《主 題》

실시간 멀티미디어 동기화 기술

박 승 철, 최 양 희
(서울대학교 컴퓨터공학과)

■ 차 례 ■

Ⅰ. 서 론

컴퓨터를 통한 멀티미디어 처리 기술의 발달과 패킷 교환망으로 표현되는 컴퓨터 통신망의 고속화로 인해, 컴퓨터 통신망에서 다양한 실시간 분산 멀티미디어 응용의 실현에 관한 연구가 활발하게 진행되고 있다. 대표적인 응용으로는 탁상용 멀티미디어 화상 회의, 멀티미디어 정보의 실시간 검색등을 들 수 있다.

패킷 교환망에 오디오와 비디오와 같은 실시간 연속미디어를 포함하는 멀티미디어 응용들을 수용하기 위해서는 해결되어야 할 여러가지 새로운 문제들이 존재한다[14]. 그중 하나는 멀티미디어 데이타의 시간 관계를 유지하는 것이다. 오디오와 비디오 데이타는 기존의 텍스트등과 달리 데이타를 구성하는 요소(패킷)들간에 일정한 시간 관

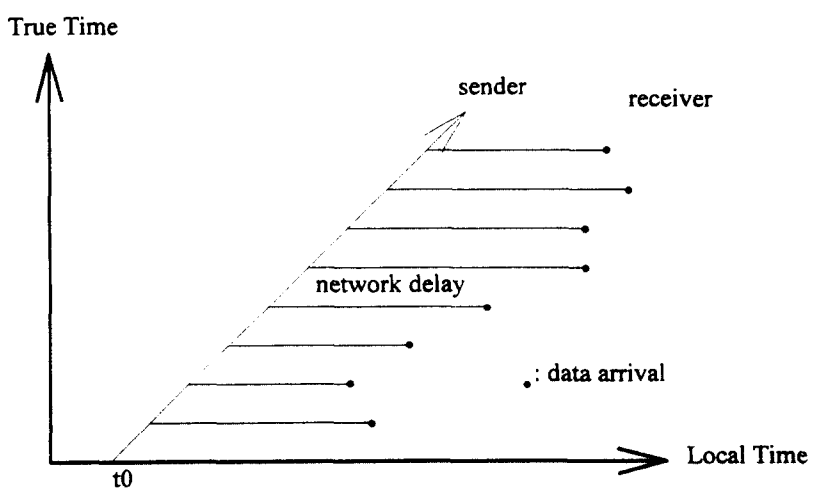


그림 1. 멀티미디어 통신에서 가변 지연시간의 영향

계가 존재한다. 예를들면, 패킷이 생성되는 시간 간격이 일정하게 주어진다. 그러나 패킷 교환망의 지연시간(delay)의 임의성은 목적지에 도착하는 데이타 요소간의 시간관계를 훼손시킬 수 있다. 그림 1에서 보는 바와 같이, 지연시간의 차이로 인해 동일한 시간 간격으로 훼손시킬 수 있다. 기존의 패킷 교환망에서 측정한 결과에 따르면 지연시간의 차이(지터)는 수백ms까지 이를 수 있음을 보여준다[19, 20]

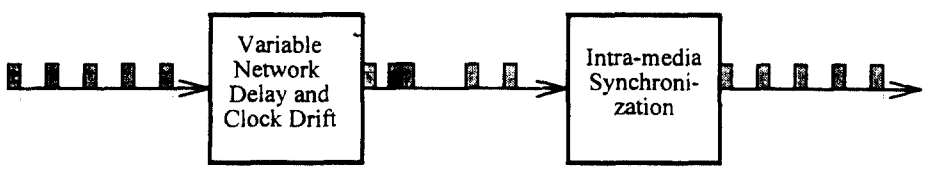
실시간 멀티미디어 동기화에서 고려되어야 할 또 하나의 요인은 송신자와 수신자간의 시계속도 불일치(clock drift)이다. 예를 들어 수신자의 시계속도가 송신자의 시계보다 느리다면, 수신자의 실질적인 데이타 처리 시간간격(playout interval)이 송신자의 데이타 생성간격(generation interval)보다 크게되어 원래의 시간 관계를 훼손한다. 반대의 경우도 마찬가지이다.

이러한 시간 관계 훼손을 실시간 멀티미디어 응용에 적합하도록 복원하는 것이 실시간 멀티미디어 동기화(realtime multimedia synchronization)기술이다[1, 2, 3]. 실시간 멀티미디어 동기화는 그림 2와 같이 미디어내 동기화(intra-media synchronization)와 미디어간 동기화(inter-media synchronization)으로 구분된다. 미디어내 동기화는 동일한 미디어를 구성하는 패킷들간의 시간 관계를 복원하는 것이고, 미디어간 동기화는 서로 다른 미디어의 패킷들간의 시간 관계를 복원하는 것이다. 미디어간 동기화의 대표적인 예는 립싱크(lip sync)이다.

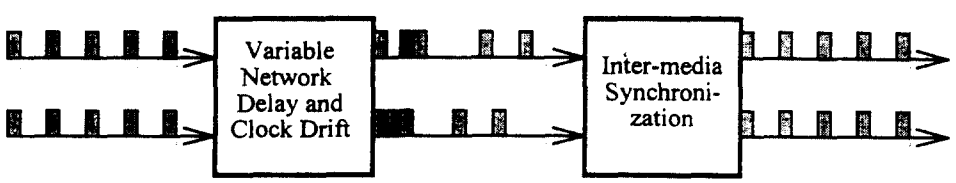
실시간 동기화는 멀티미디어 데이타의 서비스 품질(QoS : Quality of Service)를 원하는 범위내로 유지하기 위해 인위적으로 데이타 전송에 개입함으로써 이루어진다. 서비스 품질에 대한 요구사항은 응용에 따라, 그리고 사용하는 미디어의 특성에 따라 다를 수 있으므로, 동기화를 위한 인위적인 데이타 전송 개입은 응용에 따라, 미디어의 특성에 따라 적절하게 조정될 수 있어야 한다. 예를들면, 비디오 전화의 서비스 품질은 종단간 지연시간에 민감한 반면 어느 정도의 오류 발생에 대해서는 크게 영향을 받지 않는다. 특히 비디오 미디어의 오류에 대해서는 상대적으로 훨씬 덜 영향을 받는다. 반면, 스테레오 음악 분배 응용의 경우 종단간 지연시간에 대해서는 덜 민감한 반면, 오류 발생에 대해서 매우 민감하다.

