# 지연 축소를 위한 컴퓨터 영상회의 시스템의 스트림 동작 구조 비교

이 경 희[†] · 김 두 현[†] · 강 민 규[††] · 정 찬 근[†]


## 요 약

본 논문에서는 데스크탑 컴퓨터 환경하에서 영상 회의 시스템의 오디오와 비디오 데이타를 입출력하는 하드웨어 및 이를 이용하는 소프트웨어의 구조에 대해 논하고 이를 구현함에 있어 시간 지연 관점에서 스트림의 동작 방식에 대해 분석한다. 영상 회의, 원격 교육, 주문형 비디오 등의 멀티미디어 응용 서비스에 이용되어 질 수 있는 멀티미디어 입출력 처리기인 MuX는 멀티미디어 데이타의 입출력, 동기화, 접합, 분리, 합성 등에 대한 다양한 처리 요소를 제공한다. 본 논문은 이를 조합하여 영상 회의를 구현하는 방법에 대해 기술하며 오디오, 비디오 등의 스트림의 동작 방식을 Master clock과 자체 시계를 이용하는 방식을 비교한다. 오디오, 비디오 스트림의 출발점이 되는 입력부에서는 자체 시계를 이용한 방식이 Master clock을 이용하는 방식보다 시간 지연면에 있어 이득이 있었으며 오디오, 비디오 스트림의 동기화를 위해 스트림을 접합하는 경우, 주기적인 오디오 스트림을 채널 객체의 자체 시계로 이용하는 것이 Master clock을 이용하는 것보다 오디오 스트림의 전송에는 지연을 줄이는 효과는 있으나 비디오 스트림의 경우에는 큰 영향을 주지 못 함을 보였다.


# Comparisions of stream activation mechanisms in computer based teleconferencing systems for low delay

Kyung Hee Lee[†] · Doohyun Kim[†] · Min-gyu Kang[††] · Chan Geun Jung[†]


## ABSTRACT

In this paper, we present a hardware architecture and a software architecture for computer based teleconferencing systems. And also we analyse stream activation mechanisms for them from the viewpoint of delay. MuX that is a multimedia I/O server provides various processing elements for data I/O, synchronization, interleaving and mixing. We describe methods to build teleconferencing systems with the elements and compares the technique using master clock with the technique using self clcok. In the phase of data input, the technique using self clock is better than the technique using master clock. When we generate interleaved stream from audio and video stream and activate channel objects by periodic audio stream as activation clock, delay from input audio stream to interleaved stream is reduced but delay for video stream is not reduced as much as in the case of audio stream.


## 1. 서 론

통신망의 발전, 중앙 처리 장치의 연산 능력의 향상, 고속 로컬 버스의 등장 등에 힘입어 산술 데이타 연산 위주의 컴퓨터의 응용이 오디오, 비디오, 그래픽 등의 비정형화된 데이타를 처리하는 멀티미디어 응용 서비스로 확장되고 있다. 이러한 변화 속에서 아

† 정 회 원:한국전자통신연구원 분산멀티미디어연구실
†† 정 회 원:한국전자통신연구원 분산멀티미디어연구실
논문접수:1996년 8월 16일, 심사완료:1996년 12월 18일

날로그 방식의 전용 케이블을 이용한 영상 회의 시스템도 컴퓨터를 이용한 데스크탑 영상 회의 시스템으로 변모하고 있으며 주문형 비디오, 홈 쇼핑, 원격 진료 등 다양한 응용 프로그램들이 개발되고 있다. 이러한 멀티미디어 응용 프로그램을 쉽게 개발할 수 있도록 해주는 멀티미디어 입출력 서버인 MuX가 개발되었다[2].

컴퓨터 영상 회의 시스템과 관련된 연구로는 IMA (Interactive Multimedia Association)[11], ITU-T(International Telecommunication Union-Telecommunication Standardization Sector)[12] 등의 표준화 활동과 BERKOM[13] 프로젝트 등을 들 수 있다. ITU-T의 T.120 시리즈는 ISDN(Integrated Services Digital Network)을 배경으로 다지점 영상 회의를 위한 표준을 제시하고 있는데 이는 MCU(Multipoint Connection Unit)에 의한 토큰형 다자간 영상 회의를 제공한다. BERKOM 프로젝트는 ISO(International Standard Organization) 모델에 입각한 멀티미디어 트랜스포트 프로토콜의 개발을 기본 목표로 하고 그 프로토콜의 상위 응용 프로그램으로 멀티미디어 회의나 멀티미디어 메일을 제공하고자 한다. 그러나 각 응용 프로그램을 상위 계층의 프로그램에서 하위 계층의 프로그램까지 수직적으로 개발하여야 하는 단점이 있다.

그러나 기존의 전화와 같은 아날로그 오디오 신호를 전기적 회로로 단순 연결하는 방식에서보다 현재 컴퓨터 시스템에서 널리 사용하는 것으로 아날로그형의 오디오, 비디오 신호를 디지탈화하고 이를 가공, 처리하는 방식에서는 데이타의 버퍼링, 압축/복원 알고리즘의 수행, 통신망의 트래픽 등으로 인해 오디오 혹은 비디오 전송의 지연이 늘어나는 현상이 발생할 수 있다. 이러한 지연 현상은 상대방의 응답을 기다리는 대기 시간을 길어 지게 한다. 영상 회의와 같은 대화형 실시간 멀티미디어 응용 프로그램에서 지연 현상은 서비스의 질을 떨어뜨리는 중요한 요인 중의 하나로 작용할 수 있으므로 원활한 영상 회의를 위해서는 오디오, 비디오 전송의 지연을 줄여야 한다.

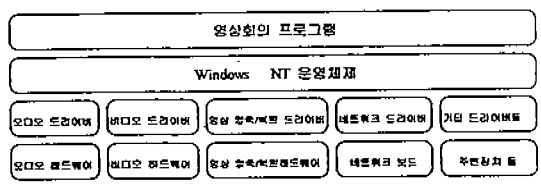
기존 오디오, 비디오 전송 지연과 관련된 연구는 통신망을 사이에 둔 시스템 종단간(end-to-end delay)의 지연을 주로 다루고 있으며 전송되어 지는 데이타 프레임의 지연, 지터(jitter) 등을 관찰하여 수신측에서 지연, 지터링을 최소화하는 연구가 이루어지고 있다[14].

본 논문에서는 데스크탑 컴퓨터와 Windows NT운영체제상에서 MuX를 기반으로 한 플랫폼상에서 스트림 연결에 기반한 객체 지향적 영상 회의 시스템의 하드웨어, 소프트웨어 구조와 오디오, 비디오 스트림의 시간 지연 축소화를 위해 영상 회의 시스템의 송신측에서 행해 질 수 있는 스트림의 연결 및 동작 방식을 비교 분석한다.

본 논문의 구성은 제 2장에서 영상 회의 시스템의 하드웨어 및 소프트웨어의 구조 및 기본 요소들에 대해 기술한다. 제 3 장에서는 영상 회의 소프트웨어의 하부 구조인 멀티미디어 스트림 처리부(MuX)의 기능 및 구조에 대해 설명하고, 제 4장에서는 영상 회의를 위한 MuX서버내의 스트림 연결 방법 및 지연 축소를 위한 스트림의 동작 방법에 대해 논한다. 그리고 제 5 장에서는 결론을 기술하였다.

