Object Composition Petri Net)^[3], XOCN(extended OCPN)^[4] 등의 모델이 있다. 수신측의 버퍼를 이용하는 방법으로는 버퍼 정책을^{[5][6]} 이용하여 버퍼를 유동적으로 조정함으로써 네트워크의 전송 지연으로 인한 미디어 간의 불일치를 적절히 보완하는 기법이 존재한다. 이러한 연구들의 문제점은 사용자와 미디어 스트림간에 발생하는 상호작용에 대한 처리 방법을 가지고 있지 못하다는 것이다. 본 논문에서는 기존에 서술했던 "스트림 전송을 위한 패트리 넷 기반의 상호대화형 동기화 기법"^{[1][7]}에 사용되었던 상호대화형 객체를 바로 적용하고 버퍼레벨^[5]의 개념을 이용하여 네트워크 상태에 따라 적절히 버퍼의 크기를 조정함으로써 연속적인 미디어 스트림을 보장할 수 있도록 한다.

2.2 전체 시스템 개요

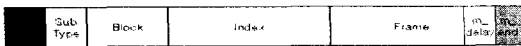
제안하는 시스템은 아래의 (그림 1)에 나타나 있다. 서버에는 분리된 미디어 스트림을 저장하고 있는 미디어 데이터베이스와 미디어 스트림에 상호대화형 객체를 삽입하는 미디어 서버로 구성되며 클라이언트에는 서버로부터 전송되는 상호대화형 객체가 삽입된 미디어 스트림을 원래의 미디어와 상호대화형 객체로 분리하고 이 객체의 정보를 해석하는 상호대화형 함수 부분과 네트워크 상태



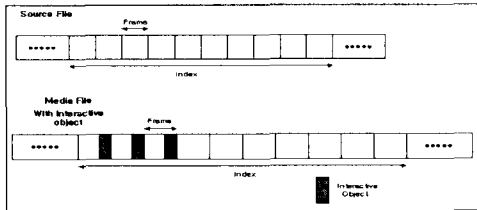
(그림 1) 시스템의 개요

2.3 상호대화형 함수(Interactive function)의 기능

정밀한 동기화를 달성하기 위해 본 논문에서 제안하는 핵심 기능이다. 상호대화형 함수는 크게 4개의 함수로 나눌 수 있으며 각 함수의 이름들은 매핑, 해쉬, 동기화, 내부통신 함수이다. 본 논문에서는 상호대화 및 동기화를 달성 하기 위해 기존 연구의 아이디어였던 상호대화형 객체(Interactive object) [1][7]를 그대로 사용한다. 이 객체는 상호대화와 동기화에 필요한 정보를 포함하는 객체로 아래 (그림 2)와 같은 데이터 포맷을 가지며 미디어 지연값(m_delay)필드가 새롭게 추가 되었다. 미디어 지연값을 이용하여 네트워크 상태에 따라서 지연이 심하게 되는 미디어 스트림의 패킷을 드롭할 수도 있고 버퍼에 저장할 수도 있으며 또한 버퍼의 크기 변화를 제어할 수 있다.



(그림 2) 상호대화형 객체의 포맷



(그림 3) 객체를 삽입한 미디어 스트림

아래에 각 세부 함수와 그 기능에 대해 간략히 설명하였다.

- 해쉬 함수(Hash function): 주 미디어 스트림 내에서 원하는 특정 프레임 위치를 찾아주는 함수이며 상호대화형 함수들 중 제일 처음으로 발동된다.
- 매핑 함수(Mapping function): 해쉬 함수의 동작 후 결과를 이용하여 주 미디어 외의 다른 상대 미디어들 내의 적절한 위치를 찾아주는 것이 매핑 함수의 역할이다.
- 동기화 함수(Synchronization function) : 네트워크 로드의 증가로 인해 필요한 미디어의 전송이 장시간 지연될 경우 미디어 지연값을 이용하여 이것을 감지하고 미디어의 품질을 감소 시키면서 버퍼를 변화시키는 변형 함수에 미디어가 지연되고 있음을 알리는 역할과 예측할 수 없는 오류로 인해 미디어간의 불일치가 한계값(skew)을 넘어설 경우 이것을 적절히 처리할 수 있는 역할을 담당한다.
- 내부통신 함수(Inter-communication function) : 이 함

에 따라 그 크기가 변하는 버퍼 부분으로 구성된다. 수는 상호대화형 객체 사이에서의 통신을 담당하며 미디어 스트림 간의 프레임 위치 정보를 서로 비교하여 스쿠값을 계산하고 객체들의 값이 변경되어야 할 경우 객체 내부의 값을 수정하는데 사용된다.