실시간 멀티미디어 동기화에 대한 기존의 많은 연구들은 서비스 품질에 따른 유연한 동기화 서비스 제공 기능을 간과하고 있다. 따라서 본 연구에서는 응용에 따라 미디어의 특성에 따라, 적절한 동기화서비스를 효율적으로 제공할 수 있는 새로운 실시간 동기화 기법에 대해 소개하고자 한다. 본 기법은 지연시간의 상한값이 보장되지 않고 지연시간이 유동적인 일반적인 패킷 교환망에서 동작할 수 있으며, 송·수신자간의 시계 속도 차이를 보완할 수 있다. 그리고 미디어내 동기화뿐만 아니라 미디어간 동기화를 함께 제공할 수 있다.

이 기법을 소개하기에 앞서 2장에서 기존의 실시간 동기화 기법들의 문제점을 유연한 동기화 서비스 제공 측면



(a) 미디어내 동기화(intra-media synchronization)



(b) 미디어간 동기화(inter-media synchronization)

그림 2. 실시간 멀티미디어 동기화

에서 분석하고, 3장에서 본 기법의 접근 방법을 설명한다. 그리고 4장에서 새로운 동기화 기법의 미디어내 동기화 실현 알고리즘을 소개하고, 5장에서 미디어간 동기화 기법에 대해 설명한다. 그리고 마지막으로 6장에서 결론을 맺는다. 본 동기화 기법에 대한 구체적인 실험 결과들은 다른 논문으로 발표될 예정이다.

II. 기존 동기화 기법의 분석

실시간 멀티미디어 동기화에 대한 기존의 연구들은 정적 동기화(static synchronization), 송신자에 의한 적응형 동기화(adaptive synchronization by sender), 그리고 수신자에 의한 적응형 동기화(adaptive synchronization by receiver)방식으로 구분될 수 있다.

2.1 정적 동기화 방식(static synchronization)

정적 동기화 방식은 데이타 전송 시작시에 송신자와 수신자간의 지연시간 옵셋(패킷의 출발 시간과 처리시간과의 간격)을 충분히 크게 설정하고, 이를 데이타 전송 종료 시점까지 유지함으로써 패킷간의 지연시간 지터를 상쇄시키는 것이다. 즉, 수신자는 패킷들의 실질적인 도착시간과 상관없이 각 패킷이 송신자에서 출발한 시점으로부터 미리 정의된 패킷들의 실질적인 도착시간과 상관없이 각 패킷의 송신자에서 출발한 시점으로부터 미리 정의된 지연시간 옵셋 후에 해당 패킷을 처리하는 것이다. 이 방식을 그림으로 표현하면 그림 3과 같다.

이 방식은 RCAP[16], ST-II[17], 그리고 RSVP[18] 등과 같이 종단간 지연시간의 상한값을 보장할 수 있는 망과 시계속도 차가 없는 시스템 환경에서, 지연시간 옵셋을 보장된 지연시간 상한값으로 설정함으로써 사용할 수 있다. Tenet 그룹의 CMTP[4]는 실제로 이와 같은 방식으로 RCAP에서 동작한다. 그러나 이 방식은 지연시간 상한값이 보장되지 않고 시계속도차가 존재하는 환경에서 사용되기 어렵다. 만약 충분히 큰 임의의 지연시간 옵셋 설정과 함께 이 방식을 사용하는 경우에는 지연시간에 민감한 응용의 서비스 품질을 불필요하게 저하시킬 수 있다. 그림 3의 점선은 송신자와 수신자간의 시계속도 차이의 영향을 설명하고 있다.

2.2 송신자에 의한 적응형 동기화 방식 (adaptive synchronization by sender)

송신자에 의한 적응형 동기화 방식은 UCSD[5]에서 제안한 방법으로써, 망 지연시간의 상한값과 하한값이 보장되고 시계속도 차이의 상한값이 알려진 상황에서, 수신자의 한정된 버퍼의 오버 플로우(over flow)나 언더플로우(under flow) 방지를 방지하면서 연속적인 패킷 처리를 가능하게 하기 위해, 송신자의 패킷 전송 시점을 적절하게 조정한다. 패킷 전송 시점의 계산은 수신자로부터의 feedback 정보에 의거하여 이루어 진다.

이 방식은 주문형 비디오 응용과 같이 점대점(point to point) 통신 응용에는 효과적으로 사용될 수 있으나, 다상용 멀티미디어 회의 응용과 같이 다점간

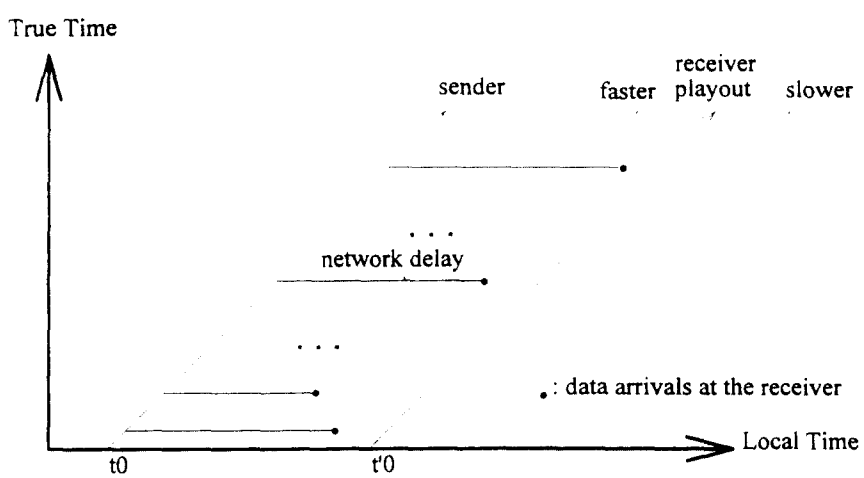


그림 3. 정적 동기화 방식(static synchronization)

(multipoint) 응용에서는 전송 시점 결정의 복잡도 증가로 인해 사용하기 어렵다. 그리고 이 방식은 여전히 지연시간 상한값의 보장을 요구하므로 일반적인 패킷교환망에서 지연시간에 민감한 응용에 사용하기 어렵다.

2.3 수신자에 의한 적응형 동기화 방식 (adaptive synchronization by receiver)

수신자에 의한 적응형 동기화 방식은 수신자가 실질망 지연시간의 변화에 따라 적절하게 동기화 조건(예, 지연시간 옵셋)을 조정함으로써 상황 변화에 적응한다. IETF(Internet Engineering Task Force)의 RTP [6], 오타와 대학의 SRTDD [8. 9], 그리고 BBN(Bolt and Newman Inc.)의 FSP [10]등 기존의 많은 실시간 동기화 기법들이 이 방식을 채택하고 있다. 이 방식은 기본적으로 많은 연속미디어들이 어느 정도의 오류 발생을 허용하고, 미디어를 구성하는 모든 패킷의 지연시간 옵셋에 대해 어느 정도의 차이를 허용하는 특성을 이용하는 것이다. 예를 들면, 발음구간(talkspurt)과 묵음구간(silent interval)의 반복으로 구성되는 음성 미디어의 경우 1%정도의 오류 발생을 허용하고, 수신측에서 묵음구간의 크기가 송신측에서의 그것과 어느정도 달라지더라도 품질에 큰 영향을 미치지 않는다 [7].