## 2. 영상 회의 시스템의 구조

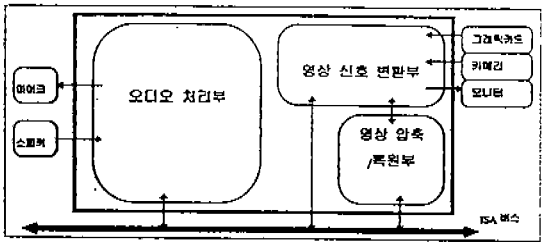
본 장에서는 영상 회의를 위한 전체 시스템의 구조를 각 부분 별로 설명한다. 시스템의 전체 구조를 나타낸 그림이 아래 (그림 1)에 나타나 있다. 최상의 계층에 회의 소집, 종료 등의 관리 및 오디오/비디오 스트림을 연결해 주는 영상 회의 프로그램이 위치하고 있으며 운영체제는 마이크로소프트사의 Windows NT3.5가 있다. 그 이하의 블록은 오디오, 비디오, 네트워크 등의 입출력 장치들과 이를 구동시켜 주는 디바이스 드라이버들이 위치하고 있다.




(그림 1) 영상 회의 시스템의 구조
(Fig. 1) An architecture of a teleconferencing system

### 2.1. 하드웨어 구조

이 절에서는 영상 회의를 위한 오디오 신호의 녹음과 재생이 이루어 지고 영상 신호의 압축 복원이 이루어 지는 하드웨어에 대해 기술한다.

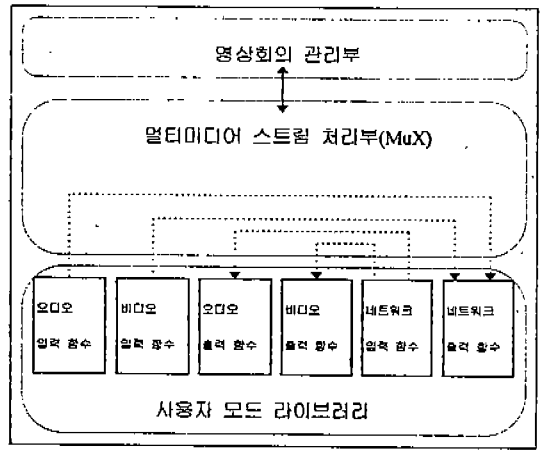



(그림 2) 영상 회의 하드웨어의 구조
(Fig. 2) A hardware architecture of a teleconferencing system

오디오, 비디오 신호를 처리하는 하드웨어는 PC (Personal Computer)상의 ISA(Industry Standard Architecture) 버스 인터페이스를 이용한다. 오디오 처리부는 아날로그의 오디오 신호를 마이크로부터 입력받아 디지탈 신호를 생성하거나 메모리에 저장된 디지탈 신호를 아날로그 신호로 변환하여 스피커로 출력한다. 이때 사용할 수 있는 샘플링율은 5Khz에서 48Khz이고 모노와 스테레오 신호를 처리할 수 있다. 그리고 샘플 당 비트 수로는 선형 PCM(linear PCM)인 경우8비트, 16비트이고 ADPCM(Adaptive Delta Pulse Code Modulation)인 경우는 4비트이다. 메모리와 오디오부 사이의 입출력은 CPU(Central Processing Unit)의 이용율을 높이기 위해 programmed I/O방식을 제하고 DMA(Direct Memory Access) 방식을 이용하여 CPU의 개입없이 데이타의 전송이 이루어지게 한다. 영상 회의 시 발언과 청취를 동시에 할 수 있도록 각각 독립적인 DMA 채널을 이용하여 오디오 신호의 녹음과 재생이 동시에 이루어 질 수 있다. 이와 아울러 입출력 신호의 크기를 조절할 수 있는 볼륨 콘트롤 기능을 내장하고 있다.

영상 신호 변환부는 비디오 카메라로부터 입력되는 아날로그의 영상 신호를 입력받아 디지탈화 하여 영상 압축/복원부로 전달하고 영상 압축/복원부에서 전달되는 디지탈 영상 신호를 아날로그로 변환하는 기능을 가지고 있고 그래픽 카드에서 만들어진 영상에다 복원된 디지탈 영상을 중첩(overlay)시킬 수 있으며 중첩된 영상을 모니터로 출력한다.

영상 압축/복원부는 영상 신호 변환부로부터 전달되어진 디지탈 영상 데이타를 H.261방식으로 압축하는 기능과 H.261 방식으로 압축된 신호를 복원하는 기능을 가지고 있다. 영상 압축/복원부는 오디오 변환부와 같이 영상 압축 및 복원을 동시에 할 수 있으며 ISA 버스의 DMA 방식을 이용하여 주기억장치와 데이타를 주고 받는다. H.261의 영상 압축 방식은 이전 영상과의 차분 정보를 이용하여 압축하는 방식으로 화면상의 움직임의 정도에 따라 데이타 발생량과 주기가 가변적인 특성을 갖는다.




(그림 3) 영상 회의 프로그램 구조
(Fig. 3) A software architecture of a teleconferencing system

### 2.2. 소프트웨어 구조

영상 회의 프로그램은 크게 세 부분으로 나누어 진다. 최상위 계층은 영상 회의 소집, 진행, 종료 등의 회의 관리 기능을 수행하는 영상 회의 관리부이고 중간 계층은 영상 회의를 위한 오디오, 비디오 데이타 스트림을 연결하는 멀티미디어 스트림 처리부이고 최하위 계층은 사용자 모드 라이브러리 계층으로 디바이스 드라이버를 호출하는 함수들의 집합인 사용자 모드 라이브러리로 구성되어 진다. 멀티미디어 스트림 처리부는 상위 계층으로부터 전달되어 오는 입출력 요구를 처리해 준다. 멀티미디어 스트림 처리부는 영상 회의 등의 상위 계층에서 전해 오는 멀티미디어 데이타 처리 요구를 받아 해당 디바이스를 연결하여 데이타들이 전달되도록 스트림을 형성해 준다. 즉 멀티미디어 스트림 처리부는 멀티미디어 데이타 스트림을 위한 일관성있는 인터페이스를 제공하므로 멀티미디어 데이타 처리를 필요로 하는 응용 프로그램은 멀티미디어 스트림 처리부(MuX)의 API(Appli-

cation Program Interface)를 이용하여 멀티미디어 스트림을 콘트롤할 수 있다. 이 멀티미디어 스트림 처리부는 영상 회의 응용 프로그램에만 국한 되지 않고 주문형 비디오, 원격 교육, 홈 쇼핑 등 다양한 응용 서비스의 멀티미디어 데이타 처리에 이용되어 질 수 있도록 설계되었다.

사용자 모드 라이브러리는 오디오 하드웨어, 비디오 하드웨어, 네트워크 하드웨어의 기능을 영상 회의 응용 프로그램과 같은 사용자 모드 프로그램들이 쉽게 이용할 수 있도록 한 함수들의 집합이다. 각 함수들은 기본적으로 Windows NT에서 제공하는 함수들을 이용한다.

본 논문에서 사용한 오디오 라이브러리는Windows NT의 low level 오디오 서비스이며 현재 구현되어 사용하고 있는 오디오 입출력 형태는 비트 수로 8비트, 샘플링 율로 11.025Khz, 모노 방식이다.