- Quality Controller : 동기화 함수의 동작 결과로 미디어의 품질을 증가, 감소시켜야 할 경우 사용되는 함수이다.
- 가변 버퍼 변형 함수 (Variable Buffer Variation function) : 미디어 패킷 지연 전송시에 무계획적인 대기 또는 드롭을 막기 위해 미디어 지연값을 이용하여 재생 되는 시점이 조정될 수 있도록 버퍼의 크기를 유동적으로 변화시키는 함수로서 버퍼를 확장, 축소 시키는 프로시저와 프레젠테이션 속도 제어 프로시저를 가지고 있다.

버퍼 확장 프로시저
 $B = Q * T$
 Q : 미디어 스트림의 패킷 크기
 T : 시간 B : 버퍼의 증가분
 (단 $T < 200ms$, Q 즉 패킷의 크기는 미디어의 종류나 응용에 따라 다름)

(그림 4) 버퍼 확장 프로시저

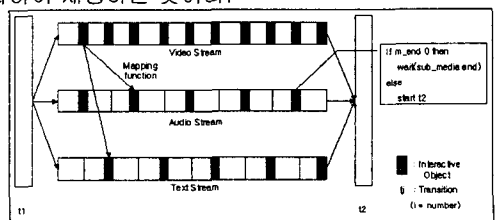
프레젠테이션 속도 제어 프로시저
 $Vp = skew * (1 / (B_{tot} - S_c))$
 Vp : 프레젠테이션 속도 (presentation speed)
 B_{tot} : 버퍼 전체 크기 (Total buffer size)
 S_c : 현재 스트림 양(current stream size)

(그림 5) 프레젠테이션 속도 제어 프로시저

3. 시스템 동작 순서 및 함수들의 동작

3.1 상호대화형 객체와 시스템의 동작

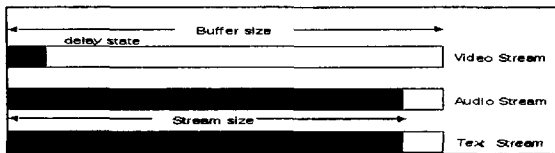
서버측에서 서비스할 미디어 품질을 결정하고 상호대화형 객체를 삽입하여 전송할 미디어를 준비한다. 만약 클라이언트에서 상호대화 동작을 발생하는 이벤트가 일어나면 클라이언트가 미디어 스트림을 해석하면서 가지고 있던 상호대화형 객체의 정보를 이용하여 서버측으로 요청 정보를 요구하게 된다. 클라이언트는 요청 정보를 전송하기 위해서 해쉬 함수를 발동시키고 탐색 버튼의 위치에 알맞은 상호대화형 객체를 찾아서 서버로 전송한다. 서버는 클라이언트측의 요청 정보를 수신하고 이 정보에 따라서 해쉬 함수와 매핑 함수를 발동시켜 주 미디어와 상대 미디어들의 프레임 위치를 결정하게 된다. 이 위치 정보에서부터 미디어 스트림들이 클라이언트로 전송된다. 마지막 동작은 이렇게 전송되어 오는 스트림들을 동기화하여 재생하는 것이다.



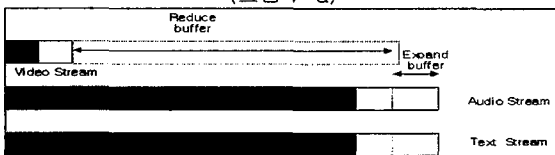
(그림 6) 각 함수들의 동작

3.2 가변형 버퍼의 동작

가변형 버퍼는 제안된 시스템의 클라이언트측에 포함된다. 미디어 지연값을 이용하여 변형 함수가 버퍼의 확장 또는 축소를 계산하게 되고 이것에 의해서 실제로 각 미디어의 버퍼 크기가 유동적으로 변화된다. 버퍼의 크기를 유동적으로 변경하게 되면 미디어 스트림의 플레이 아웃(play out)시간을 의도한 대로 조정할 수가 있다. 만약 네트워크의 문제로 인해 한 미디어의 패킷이 늦게 전송되면 다른 미디어들은 이 미디어 패킷의 도착을 기다리게 되고 이 때 버퍼에 쌓이게 되는 패킷이 늘어나게 된다. 이러한 현상이 발생한 후 시간이 지나면 버퍼 오버플로우가 발생한다. 일반적으로 비교적 용량이 적은 오디오는 충분히 확보가 되었으나 그 시점에서 동기화 되어야 할 비디오 스트림이 충분히 확보되지 못하여 오디오는 플레이 아웃 상태가 될 가능성이 있고 비디오는 아직 그렇게 되지 못한 경우에 일어난다. 이러한 상태의 감지는 상호대화형 객체의 프레임 정보를 이용하여 할 수 있다. 문제의 해결을 위해 가변형 버퍼를 사용한다. 가변형 버퍼의 기본적인 목적은 크기를 유동적으로 변경함으로써 미디어의 플레이 아웃 시간을 의도적으로 조정하는 것이다. 즉 버퍼 오버플로우가 일어날 경우 패킷들이 드롭되는 현상이 발생하므로 스큐값 200ms이내에서 버퍼를 최대한 확장하는 것이다. 버퍼의 최대 임계치와 최소 임계치는 구성 시스템의 사양에 따라 차이가 있다. 그 후에 200ms 이내의 스큐값에서는 사용자가 미디어간의 불일치를 인식할 수 없으므로 이 시간 동안 버퍼를 최대한 확장하였다가 이 시간이 지나면 다른 미디어 버퍼에 있는 스트림들은 강제로 재생하여 버퍼를 비운다. 이 때 지연되고 있는 미디어의 패킷들은 드롭을 하게 된다. 또한 이 시점에서 미디어의 프레젠테이션 시간을 미디어간 스큐값에 의존하여 계산함으로써 오버플로우가 될 상태에 있는 다른 미디어들의 프레젠테이션 시간을 짧게 하여 최대로 불연속을 느낄 수 없도록 재생을 한다. 비디오의 버퍼는 기다리던 패킷을 드롭하고 현재 버퍼를 최소화하여 패킷이 도착하는 즉시 플레이 아웃 할 수 있는 상태로 만들어 준다. 즉 늦게 재생될 미디어의 재생 속도를 일시적으로 증가시켜 네트워크 지연에 의한 미디어 간의 불일치를 최대한 흡수한다. 아래 (그림 7-a)와 (그림 7-b)는 가변 버퍼의 동작을 보이고 있다.