2.3.1 RTP의 동기화 기법[6]

RTP의 동기화 기법은 음성 미디어의 이러한 특성을 이용하여 인터네트와 같은 패킷 교환망에서 음성 미디어간의 실시간 동기화를 실현한다. RTP의 적응형 동기화 기법의 기본개념은 그림 4와 같이 설명될 수 있다. 여기에서 수신자는 발음구간과 묵음구간으로 구성되는 매 동기화 구간(synchronization- interval)의 시작 시점에서 실질 망 지연시간의 관찰 결과에 따라 지연시간 옵셋을 적절하게 조정한다.

전반적인 실질 지연시간의 크기가 현재의 지연시간 옵셋보다 작으면, 해당 동기화 구간의 시작 시점(t, 2)에서 지연시간 옵셋이 축소되고(t'2-t'1만큼), 반대의 경우(t'3), 현재의 지연시간 옵셋이 확장된다. 지연시간 옵셋의 조정은 직전 묵음구간 크기의 확장 또는 축소를 의미하므로 음성 서비스 품질에 별 영향을 미치지 않는다.

그러나 음성에서의 묵음구간과 같은 off 구간이 존재하지 않는 비디오, 음악등과 같은 연속 미디어 응용에 RTP에서와 같은 동기화 방식을 적용하기는 어렵다. 이들의 경우 지연시간 옵셋조정으로 인한 불연속성이 서비스 품질에 큰 영향을 미칠 수 있다. 더우기 송·수신자간에 시계속도에 차이가 있을 경우 불연속 상태의 발생 가능성이 더욱 커진다.

2.3.2 SRTDD [8. 9]

SRTDD에서는 수신자에 의한 적응형 동기화 기법을 미디어간 동기화(주로 ← lip ← sync)에 적용하고 있다.

SRTDD에서는 관련 데이타 스트림들을 동일한 크기의 동기화 구간으로 구분한다. 매 동기화 구간의 시작 시점에서 SRTDD는 데이타 스트림의 처리 시점을 다른 데이타 스트림들이 준비가 완료될 때 까지 지연시킴으로써 의도적으로 미디어간의 처리 시점을 일치시킨다.

이와 같은 간단한 미디어간 동기화 정책은 관련 데이타 스트림의 미디어내 동기화 오류를 유발할 수 있으며, 종단간 지연시간을 불필요하게 증가시킬 수 있다. 예를 들어, lip sync에서 늦게 도착한 비디오 패킷을 위한 음성 패킷의 의도적인 지연으로 인한 전체 지연시간의 증가는, 이로인해 얻을 수 있는 비디오 오류 감소 보다 전체 서비스 품질에 더욱 나쁜 영향을 미칠 수 있다. 이로부터 우리는 미디어간 동기화가 관련 데이타 스트림의 미디어내 동기화 요구 사항과 종단간 지연시간 요구사항등 응용의 서비스 품질을 제대로 반영할 수 있어야 한다는 것을 알 수 있다.

2.3.3 FSP [10]

BBN의 FSP는 수신자에 의한 동기화 조건의 조성이 응용이 지정한 동기화 함수(synchronization function)에 의해 이루어지게 함으로써 응용의 서비스 품질과 동기화가 연관될 수 있게 한다. 이 방식은 RTP에서와 같이 수신측에서 지연시간 옵셋을 적절하게 조정함으로써 적응형 동기화를 실현한다. 미디어간 동기화를 위해 모든 스트림의 수신자들은 자신의 실질 지연시간 관찰 결과를 주기적으로 상호 교환하고, 응용에 의해 주어진 공통의 동기화 함수는 이렇게 모아진 전체 지연시간 관찰값으로부터 조정된 지연시간 옵셋을 계산함으로써, 모든 스트림이 동일한 지연시간 옵셋을 가지도록 한다.

FSP의 동기화 기법에서 지적될 수 있는 문제점중의 하나는 RTP에서와 마찬가지로 지연시간 옵셋의 조정에만 의존하는 조정 정책이다. 불연속성에 민감한 응용의 경우 한 시점에서 조정될 수 있는 지연시간 옵셋의 크기에 한계가 있다. 지연시간 옵셋의 주기적인 조정도 하나의 문제점으로 지적 될 수 있다. FSP에서는 기본적으로 서로 다른 시스템에 존재하는 데이타 스트림의 수신자들간의

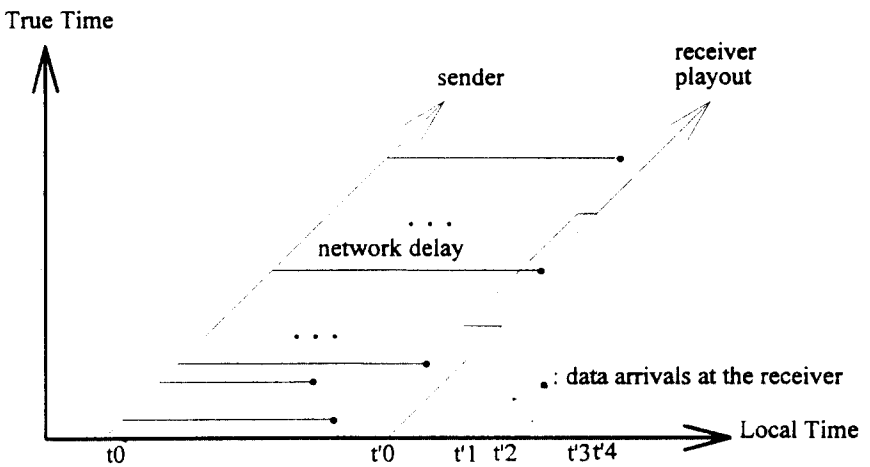


그림 4. RTP의 수신자에 의한 적응형 동기화

동기화에 촛점을 맞춰 설계되었으므로, 수신자들간의 지연시간 옵셋 조정 시점 일치를 위해 일정한 크기의 주기마다 조정이 이루어지게 한다. 따라서 음성 미디어의 발음구간 시작시점과 같이 임의의 시점에서의 조정을 통한 서비스 품질 저하 방지등과 같은 이점을 활용할 수 없다.

Ⅲ. 본 동기화 기법의 접근 방법

다양한 응용과 미디어에 대한 실시간 동기화 서비스를 다양한 망과 시스템 환경에서 효율적으로 제공하기 위한 동기화기법은 다음과 같은 특성을 가진다.