비디오 입출력 함수는 동기식의 입출력을 행한다. 사용자가 데이타를 요구하면 비디오 드라이버를 통해 비디오 하드웨어에 저장된 압축 영상 데이타를 전달받는다. 데이타를 입출력 요구 시 사용자 모드 프로그램은 다른 수행을 하지 않고 입출력 완료 시까지 대기한다. 데이타 입출력 시 소요되는 시간과 입출력 데이타의 크기는 입력 영상에 따라 가변적이다.

기타 여러 하드웨어를 위한 사용자 모드 함수들도 기본적으로 Windows NT에서 제공하는 함수를 그대로 이용한다.

## 3. 멀티미디어 스트림의 처리

이 장에서는 2장에서 기술한 멀티미디어 스트림 처리부인 MuX는 멀티미디어를 처리하기 위한 다양한 기능을 제공하며 객체 지향 방식을 이용하여 구현되었다. MuX는 영상 회의 응용 서비스에만 국한되는 것이 아니라 다양한 멀티미디어 응용 서비스에 이용되어 질 수 있다. 아래 절들에 기본 요소 및 기능에 대해 기술한다.

### 3.1. 기본 요소

#### 3.1.1. 미디어와 프레임

본 논문에서 취급하는 미디어는 컴퓨터 영상 회의에서 사용되는 실시간 미디어이다. 실시간 미디어(media)의 특징은 재생이 불가능함을 물론이고 프레임율(Frame Rate)이나 샘플링율(Sampling Rate) 등 QoS(Quality of Service)가 실시간 입출력 단말장치의 성능에 일차적으로 좌우된다.

프레임(frame)은 시간 축 상에서 임의 시간 간격에 해당되는 미디어 데이타를 포함하고 있는 구조체로서, 미디어의 종류나 임의 시점의 QoS(Quality of service)에 따라 그 크기가 가변적일 수도 있다. 본 논문에서 논하는 프레임은 입출력 시 사용자가 느끼는 감각적 단위와 다를 수도 있다. 예를 들어 비디오의 경우 한 프레임을 사용자가 시각적으로 느끼는 정지화면 한 장을 한 프레임으로 삼을 수 있으나 압축 방법에 따라서는 그렇지 않을 수도 있다. 또한 오디오의 경우에는 기본적으로 프레임 개념이 없기 때문에 가상적인 프레임을 만들어야 할 경우도 있다. 따라서 프레임은 연속 미디어의 컴퓨터 처리를 위하여 사용하는 내부 구조체이다. 모든 프레임은 입력 단말장치에 의하여 부여된 시간표지(Time Tag)를 갖고 있다.

#### 3.1.2. 스트림

스트림은 단일 미디어의 흐름으로서 미디어 데이타 입출력의 가장 기본적인 구조이다. 스트림은 각 미디어에 대하여 화일, 디바이스, 네트워크, 분배기(Copier), 접합기(Weaver), 분리기(Unweaver), 합성기(Mixer)와의 입출력 관계를 맺어 줌으로써 미디어간의 데이타 흐름이 형성되도록 한다.

##### 3.1.2.1. 스트림 기능

- 화일, 디바이스, 통신 등 입출력 장치와의 입출력 기능을 수행한다.
- 미디어 데이타를 분배 또는 합성하기 위한 Mixer, Weaver, Copier, Unweaver 등과 접속하고 이들과 데이타를 입출력하는 기능을 수행한다.
- 다른 스트림과의 동기화를 위해 스트림으로 들어오는 데이타의 시간 표지를 기록한다.

##### 3.1.2.2. 스트림 처리 요소

- Source 객체

Source 객체는 입력용 미디어를 소유하고 있는 객체로서 스트림에 흐르는 단위 데이타인 프레임의 시발점이 된다. 따라서 Source 객체에서는 연결된 Medium

객체를 통해 각종 입력 장치로부터 프레임 단위의 정보가 읽혀 들여진다. 따라서 Source 객체는 미디어에서 프레임을 읽어 Destination으로 전달하는 주기적 동작을 위한 쓰레드의 출발점이다.

• Destination 객체

Destination 객체는 출력용 미디어를 소유하고 있는 객체로서 Source에서 생성된 프레임의 종착점이 된다. 따라서 프레임 구조체를 위하여 할당된 메모리를 돌려준다.

• Stream 객체

Stream 객체는 Source 객체와 Destination 객체를 연결하여 주는 객체로 Source 객체에 연결된 Medium 객체로부터 데이타를 입력받아 Destination 객체에 연결된 Medium 객체에 전달하는 역할을 한다.

• Medium 객체

Medium 객체는 오디오, 비디오, 네트워크 등의 데이타의 입출력이 실지로 발생하는 디바이스를 나타내는 객체로 Source 객체 혹은 Destination 객체에 연결되어 해당 디바이스로 데이타를 입출력하는 함수들로 구성되어 있다. 이들은 2. 2절에 기술된 사용자 모드 라이브러리를 호출하는 함수들로 구현되었으며 DLL(Dynamic Linking Library) 형태를 갖는다.

### 3.1.3. 채널 (Channel)

스트림이 미디어 입출력의 가장 기본적인 독립 구조라 할 수 있다면, 채널은 여러 개의 독립적인 스트림을 하나의 스트림으로 합치는 작업을 한다. 즉 두 개의 서로 다른 오디오 스트림을 믹싱하여 믹싱된 하나의 스트림을 생성하거나, 오디오 스트림과 비디오 스트림과 같은 이질의 스트림을 동기화를 위해 인터리브(interleave)시키는 역할을 한다. 채널도 Source 객체와 마찬가지로 하나의 단위 스트림의 출발점으로 볼 수 있으며 주기적으로 데이타를 접합 또는 합성하여 스트림 객체로 전달한다.

#### 3.1.3.1. 채널 처리 요소

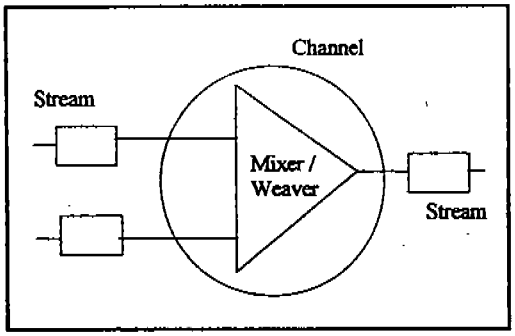
채널은 상기의 서비스들을 제공하기 위하여 합성기(Mixer), 접합기(Weaver) 등의 요소를 제공한다.

• 합성기(Mixer)

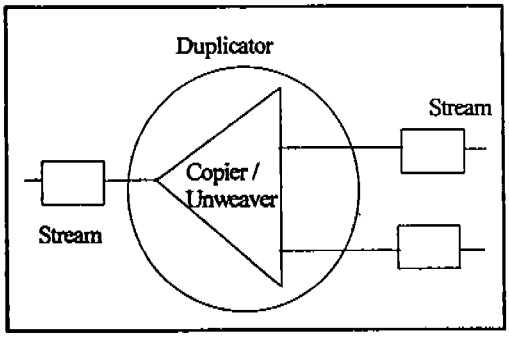
합성기의 주요 기능은 입력되는 각 스트림들로부터 채널 시점에 가장 근사한 시간 표지를 갖는 프레임을 인출하여 하나의 합성된 프레임을 생성하고 이를 출력 스트림으로 출력시키는 기능을 한다. 예를 들어 3자 영상 회의 시 상대방 두 음성을 믹싱하는 경우에 사용한다.