(그림 7-a)



(그림 7-b)

4. 실험 결과

본 연구에서는 시뮬레이터를 통해 일반적인 스트림 전송과 가변형 버퍼를 적용한 스트림 전송을 비교하였다. 시뮬레이터는 Visual C를 이용하여 구현하였고 실험은 하나의 컴퓨터에서 로컬하게 전송하여 실제 네트워크 상의 지연은 무시하였다. 5회에 걸친 실험 결과는 아래와 같다. 이전 연구의 상호대화형 객체를 삽입하고 본 연구에서 제안한 가변 버퍼를 이용하였을 경우가 상호대화형 객체를 적용한 경우보다 최소 4.5% 패킷 재생율이 높으며 최대 10.39% 재생율이 높은 것을 실험을 통하여 확인할 수 있다.

<표1> 전송 방법에 따른 패킷 재생율

전략	일반 전송 방법	가변 버퍼를 적용한 전송 방법
1회	64.06 %	70.01 %
2회	67.23 %	72.87 %
3회	65.57 %	77.12 %
4회	68.01 %	72.53 %
5회	64.74 %	74.45 %

5. 결론 및 향후 과제

본 논문에서 제안된 기법은 첫째 스큐값을 20ms 이내^[1]로 할 수 있어 기존의 기법들 보다 우수하며, 둘째 재생율에 있어서도 가변 버퍼를 적용하지 않은 기법보다 4.5%~10.4% 정도 우수하다는 것을 알 수 있다. 시뮬레이터를 이용한 실험에서 예상했던 것과 같이 기존의 기법들과 비교해 부가적인 처리 시간이 소요되지만 상호대화형 객체를 삽입함으로써 보다 높은 수준의 동기화를 달성할 수 있다. 또한 가변형 버퍼 정책을 적용함으로써 드롭되는 패킷들을 최대한 흡수하여 보다 원활한 재생을 유지할 수 있다는 점이 확인 되었다.

향후 과제는 제안 기법에 사용된 함수들의 알고리즘을 보다 최적화하고 실제 시스템에 적용시키는 것이다.

[참고문헌]

[1] 이양민, 이재기, " 스트림 전송을 위한 패트리 넷 기반의 상호대화형 동기화 기법", 한국정보처리학회 추계학술발표논문집 8권 2호, pp.1517-1520, 10.12. 2001
 [2] Allen, J.F, " Maintaining Knowledge About Temporal Intervals", CACM, 11, vol. 26, pp.832-843, 1983.
 [3] Naveed. Qazi " A Synchronization and Communication Model for Distributed Multimedia Objects, " Proc. Of the First ACM Conference on Multimedia, pp.147-155, Aug. 1993.
 [4] Lynda Hardman Dick C. A. Bulterman, " Composition and Linking in Time-based Hypermedia", European Union ESPRIT Chameleon project, 1999
 [5] 성경상, 황민구, 이기성, 이근왕, 오해석, " 버퍼레벨을 이용한 적응형 멀티미디어 동기화 재생 기법, 한국정보처리학회 추계학술발표논문집, 제8권, 제1호, pp.619-622, 2001.
 [6] 이기성, 이근왕, 이종찬, 오해석, " 대기시간을 이용한 적응형 멀티미디어 동기화 기법", 한국정보처리학회 논문지, 제7권 제2호, pp.649-655, 2000. 2.
 [7] 이양민, 이재기, " 가변 버퍼와 상호대화형 객체 삽입을 이용한 패트리 넷 기반의 스트림 동기화 기법", 한국멀티미디어학회 학술발표 논문집, 제5권 1호, pp431-436, 2002. 5.24