1) 수신자에 의한 적응형 동기화 기법(adaptive synchronization by receiver) : 지연시간이 시간에 따라 유동적이고, 송수신자간에 시계속도에 차이가 있을 수 있는 망과 시스템 환경에서 지연시간에 민감한 응용을 포함하는 다양한 응용을 지원하기 위해서는, 수신자가 현재의 동기화 상태에 따라 가능하면 가장 최선의 동기화 조건을 동적으로 설정할 수 있는 적응형 동기화 기법이 바람직하다. 필요한 경우 동기화 조건을 고정함으로써 정적 동기화 서비스를 제공할 수도 있다.

2) 유연한 동기화 구간 구분(flexible synchroniz-ation intervals) : 동기화 조건 조정의 단위가 되는 동기화 구간의 구분은 미디어의 특성에 따라 유연하게 이루어질 수 있어야 한다. 예를 들면, 음성미디어의 경우 발음 구간과 묵음 구간의 발생이 임의의 시점에 일어날 수 있고, 이를 동기화 구간으로 저장하기 위해서는 응용이 데이타 스트림의 동적 특성에 따라 동기화 구간을 지정할 수 있게 하여야 한다.

3) 지연시간 옵셋과 처리율 조정을 통한 보다 유연한 적응정책(flexible adaptation policy based on delay offset and playout rate adjustment) : 지연시간 옵셋 조정과 함께 동기화 구간내의 패킷 처리율을 동적으로 조정할 수 있도록 함으로써 동기화 상태에 보다 적절하게 적응할 수 있도록 한다.

4) 미디어내 동기화와 미디어간 동기화의 조화(harmonization between intra-media and intermedia synchronization) : 미디어간 동기화는 기본적으로 관련 데이타 스트림들의 미디어내 동기화 요구사항을 충족시키면서 이루어진다. 즉, 미디어간 동기화는 곧 관련 데이타 스트림의 미디어내 동기화를 포함한다. 따라서 미디어간 동기화로 인한 허용되지 않은 미디어내 동기 오류 유발등으로 인한 서비스 품질 저하를 방지한다.

5) 동기화 요구사항의 명시적 표현(user specification of synchronization requirements) : 응용의 서비스 품질 요구사항에 적합한 실시간 동기화를 제공하기 위해서 본 동기화 기법에서는 사용자가 동기화 요구사항을 지시할 수 있다. 데이타 전송중에 실시간 동기화를 위한 데이타 전송에의 인위적인 개입은 이 요구사항에 근거하여 이루어짐으로써, 동기화가 응용의 요구에 따라 적절하게 이루어질 수 있게 한다.

Ⅳ. 미디어내 동기화(Intra-media Synchronization)

4.1 실시간 멀티미디어 데이타 스트림(Realtime Multimedia Data Stream)

오디오나 비디오와 같은 실시간 멀티미디어 스트림은 기존의 텍스트등과 달리 데이타 단위간의 연속성(continuity)화 종단간 지연시간에 대한 허용 한계 특성을 가진다. 연속성은 송신자가 데이타 스트림 전송시 한정된 갯 수 이하의 데이타 단위(ADU : Application Data Unit)를 일정한 주기에 따라 생성함을 의미한다. 이렇게 생성된 데이타 단위들은 수신측에 주어진 지연시간 허용한계내에 도착하여야 한다. 주기의 크기, 하나의 주기 동안 생성되는 ADU의 갯 수, 그리고 종단간 지연시간 허용한계등은 응용에 의해 결정된다. 여기에서는 주기의 크기는 T_p, 그리고 종단간 지연시간 허용한계가 D_{target}로 주어지는 것으로 가정한다. 그리고, 하나의 주기 동안 생성되는 ADU 들을 논리적으로 하나의 단위, 즉 GADU(Group of ADU)로 칭한다. 모든 GADU가 존재하는 것으로 가정한다.

실시간 멀티미디어 데이타 스트림 전송에서 수신자의 데이타 처리 시작 시점(t'_0)은 송신자가 데이타 전송을 시작한 시점(t_0)으로부터 D_{target}보다 늦어질 수 없다. 그리고 일단 데이타 처리가 시작되면 수신측에서 연속적인 데이타 처리가 보장되어야 함을 알 수 있다. 여기서 S_t^i와 P_t^i는 각각 i번째 GADU의 출발시점과 수신측에서의 처리 시점을 나타낸다.

$$S_t^i = t_0 + i \times T_p \tag{1}$$

$$P_t^i = t'_0 + i \times T_p \tag{2}$$

$$P_t^i - S_t^i = t'_0 - t_0 \le D_{target} \tag{3}$$

이 식들이 의미하는 바는, 실시간 멀티미디어 데이타 스트림 전송에서 모든 GADU의 출발 시점(departure time)으로부터 수신측에서의 처리 시점(playout time)까지의 간격, 즉 지연시간 옵셋(delay offset)이 일정하게 주어져야 하고, 이것은 초기 지연시간 옵셋과 동일하여야 함을 의미한다.

4.2 수신자에 의한 적응형 동기화(adaptive synchronization by receiver)

패킷 교환망에서 종단간 지연시간은 가변적이므로 식(1), (2), (3)을 만족시키기 위해서는 지연시간 옵셋을 인위적으로 일치시키기 위한 버퍼링이 필요하다. 이론적으로 송신자와 수신자간의 초기 지연시간 옵셋(D_{eq}^0)을 망의 실질 지연시간의 최대값(D_{actual}^{max})보다 크게 설정하고, 이 값이 종단간 지연시간 허용한계 보다 작아질 때 식(1), (2), (3)의 관계가 만족된다. 즉, 이상적인 버퍼링 조건은 다음과 같다.

$$D_{actual}^{max} = \le D_{eq}^0 = t'_0 - t_0 \le D_{target} \tag{4}$$

그러나 이 조건은 실질 지연시간의 최대값(D_{actual}^{max})을 실제 환경에서 예측하기가 어렵기 때문에 실현되기가 어렵다. 비록 예측할 수 있다 하더라도 이것은 최악의 지연시간 이후에 데이타 처리 시작을 의미하므로 불필요한 지연시간을 초래할 수 있다. 수신자에 의한 적응형 동기화는 설정된 초기 버퍼링 조건을 상황의 변화에 맞게 동적으로 변화시킴으로써 최적의 실시간 동기화 서비스를 제공하는 것이다. 현재의 버퍼링 조건의 적정성 여부는 버퍼의 상태를 관찰함으로써 파악될 수 있다.