• 접합기(Weaver)

스트림 통합과 스트림간 동기화는 합성기에 의하여 주로 이루어 지지만 때로는 합성을 하지않고 동기를 맞추어 단순히 접합만을 하여 인터리브시켜야 할 경우가 있다. 이를 위한 것이 접합기이다. 접합기는 합성기와 마찬가지로 각 입력 스트림으로부터 채널 시점에 가장 근사한 시간 표지를 갖는 프레임을 인출하여 복합적인 구조의 하나의 접합 프레임을 생성하고 이를 출력 스트림으로 출력시키는 기능을 수행 한다. 이렇게 만들어진 복합 프레임은 Weaver의 짝인 Unweaver에 의하여 다시 단위의 프레임으로 분리된다. Unweaver는 Duplicator를 통하여 각각의 개별 스트림으로 분리하여 사용할 수 있다.




(그림 4) 스트림 합성 및 접합 개념도
(Fig. 4) The concept of stream mixing and interleaving




(그림 5) 스트림 분리(Unweave) 및 복제(Copy) 개념도
(Fig. 5) The concept of stream unweaving and copying

3.1.4. Duplicator

Duplicator는 채널 객체와 대칭되는 개념으로서 (그림 5)와 같이 하나의 스트림을 여러 개의 스트림으로 복제하거나 인터리브되어 있는 스트림에서 각 단위 스트림들을 분리하는 기능을 한다.

3.1.4.1. Duplicator 처리 요소

• 복제기(Copier)

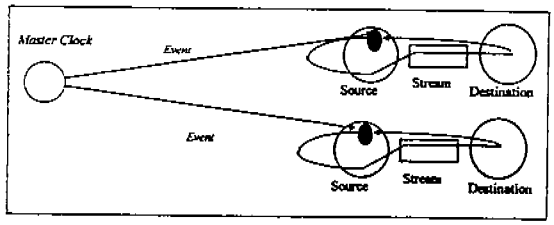
복제기는 하나의 입력 스트림을 다중의 목적지에 보내는 것이다. 예를 들어 비디오 디바이스로 부터 생성되는 YUV 입력 스트림을 디스플레이하기 위해 RGB 비디오로 변환된 스트림으로 출력되어질 수 있고, 원거리에 위치한 곳으로 보내기 위해 비디오 스트림을 MPEG(Motion Picture Expert Group)과 같은 압축된 스트림으로 출력할 수 도 있다.

• 분리기(Unweaver)

스트림 분리기는 입력되는 스트림이 하나 이상의 미디어를 접합(Interleaving)하고 있을 경우 각 접합된 미디어들을 분리하여 각각을 해당 출력 측 스트림으로 출력한다. 일례로 영상 회의 시 사정에 의하여 PCM 방식의 오디오와 H.261의 비디오가 채널 객체의 접합기 기능에 의하여 접합되어 전송되어 오는 경우 이를 분리하여 오디오와 비디오를 각 해당 목적지인 오디오 및 비디오 디바이스로 전달 주는 것이 가능하다.

3.1.5. 데이타 스트림의 제어

Source객체와 채널 객체 등 단위 스트림의 출발점으로 데이타의 전달을 시작하는 객체를 동적 객체(active object)라 한다. MuX는 MuX내에 생성된 동적 객체들을 제어하는 Master Clock을 가지고 있다. Master Clock은 등록된 동적 객체들의 semaphore에 주기적으로 event를 보냄으로 정해진 시간에 동적 객체가 수행될 수 있도록 한다. Source객체가 원하는 주기를 Master clock에 등록할 수 있으며 기본적으로는 100msec로 동작하게 되어 있다. 동적 객체는 동적인 역할을 하는 객체로서 Master Clock이 보내는 event를 수신하는 쓰레드를 가지고 있다. 일반적으로 Master Clock에서 보내온 event를 동적 객체가 수신하게 되면 동적 객체는 미디어 처리에 필요한 단위 작업들을 동적으로 수행한다. 그렇지만 동적 객체가 Master Clock에 등록되지 않고도 자체적으로 주기적 동작이 가능하다면 동적 객체가 Master Clock의 제어를 받지 않아도 된다. 반면 MuX내 객체 중 다른 객체들은 동적 객체에 접속되어 접속된 동적 객체를 수행하는 쓰레드의 지배하에 수행된다.