수신자에 의한 적응형 동기화를 요구하는 또 하나의 요인은 송신자와 수신자간의 시계속도 차이이다. 시계속도의 차이는 연속된 GADU에 대한 실질 지연시간 옵셋을 점점 증가 또는 감소시키므로 실시간 멀티미디어 스트림의 식(3)의 조건을 훼손시킨다. 이 상황은 수신자의 데이타 처리율을 적절하게 조정함으로써 개선될 수 있다. 비록 실질적인 시계속도의 차이를 정확하게 파악할 수 없지만, 버퍼의 상태를 관찰함으로써 간접적으로 그 차이로 인한 동기 불일치 정도를 파악할 수 있다.

4.3 버퍼 상태 관찰

하나의 실시간 멀티미디어 데이터 스트림에 대한 초기 지연시간 옵셋(D_{eq}^0) 설정 방식은 두가지가 있을 수 있다. 하나는 송신측의 데이타 전송 시작 시점과 수신측의 데이타 처리 시작 시점의 절대적인 차이를 지정하는 절대적인 방법이고, 또 다른 하나는 첫번째 GADU가 수신측에 도착한 시점으로부터 처리 시작 시점까지의 간격을 표현하는 간접적인 방법이다. 직접적인 방식에서의 초기 지연시간은 $D_{sta\ rtup}^{abs}$으로 표현되는 것으로 가정하고, 간접적인 경우 $D_{sta\ rtup}^{abs}$으로 표현되는 것으로 가정한다.

직접 또는 간접적인 방법으로 초기 지연시간 옵셋(D_{eq}^0)이 설정되면, 식 (1), (2), (3)에 의해 후속 GADU들의 처리 시점은 자동으로 결정된다. 버퍼의 상태는 GADU들의 버퍼 지연시간을 관찰함으로써 파악될 수 있다. i번째 GUAD의 버퍼 지연시간(I_d^i)은 다음의 식과 같이 버퍼에 도착한 실질 도착 시점(arrival time)으로부터 처리 시점(playout time)까지의 시간 간격으로 표현된다.

$$I_d^i = P_t^i - A_t^i \tag{5}$$

만약 버퍼 지연시간이 1보다 작다면 해당 GADU는 오류이다. 오류 GADU는 손실된 GADU로 구분된다. 늦게 도착한 GADU와는 식(6)의 관계를 만족시키고, 식(7)의 관계가 성립하면 해당 GADU는 손실된 것으로 간주된다. 여기서 D_{eq}^c는 현재의 지연시간 옵셋을 나타낸다. 손실된 GADU의 버퍼 지연시간은 식(8)에 의해 계산된다. 이것은 종단간 지연시간 허용 범위내에 도착하지 않은 GADU들은 모두 손실된 것으로 간주함을 의미한다.

$$P_t^i < A_t^i \le P_t^i + (D_{target} - D_{eq}^c) \tag{6}$$

$$P_t^i + (D_{target} - D_{eq}^c) < A_t^i \tag{7}$$

$$I_d^i = (P_t^i - S_t^i) - D_{target} \tag{8}$$

N개의 GADU에 대한 버퍼 지연시간을 관찰한 후 현재의 버퍼 상태(buffer status)가 결정된다. 만일 이들 중 오류 GADU의 비율이 응용에 의해 주어진 허용치, E^B보다 작다면, 해당 버퍼의 상태는 양의 상태(positive status)가 되고, 반대의 경우 음의 상태(negative status)가 된다. 음의 버퍼 상태는 현재의 전반적인 버퍼 지연이 너무 작아 처리 시점에 맞게 GADU들이 제대로 도착하지 못함을 나타낸다. 이것은 곧 현재의 지연시간 옵셋의 불충분하거나 또는 처리율이 너무 높음을 나타낸다. 반대의 경우, 현재의 지연시간 옵셋이 불필요하게 크거나 처리율이 낮음을 의미한다. 따라서 각각의 상태에 대해 전반적인 버퍼 지연시간을 적절하게 조정할 필요가 있으며, 그 값은 조정가능 지연시간(adjustable delay), D_{adj},으로 표현된다.

음의 상태(negative status)에서 조정가능 지연시간(Dadj)은 식(9)와 같이 오류 GADU의 버퍼 지연시간들을 인자로 가지는 하나의 함수로 표현될 수 있다. 여기서 n은 관찰된 N개의 GADU들중 오류가 발생한 GADU의 전체 갯수를 나타내고, I_d^i는 모든 오류 GADU의 버퍼 지연시간의 절대값을 오름순으로 정렬하였을 때 i번째 GADU의 버퍼 지연시간을 나타낸다.

$$D_{adj} = f(|I_d^1|, \cdots, |I_d^i|, \cdots, |I_d^n|) \tag{9}$$

그리고 이 함수의 값은 식 (10)에 의해 결정된다. 이 값의 의미는 전체적인 버퍼 지연시간이 이값의 크기 만큼 더해졌었으면 오류 GADU의 비율이 주어진 동기 오류 허용 한계(E^B)보다 작아진다는 것이다.

$$f(|I_d^1|, \cdots, |I_d^i|, \cdots, |I_d^n|)$$
$$= |I_d^i| s.t. (n-i)/N < E^B \le (n-i+1)/N \tag{10}$$

양의 상태(positive status)의 조정가능 지연시간도 비슷한 방식으로 계산될 수 있다. 이경우 조정가능 지연시간(D_{adj})는 식(11)과 같이 정상적인 GADU들의 버퍼 지연시간을 인자로 가지는 함수로 표현될 수 있으며, 그 값은 식(12)와 같이 결정된다. 여기서 I_d^j는 모든 정상적인 GADU의 버퍼지연시간을 오름순으로 정렬하였을 때 j번째 GADU의 버퍼 지연시간을 나타내고, 그값의 크기 만큼 버퍼 지연시간이 줄었어도 오류 GADU의 비율이 동기 오류 허용 한계를 넘지 않았음을 의미한다.

$$D_{adj} = f(I_d^1, \cdots, I_d^j, \cdots, I_d^{N-n}) \tag{11}$$

$$f(I_d^1, \cdots, I_d^j, \cdots, I_d^{N-n}) = I_d^j \; s.t. (n+j)/N < E^B$$
$$\le (n+j+1)/N \tag{12}$$

4.4 지연시간 옵셋과 처리율 조정

버퍼 상태 관찰 결과로 계산된 조정가능 지연시간은 버퍼링 조건을 인위적으로 적절하게 조정함으로써 흡수된다. 기존의 대부분 수신자에 의한 적응형 동기화 기법들의 경우 미리 정해진 동기화 구간 시작 시점에서의 지연시간 옵셋의 동적인 조정을 통해 버퍼링 조건을 인위적으로 조정한다. 지연시간 옵셋의 조정은 곧 조정되는 시간 만큼의 데이타의 불연속성 유발을 의미한다. 따라서 지연시간 옵셋 조정은 곧 조정되는 시간 만큼의 데이타의 불연속성 유발을 의미한다. 따라서 지연시간 옵셋 조정은 불연속성에 대한 응용의 허용가능한 범위내에서 이루어져

야 한다. 즉, 응용에 의해 제어가능한 지연시간 옵셋 조정을 통해 조정가능 지연시간이 흡수될 수 있어야 한다.

조정가능 지연시간을 흡수할 수 있는 또 하나의 방법은 동기화 구간내의 GADU처리율, 즉 처리구간의 크기를 조정하는 것이다. 대부분의 실시간 멀티미디어 데이터 스트림은 데이타의 처리구간 크기에 대한 어느 정도의 지터를 허용한다. 즉, 어느 정도의 범위내에서 처리율이 달라지더라도 서비스 품질에 거의 영향을 받지않는다. 따라서 본 동기화 기법에서는 지연시간 옵셋 조정과 처리율 조건을 병행하여 조정가능 지연시간을 흡수하는 이중적인 조정 정책을 채택하고 있다. 그림 5는 본 동기화 기법의 조정 정책에 대한 기본 개념을 설명하고 있다.

먼저 현재의 조정가능 지연시간 ($t'3 - t'1$) 은 동기화 구간의 시작 시점($t'3$)에서 응용이 허용하는 범위내에서 지연시간 옵셋 조정($t'3$-$t'2$)을 통해 흡수된다. 지연시간 옵셋 조정에 대한 응용의 허용 범위는 순간적인 불연속 상태 지속 시간에 대한 허용 기준으로써 D^{B}_{adj}로 주어지는 것으로 가정한다. 따라서, 현재 동기화 구간을 s번째 동기화 구간이라 할때 s번째 동기화 구간의 지연시간 옵셋(D^{s}_{eq})은 식(13)과 같이 조정된다. 즉, 직전 동기화 구간의 지연시간 옵셋(D^{s-1}_{eq})에 현재 조정가능 지연시간(D_{adj})과 지연시간 옵셋 조정 허용 한계(D^{B}_{adj})중작은 값을 더하거나(음의 버퍼 상태) 빼줌으로써 (양의버퍼 상태) 조정된 지연시간 옵셋을 구한다.

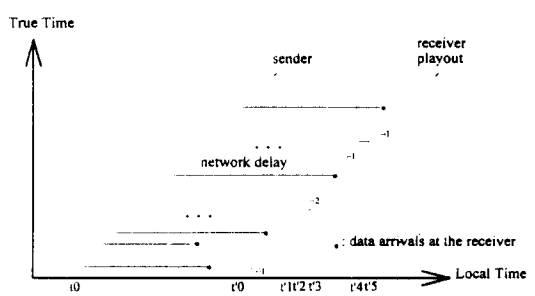


그림 5. 지연시간 옵셋과 처리율 조정

$$D^{s}_{eq} = D^{s-1}_{eq} \pm MIN(D_{adj}, D^{B}_{adj}), s \geq 1 \quad (13)$$

응용에 의해 주어진 한계(D^{B}_{adj})에 의해 지연시간 옵셋 조정에 의해 흡수되지 못한 조정가능 지연시간(P_{adj})은 식(14)와 같이 주어지고, 이 값은 해당 동기화구간의 일정 기간 동안 GADU 처리율을 조정($\theta_1 \longrightarrow \theta_2$)함으로써 최대한 흡수된다. 처리율 조정은 응용에 의해 주어지는 처리율 지터 허용 한계(P^{B}_{adj})내에서 이루어진다.

$$P_{adj} = MAX[\ 0, D_{adj} - MIN(D_{adj}, D^{B}_{adj})] \quad (14)$$

현재의 s번째 동기화 구간의 GADU 처리 구간의 갯수를 n이라 하고, 직전 동기화 구간의 처리 구간 크기를 T^{s-1}_{p}라 할때, s번째 동기화 구간의 처리율 조정으로 흡수할 수 있는 최대 조정가능 지연시간의 크기는 $n * T^{s-1}_{p} * P^{B}_{adj}$이 된다. 따라서 (14)의 P_{adj}의 값이 $n * T^{s-1}_{p} * P^{B}_{adj}$보다 큰 경우, s번째 동기화 구간의 모든 GADU 처리 구간의 크기는 식(15)와 같이 동일하게 조정된다.

이것은 응용이 허용하는 범위내에서 최대한의 처리율 조정을 통해 조정가능 지연시간을 흡수함을 의미한다.

$$T^{(s,i)}_{p} = T^{s-1}_{p} \pm T^{s-1}_{p} \times P^{B}_{adj}, 1 \leq i \leq n \quad (15)$$

그렇지 않은 경우, 현재 동기화 구간의 처음 k개의 GADU처리 구간의 크기만 조정되고 ((16), (17)), 나머지는 직전 동기화 구간의 처리 구간의 크기와 동일하게 남는다(18).

$$k = [\ P_{adj} \ / \ (T^{s-1}_{p} \times P^{B}_{adj})] \quad (16)$$

$$T^{(s,i)}_{p} = T^{s-1}_{p}, \pm T^{s-1}_{p} \times P^{B}_{adj}, 1 \leq i \leq k \quad (17)$$

$$T^{(s,i)}_{p} = T^{s-1}_{p}, k+1 \leq i \leq n \quad (18)$$

4.5 동기화 요구사항 표현

본 동기화 기법에서 실시간 동기화를 위한 인위적인 데이타 전송에의 개입은 초기 지연시간의 설정과 버퍼 상

태에 따른 지연시간 옵셋 및 처리율 조정으로 구분된다. 앞에서 설명한 바와 같이 초기 지연시간은 $D^{abs}_{sta\ rtup}$ 또는 $D^{abs}_{sta\ rtup}$의 형태로 주어질 수 있다. 지연시간 옵셋 및 처리율 조정은 버퍼의 상태를 판단함으로써 적절하게 이루어진다. 본 동기화 기법에서 버퍼의 상태 판단 기준은 응용에 의해 주어지는 동기 오류 허용 한계(E^B)이다.

그리고 지연시간 옵셋 조정은 응용에 의해 주어지는 지연시간 옵셋 조정 한계(D^B_{adj})에 의해 제어되고, 처리율 조정은 처리율 조정 한계 (P^B_{adj})에 의해 제어된다.

따라서 본 동기화 기법에서 동기화 요구 사항은 표 1의 인자들에 의해 적절하게 표현됨을 알 수 있다.