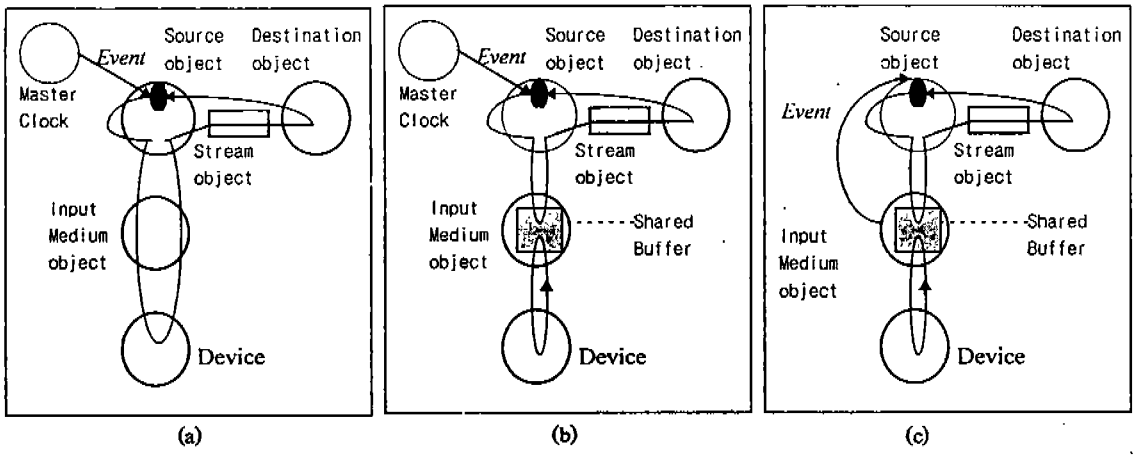



(그림 6) Master Clock동작의 예
(Fig. 6) An example of Master clock operation

3.2. 영상 회의를 위한 스트림의 생성

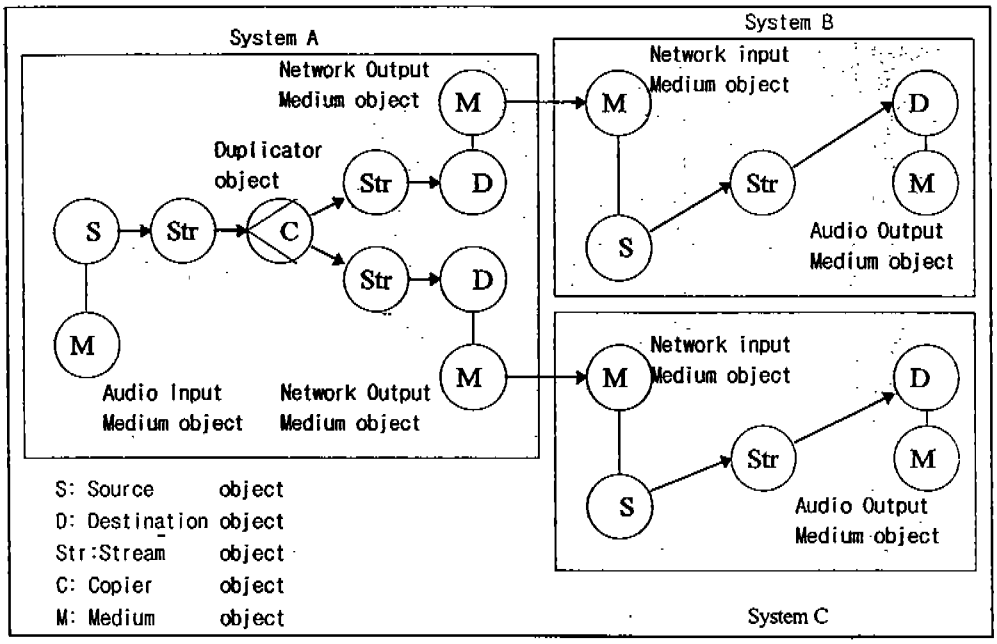
본 절에서는 3장에서 설명된 각종 멀티미디어 응용 프로그램을 위한 객체를 이용하여 다자간 오디오 비디오 회의를 구성하는 방식을 설명한다. 각각에 오디오 입력용 마이크와 출력용 스피커, 그리고 비디오 입력용 카메라와 출력용 모니터가 구비되어 있고, MuX를 통하여 이러한 각 멀티미디어 디바이스와 프레임을 입출력한다.

(그림 7)의 (a)에 전형적인 스트림의 연결 구조가 나타나 있다. Master clock이 등록된 Source 객체에 주기적으로 event를 보낸다. Source 객체는 event를 수신 후 Medium 객체에 데이타를 요구하게 되고, Medium
객체는 해당 디바이스로부터 데이타를 읽은 후 그 데이타를 프레임 객체에 실어 전송한다. 만약 Medium 객체가 디바이스에 데이타를 요구한 순간 준비된 데이타가 없으면 디바이스는 데이타를 이 시점부터 준비하기 시작하여, 준비된 후 데이타를 전달하게 되면 데이타의 요구 시점과 Source 객체에서 데이타를 전달하는 시점 사이에 지연이 발생하게 된다. (그림 7)의 (a)에서 발생하는 시간 지연을 줄이기 위해 (b)와 같은 연결을 할 수 있다. (a)와 (b)의 차이는 Medium 객체와 디바이스 사이에 공유 버퍼를 두어 디바이스로부터 데이타를 입력받아 공유 버퍼에 쌓아 두는 독립적인 쓰레드(thread)를 생성하고 Medium 객체는 이 버퍼의 데이타를 스트림으로 전달하는 것이다. 이

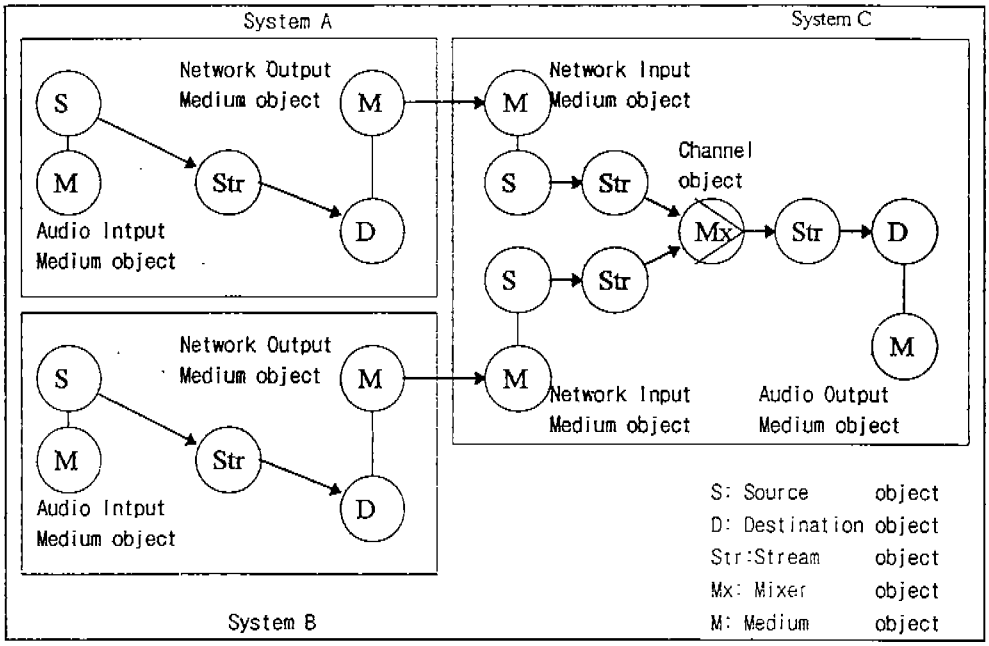



(그림 7) 스트림의 생성 및 동작
(Fig. 7) Stream generation and activation

는 Source 객체가 Master clock으로부터 event를 수신한 경우 Medium 객체로부터 데이타를 즉시 얻을 수 있으므로 오디오 디바이스에 데이타가 준비되기까지 기다리는 시간을 줄일 수 있다. 그러나 버퍼를 이용하므로 데이타가 발생하는 주기와 Master clock에 의해 불려지는 주기가 일치하지 않을 있으며 데이타가 생성된 시점과 데이타 전송을 위해 Master clock에 의해 불려지는 시점은 다를 수 있어 시간 지연이 생길 수 있다. (그림 7)의 (c)는 Medium 객체가 자체적인 시계를 가진 경우 그 시계를 이용하여 스트림의 주

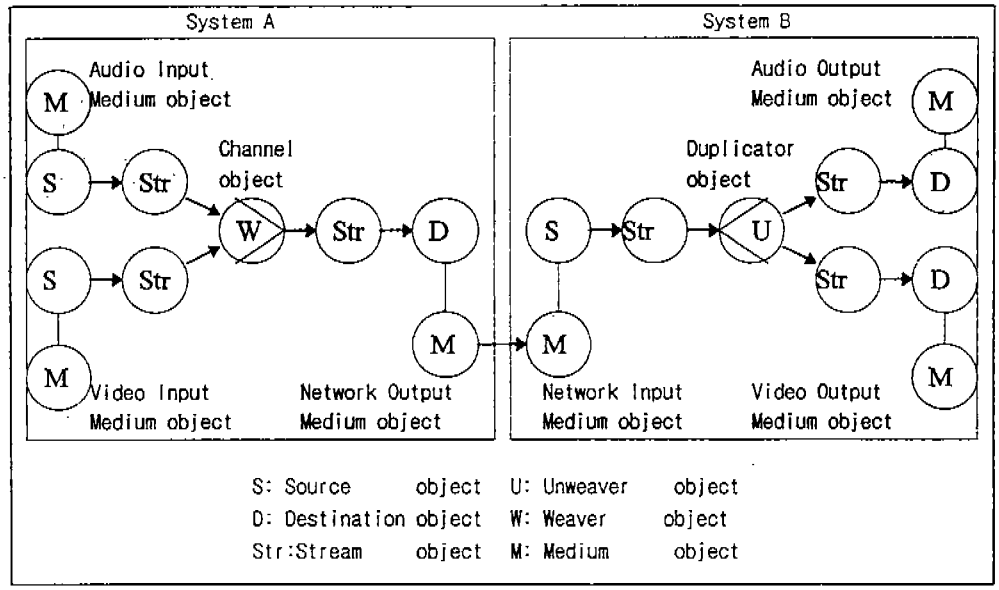



(그림 8) 오디오 분배 시 객체들의 연결 예
(Fig. 8) An example of object connections for audio distribution