표1. 동기화 요구사항 표현인자

인 자	의 미
E^B	동기 오류 허용 한계
$D^{abs}_{\star tup}$	절대적인 초기 지연시간
$D^{re}_{\star up}$	상대적인 초기 지연시간
D^B_{adj}	지연시간 옵셋 조정 한계
P^B_{adj}	처리율 조정 한계

V. 미디어간 동기화

수신자에 의한 적응형 동기화 기법이 적용되는 두 개 이상의 실시간 데이타 스트림간의 미디어간 동기화는 그림 6에서와 같이 모든 스트림들의 처리 시작 시점이 동일하고, 모든 데이타 스트림들에 대한 지연시간 옵셋 및 처리율의 조정이 동일한 시점에서 동일하게 이루어질 때 이루어진다. 이것은 곧, 모든 데이타 스트림에 대한 미디어내 동기화를 위한 정적, 동적 동기화 조건이 완전히 일치될 때 비로소 미디어간 동기화가 이루어짐을 의미한다. 따라서 수신자에 의한 적응형 미디어간 동기화는 관련 데이타 스트림에 대한 미디어내 동기화 동작 조건들을 어떻게 일치시킬 것인가의 문제로 파악될 수 있다. 여기서 모든 스트림의 송신자들은 동일한 시계를 공유함을 가정하고, 모든 수신자들 역시 동일한 시계를 공유함을 가정한다.

본 동기화 기법에서 적응형 미디어내 동기화를 위한 정적, 동적 동기화 조건은 다음과 같은 6가지이다.

1) 초기 지연시간
2) 동기화 구간의 크기
3) 버퍼 상태 관찰 구간의 크기
4) 지연시간 옵셋 조정 한계
5) 처리율 조정 한계
6) 조정가능 지연시간

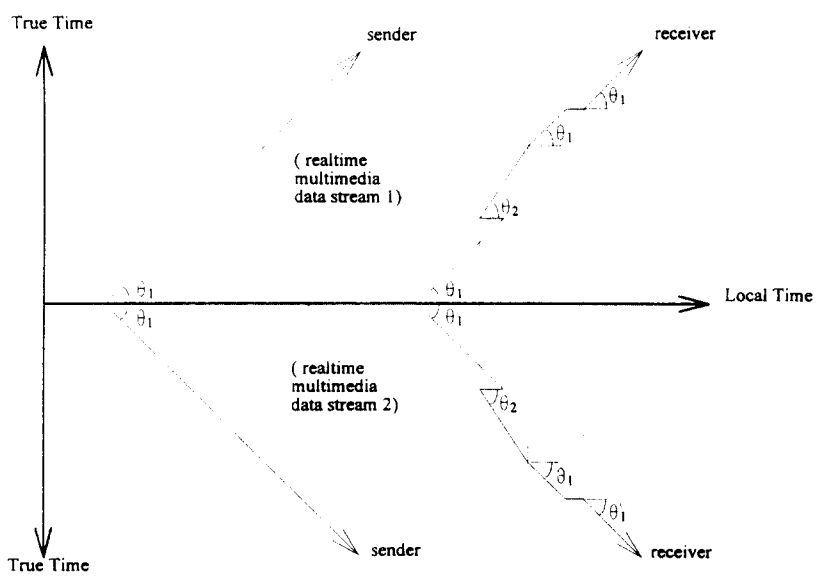


그림 6. 적응형 미디어간 동기화

여기서 1-5)까지의 동작 조건은 응용에 의해 데이타 전송 시작 전에 결정되는 정적 조건이며, 6)은 망과 시스템 상태에 따라 변화하는 동적 조건이다. 정적 조건들중 동기화 구간의 크기와 버퍼 상태 관찰 구간의 크기는 미디어내 동기화에 대한 명시적인 요구사항과 무관한 것으로써, 응용에 의해 적절하게 설정되는 것으로 보아도 무방하다. 나머지 정적 조건들은 각 스트림에 대한 미디어내 동기화 요구사항을 표시하는 인자들로써, 이들의 인위적인 일치는 관련 데이타 스트림에 대한 미디어내 동기화 요구사항 정도를 고려하여 적절하게 결정할 수 있다.

본 동기화 기법에서 동적인 조건인 조정가능 지연시간의 일치는 응용이 적절한 공유 관찰 모델(shared measurement model)을 지정함으로써 이루어진다. 공유 관찰 모델은 응용에 따라 다양하게 정의될 수 있다. 여기에서는 대표적인 예로써 주종 모델(master/slave model)과 보수적 모델(conservative model)을 제시하고자 한다.

보수적 공유 관찰 모델에서는 그림 8에서와 같이 모든 데이타 스트림에 대한 버퍼 상태의 관찰이 이루어지고, 각각에 대한 조정가능 지연시간이 계산된다. 그리고 이들 중 가장 보수적인 값을 공유 조정가능 지연시간으로 선택한다.

VI. 결 론

기본적으로 실시간 멀티미디어 동기화는 송신측에서 주어진 미디어내 또는 미디어간 시간 관계가 수신측에서도 응용이 수용할 수 있는 범위내에서 유지될 수 있도록 데이타 전송에 인위적으로 개입하는 것이다. 따라서 실시간 동기화를 위한 인위적인 개입은 응용의 서비스 품질 요구 사항과 잘 연계될 수 있어야 한다. 본 논문에서는 다양한 망환경과 시스템 환경에서 응용의 서비스 품질 요구사항과 적절하게 연계되어 동기화 서비스를 제공할 수 있는 보다 유연한 실시간 동기화 기법의 개발에 대해 논의하였다.

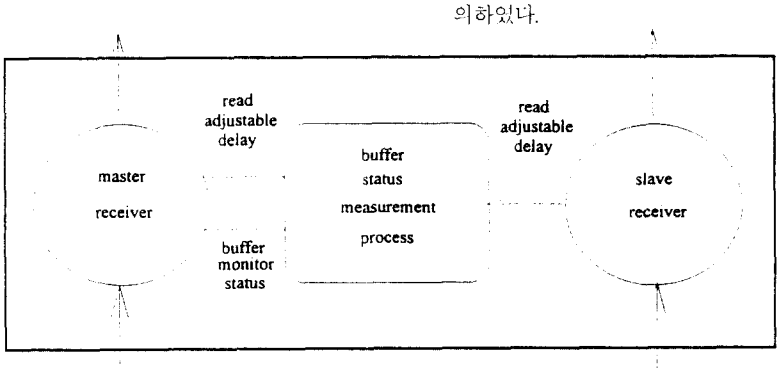


그림 7. 주종 공유 관찰 모델

주종 고유 관찰 모델에서는 관련 데이타 스트림중 하나가 주 스트림으로 선택되고 나머지는 모두 종 스트림으로 간주된다. 이 모델에서 조정가능 지연시간은 주 스트림의 버퍼 상태 관찰 결과에 따라 계산되고, 이값이 모든 데이타 스트림의 조정가능 지연시간으로 공유된다. 이 모델은 비디오 전화등과 같이 우선 순위가 분명한 스트림들간의 미디어간 동기화에 사용된다.

주종 모델에서는 종 스트림의 버퍼 상태가 전체 동기화 조건의 조정에 반영이 되지 않으므로 종 스트림의 미디어내 동기화가 희생될 수 있다. 따라서 이 모델은 스트레오 음악 분배 응용의 경우와 같이 특정 미디어의 미디어내 동기화 희생을 허용하지 않는 응용에서는 사용할 수 없다.

본 연구에서 제시한 유연한 실시간 동기화 기법은 기존의 방법들과 마찬가지로 수신자에 의한 적응형 동기화 방식에 근거하고 있으나, 버퍼의 상태에 따라 지연 시간 옵셋 뿐만아니라 처리율을 적절하게 조정할 수 있도록 함으로써 미디어의 특성에 맞는 동기화 조건의 동적 조정을 가능하게 하고 있다. 그리고, 실시간 동기화를 위한 인위적인 개입은 응용이 지정하는 동기화 요구사항에 의해 분명하게 제어될 수 있도록 함으로써, 동기화로 인한 불필요한 또는 허용되지 않는 서비스 품질 저하가 방지된다. 그리고, 미디어간 동기화는 관련 데이타 스트림들의 미디어내 동기화 응용의 특성에 맞게 일치시킴으로써 미디어내 동기화와 조화속에서 이루어 질 수 있도록 한다.

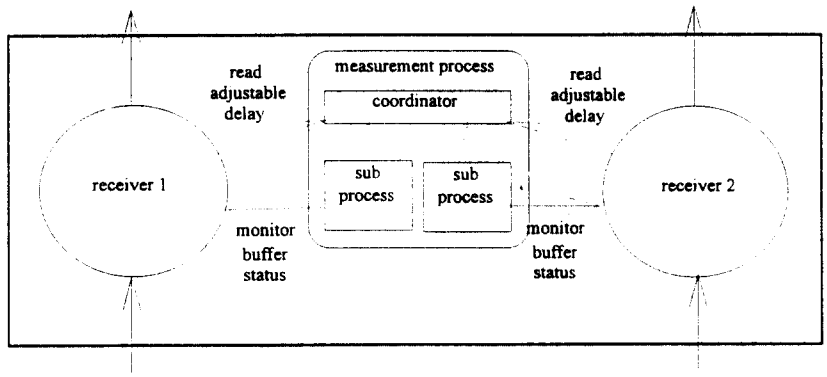


그림 8. 보수적 공유 관찰 모델

References

[1] R. Steinmetz, "Multimedia Synchronization Techniques : Experiences Based on Different System Structures", Proc. of IEEE Multimedia '92. 1992.

[2] K.Ravindran, "Real-Times Synchronization of Multimedia Data Streams in High Speed Networks", Proc. of IEEE Multimedia '92. 1992.]

[3] D.Shepherd and M.Salmony, "Extending OSI to Suport Synchronization Required By Multi media Applications", Computer Comm., Vol.13, No.7, Sept, 1990.

[4] M.Moran and B.Wolfinger," Design of a Continuous Media Data Transport Service and Protocol", International Computer Science Institute, TR-92-019, April 1992.

[5] S.Ramanathan and P.V.Rangan, "Feedback Techniques for Intra-Media Continuity and Inter-Media Synchronization in Distributed Multi media Systems", The Computer Journal, March 1993.

[6] H.Schulzrinne, "Issues in Designing a Transport Protocol for Audio and Video Conferences and Other Multiparticipant Real Time Applications", IETF AVT-WG, Internet draft, Oct. 1993.

[7] M.A.Montgomery, "Techniques for Packet Voice Synchronization", IEEE Journal on Selected Areas in Communications, Vol. SAC-1, Dec. 1983.

[8] L.Li,A.Karmouch, and N.D.Georganas, "Synchronization in Real Time Multimedia Data Delivery, Proc. of ICC'92, 1992.

[9] L.Li,A.Karmouch, and N.D.Georganas, "Multi-mdeia Segment Delivery Scheme and Its Performance for Real Time Synchronization Control", Proc. of ICC'94, 1994.

[10] J.Escobar, D.Deutsch, and C.Patridge, "Flow Synchronization Protocol", Proc. of Globecom'92, 1992.

[11] T.D.C.Little and A.Ghafoor, "Multimedia Synchronization Protocols for Broadband Integrated Services",IEEE Journal on Selected Areas in Communications, Vol.9. No.9, Dec. 1991.

[12] L.Ehley, B.Furht, and M.Ilyas, "Evaluation of Multimedia Synchronization Techniques", Proc. of The International Conference on Multimedia Computer and Systems, Boston, May 1994.

[13] G.Karlsson and M.Vetterli, "Packet Video and Its Integration into the Network Architecture", IEEE Journal on Selected Areas in Communication, Vol.7, No.5, June 1989.

[14] L.Besse, L.Dairaine, W.Tawbi, and K.Thai, "Towards and Architecture for Distributed Multi-media Applications Support", Proc. of The International Conference on Multimedia Computer and Systems, Boston, May 1994.

[15] H.Cho and Yanghee, Choi, "Synchronization Protocol for Multimedia Communications", Proc. of APCC, Daegeon, Korea, Aug, 1993.

[16] A.Banerjea ad B.Mah, "The Realtime Channel Administration Protocol", Proc. of the 2nd International Workshop on Network and OS Support for Digital Audio and Video", HeidelBerg, Now. 1991.

[17] C.Topolcic, ed., "Experimental Internet Stream Protocol(version 2)", RFC 1190, Oct. 1990.
[18] L.Zhang, S.Deering, D.Estrin, S.Shenker, and D.Zappala, "RSVP : A New Resource ReSerVation Protocol", IEEE Network, Sept. 1993.
[19] D.Sanghi, et al., "Experimental Assessment of End-to-End Behavior on Internet", Proc. of INFOCOM'93. 1993
[20] J-C Bolot, "End-to-End Packet Delay and Loss Behavior in the Internet", Proc. of SIGCOM'93, 1993

박 승 철

- 1985년 2월 : 서울대학교 계산통계학과 (학사)
- 1987년 2월 : 한국과학기술원 전산학과 (석사)
- 1987년 - 1990년 : 한국전자통신 연구소
- 1990년 - 1992년 : 한국 IBM 소프트웨어 연구소
- 1992년 - 현재 : 서울대 컴퓨터공학과 박사과정

최 양 희

- 정보통신 제11권 2호, 1994 p.124 참조