(그림 9) 오디오 믹싱 시의 객체들의 연결 예

(Fig. 9) An example of object connections for audio mixing




(그림 10) 오디오와 비디오 스트림의 접합 전송과 수신 분리 시 객체들의 연결 예

(Fig. 10) An example of object connections for audio and video stream interleaving

기적 동작을 일으키는 스트림의 연결이다. 즉 Medium 객체의 쓰레드에서 데이타가 준비되었으면 연결된 Source 객체에 event를 전송하여 데이타의 전송을 하도록 한다.

(그림 8)에 MuX의 객체를 조합하여 두 시스템에 오디오 스트림을 전달하는 예가 나타나 있다. 시스템 A에서는 오디오를 입력받을 디바이스를 나타내는 오디오 입력 Medium 객체를 생성하고 이를 Source 객체에 연결시킨다. 스트림을 복제하는 Duplicator 객체를 생성하고 이와 Source 객체를 Stream 객체로 연결한다. 출력쪽의 시스템에서는 오디오 출력을 하는 디바이스를 나타내는 Medium객체를 생성하고 이를 Destination 객체와 연결시킨다. 비디오 스트림의 경우도 오디오 스트림의 연결에서와 같이 비디오 입출력 Medium 객체, Source 객체, Stream 객체, Destination 객체, Duplicator 객체를 연결하면 된다. (그림 9)는 두 시스템에서 전달되는 오디오 스트림을 믹싱하여 스피커로 출력하는 과정을 나타낸 것이다. (그림 10)은 오디오 비디오 스트림의 동기화를 위해 오디오 스트림과 비디오 스트림을 접합(interleave)시켜 전송하는 예를 나타낸다.

## 4. 지연 축소를 위한 영상 회의 스트림의 동작 구조

본 장에서는 3장에 기술한 MuX를 이용한 영상 회의 스트림의 데이타 전달을 빠르게 하기 위한 방법에 대해 논한다.

### 4.1. 영상 회의 시스템에서의 지연

디지탈 컴퓨터는 아날로그의 오디오, 비디오 신호를 디지탈로 변환한다. 디지탈화되는 데이타를 저장하기 위해 디지탈 컴퓨터에서는 버퍼를 이용한다. 이 버퍼를 이용하여 데이타를 전달하는 방식은 버퍼를 통과하는 지연이 생기므로 아날로그 신호를 전기적으로 연결하는 방식보다 지연이 길어 지게 된다.

멀티미디어 응용 프로그램은 주문형 비디오와 같은 저장된 멀티미디어 데이타를 전송하는 것이 있고 영상 회의와 같은 실시간에 멀티미디어 데이타를 입력받아 전송하고 전송되어 오는 멀티미디어 데이타를 실시간에 재생하여야 하는 것으로 크게 분류할 수 있다. 이때 송신측에서 데이타가 발생되어 수신측에서 재생될 때까지 걸리는 시간을 종단간 지연(end to end delay)라고 한다. 저장된 데이타를 전송하는 응용 프로그램에서는 종단간 지연이 크게 문제가 되지 않는다. 종단간의 지연이 있더라도 데이타들간의 시간 간격을 일정하게 유지하여 지터링을 제거하면 사용자는 원래 저장된 데이타를 순서와 시간에 따라 재생하여 볼 수 있다. 그러나 실시간 데이타 입출력을 행하는 영상 회의와 같은 응용 프로그램에서는 지연은 서비스의 질에 큰 영향을 미친다. 지연이 길어 지면 송신측 사용자가 오디오, 비디오 데이타를 전송하고 이에 대한 답을 기다리는 시간이 길어 지게 된다. 즉 디지탈 컴퓨터 상의 사용자가 사용하는 영상 회의 시스템에서 종단간 지연이 길어 지면 아날로그 전화를 이용하는 것보다 서비스의 질이 떨어 질 수 있다. 그러므로 컴퓨터를 이용한 영상 회의 시스템에서는 지연을 축소하여 서비스의 질을 높여야 한다.

종단간 지연은 디바이스로부터 데이타를 입력받는 부분과 입력된 데이타를 전송하는 통신망 그리고 수신된 데이타를 재생하는 부분에서 발생하는 지연의 합으로 구성되어 질 수 있다. 지연에 관련된 연구는 지연에 민감한 저장 멀티미디어 데이타를 재생하는 경우 동기화를 위한 연구[15]와 압축 알고리즘 및 통신망의 라우터 등에서 지연을 축소하려는 연구 등이 있다[16]. 본 논문에서는 데이타를 전송하는 부분과 데이타를 재생하는 부분의 지연을 고려하지 않고 송신측의 데이타의 입력이 이루어지는 부분에서 이루어 질 수 있는 오디오, 비디오 데이타 스트림과 오디오 스트림과 비디오 스트림을 접합한 스트림의 전달을 스트림 동작 구조에 따른 시간 지연만을 비교 분석한다.

### 4.2. 지연 축소를 위한 스트림의 동작 제어

이 절에서는 영상 회의의 기본 미디어인 오디오, 비디오 데이타의 빠른 전송을 위한 MuX 서버내의 스트림 연결 구조를 비교한다. 본 논문에서는 CBR (constant bit rate) 특성을 갖는 주기적 연속 미디어의 예로 선형 PCM 방식의 오디오 스트림과 VBR(variable bit rate)의 특성을 갖는 H.261방식의 비디오 스트림을 예로 들어 각 Source 객체와 채널 객체에서 Master clock을 이용하는 방식과 자체 시계를 이용한 스트림

연결 방식을 지연(delay)을 중심으로 비교한다.

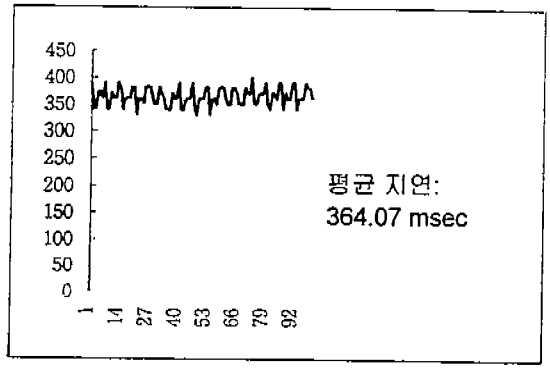
실험을 위한 하드웨어 환경으로 CPU는 Pentium 60Mhz, 주기억장치 용량은 32Mbyte이고 그래픽 보드는 1M의 VRAM을 가진 Cirrus사의 GD5434칩을 이용하였다. Ethernet을 이용하여 네트워크 환경을 갖추었고 2.1절에 기술한 하드웨어를 이용하여 오디오/비디오 처리를 하였으며 운영체제로는 마이크로소프트사의 한글 Windows NT3.5를 사용하였다. 이때 사용한 오디오 데이타는 PCM방식으로 11.025Khz의 샘플링율, 8비트, 모노 형식이고 비디오 데이타는 H.261형식이며 초당 평균적으로 200kbps의 데이타의 발생량을 갖는다. 데이타 추출을 위해 시험 프로그램의 스트림 연결 부분의 입력단과 출력단에 시스템의 시간을 Windows NT3.5가 실행된 후 진행된 시간을 millisecond단위로 출력해 주는 Win32 API의 하나인 GetCurrentTime()를 사용하여 측정하였다. 오디오 데이타와 비디오 데이타의 입출력을 위해 사용한 버퍼는 각각 1Kbyte의 버퍼를 이용하였다.

#### 4.2.1. 오디오 스트림의 연결 및 동작

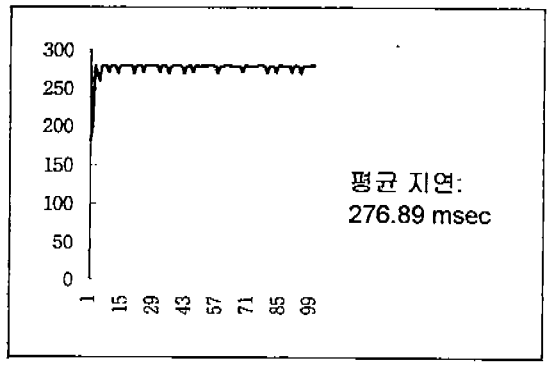
본 논문에서 사용하는 오디오는 선형 PCM 방식으로 데이타의 발생량이 주기적이다. 오디오 입력을 구현한 오디오 입력 Medium 객체는 Windows NT의 low-level 입출력 함수를 바탕으로 구현되었다. Low-level 오디오 입력은 비동기식 입력을 채택하고 있다. 사용자가 오디오 데이타의 입력을 요구하면 오디오 데이타의 입력이 완료될 때까지 기다리지 않고 입력 요구만 디바이스 드라이버에 전달하고 반환된다. 이 후 데이타의 입력이 완료되었으면 Windows NT의 입출력 시스템이 입력을 요구한 프로그램으로 callback을 보낸다.

(그림 11)에 영상 회의 시스템의 오디오 입력부에서 입력 스트림 연결 시 (그림 7)의 (b)와 (c)방식으로 스트림을 동작시키는 경우의 지연 값을 측정한 결과가 나타나 있다. (그림 11)의 지연은 오디오 입력을 요구한 시점과 오디오 데이타가 Source 객체에서 Stream 객체로 전달될 때의 시간차를 의미한다. (그림 11)의 (a)는 (그림 7)의 (b)와 같이 Source객체와 Medium 객체 사이에 공유 버퍼를 두고 Medium 객체에서 생성된 데이타를 공유 버퍼에 연속적으로 전달하도록 하고 Source객체는 Master clock에서 전달되는 주기적 이벤트 신호에 따라 데이타를 전송하는 방식에서의 지연을 나타낸 것이다. (그림 11)의 (b)는 Medium 객체에서 시스템으로부터 callback을 수신하여 데이타의 입력이 완료되었을 때 Source 객체에 이벤트 신호를 보내는 방식의 스트림 연결 시 지연을 나타낸 그림이다. X축은 각 데이타 프레임의 순서를 나타낸 것이고 Y축은 시간 지연을 millisecond로 나타낸다.

(그림 11)에서 보는 바와 같이 주기적 연속 미디어인 오디오 스트림의 입력 시 자체 시계에 의한 스트림의 동작 방식이 Master clock을 이용하는 방식보다 시간 지연면에서 유리하다고 할 수 있겠다.




(a)




(b)

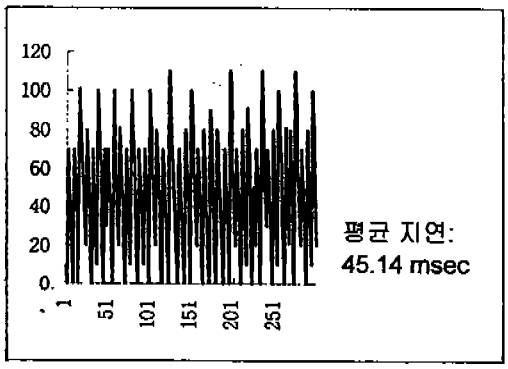
(그림 11) 오디오 스트림 입력 시의 지연
(Fig. 11) Delay of audio stream input

#### 4.2.2. (비디오 스트림의 연결 및 동작

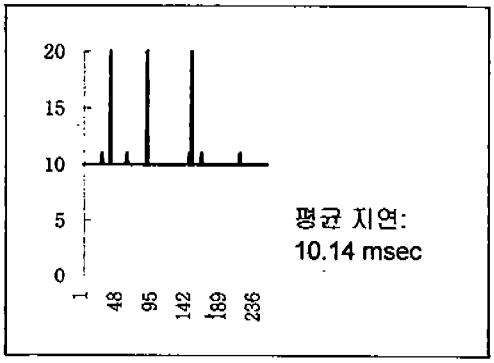
본 논문에서 사용하는 비디오는 H.261 방식으로 데이타의 발생량이 시간에 따라 가변적이다. 데이타

의 입출력은 오디오의 경우와는 다르게 동기식 입출력을 한다. 사용자가 비디오 데이타의 입력을 요구하면 오디오 데이타의 입력이 완료될 때까지 기다리고 있다가 데이타의 입력이 완료되었을 때 반환된다.

(그림 12)에 영상 회의 시스템의 비디오 입력부에서 입력 스트림 연결 시 (그림 7)의 (b)와 (c)방식으로 스트림을 동작시키는 경우의 지연 값을 측정한 결과가 나타나 있다. (그림 12)의 지연은 비디오 입력을 요구한 시점과 비디오 데이타가 Source 객체에서 Stream 객체로 전달될 때의 시간차를 의미한다. (그림 12)의 (a)는 (그림 7)의 (b)와 같이 Source 객체와 Medium 객체 사이에 공유 버퍼를 두고 Medium 객체에서 생성된 데이타를 공유 버퍼에 연속적으로 전달하도록 쓰레드를 생성하고 Source 객체는 Master clock에서 전달되는 주기적 이벤트 신호에 따라 데이타를 전송하는 방식에서의 지연을 나타낸 것이다. (그림 12)의 (b)는 Medium 객체에서 데이타가 생성되면 Source 객체에 이벤트 신호를 보내는 방식의 스트림 연결 시 지연을 나타낸 그림이다. X축은 각 데이타 프레임의 순서를 나타낸 것이고 Y축은 시간 지연을 millisecond로 나타낸다.




(a)




(b)

(그림 12) 비디오 스트림 입력 시의 지연
(Fig. 12) Delay of video stream input

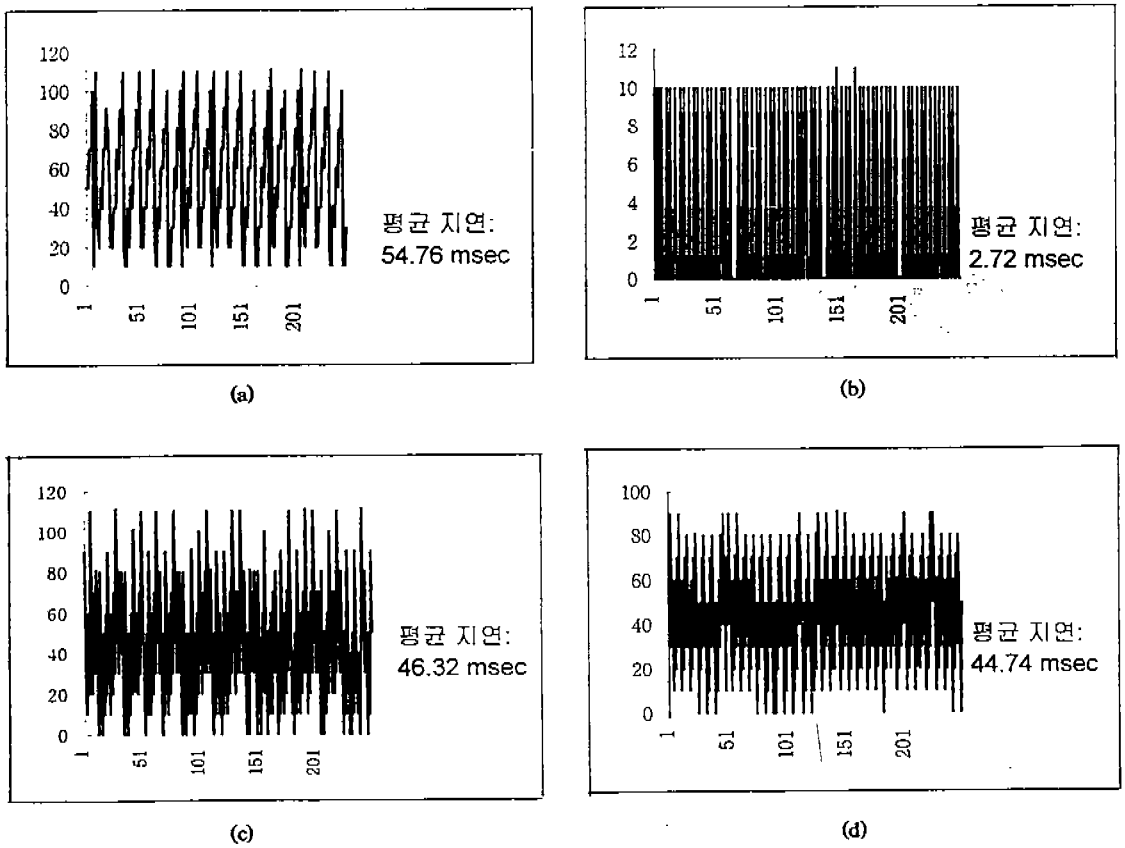
(그림 12)에서 보는 바와 같이 비주기적 연속 미디어인 비디오 스트림의 입력 시 자체 시계에 의한 스트림의 동작 방식이 Master clock을 이용하는 방식보다 시간 지연면에서 유리하다고 할 수 있겠다.

#### 4.2.3. 채널 객체의 제어

채널 객체는 동질의 여러 스트림을 믹싱하여 하나의 스트림을 생성하거나 동기화를 위해 이질의 스트림들을 접합하여 접합된 스트림을 생성하는 역할을 수행한다. 채널에 연결된 입력 스트림은 주기적 혹은 비주기적으로 채널 객체의 입력 버퍼들에 데이타 프레임을 쌓아 놓는다. 채널 객체는 주기적으로 입력 버퍼들의 데이타 프레임들을 인터리빙하여 하나의 데이타 스트림으로 변환하여 전달한다. 채널 객체의 주기적인 동작을 위해 기본적으로 Master clock을 이용한다. (그림13)에 오디오 스트림과 비디오 스트림의 채널 객체를 통과하는 시간이 나타나 있다. 각 그래프의 X축은 데이타 프레임의 순서를 나타내고 Y축은 millisecond를 나타낸다. (그림 13)의 (a)는 채널 객체를 Master clock을 이용하여 동작시켰을 때 오디오 스트림이 채널 객체에 입력되는 시간과 인터리브되어 출력되는 사이의 시간 차를 나타낸 것이고 (c)는 비디오 스트림의 경우를 나타낸다.

채널 객체도 Source 객체와 같이 새로운 스트림의 출발점이 되므로 4.2.1절과 4.2.2절에 기술한 것처럼 Master clock을 이용하지 않고 입력되는 스트림에 동작을 시킬 수 도 있을 것이다. 본 논문에서 사용한 비디오 데이타는 단위 시간당 데이타의 발생량이 일정하지 않으나 오디오는 단위 시간당 데이타의 발생량이 일정하므로 채널 객체로 오디오 데이타 프레임 도착하는 것을 채널 객체를 동작시키는 clock과 같이 이용할 수 도 있을 것이다. (그림 13)의 (b)는 오디오 데이타 프레임의 도착을 채널 객체의 동작으로 연결한 경우의 채널 객체를 통과하는 지연을 나타낸 것이고 (d)는 비디오 데이타 프레임이 채널 객체를 통과하는 지연을 나타낸 것이다. (b)의 그래프에 나타난 것처럼 오디오 스트림의 경우는 지연은 대폭 감소함을 알 수

있다. (d)의 비디오 경우는 오디오처럼 지연이 대폭 감소하지는 않음을 보여 준다. 이는 채널을 동작시키는 스트림의 경우 채널을 통과하는 자신의 스트림의 지연을 줄이기는 하지만 채널에 연결된 다른 스트림의 지연 축소에는 큰 영향을 주지 않음을 의미한다. 그러므로 주기적 미디어 스트림과 비주기적 미디어 스트림을 접합하여 하나의 스트림을 생성하는 경우 Master clock을 이용하는 스트림 동작에 비해 주기적 미디어 스트림에 동작하게 하는 방식이 두 입력 스트림 모두의 지연을 균등하게 축소한다고 할 수는 없을 것이다. 그러나 Master clock을 이용하는 방식에서는 Master clock이 동작하는 시각과 오디오 스트림이 동작하는 시각 그리고 비디오 스트림이 동작하는 시각들이 모두 다를 수 있다. 즉 오디오 비디오 스트림 모두가 시간축 상에서 정확하게 접합되지 않고 Master clock의 주기내에서 가변적일 수 있다. 이에 반해 오디오 스트림에 동작 주기를 맞추는 방법에서는 오디오 스트림의 주기내에서 있을 수 있는 비디오 스트림의 프레임들을 접합하므로 오디오 비디오 스트림의 정확도가 Master clock의 경우보다 높아 진다고 할 수 있을 것이다. 그러므로 동기화의 측면에서는 오디오 스트림의 동작 주기에 Master clock을 이용하는 것보다 유리하다고 할 수 있을 것이다. 이는 두 입력 스트림의 데이타 발생 주기성과 데이타의 발생 시점들이 다르고 데이타의 발생량이 다르기 때문으로 분석된다.




(그림 13) 인터리빙 시 오디오, 비디오 스트림의 지연
(Fig. 13) Delay of audio and video stream for interleaving

## 5. 결 론

MuX는 여러 가지 멀티미디어 응용 프로그램을 쉽게 할 수 있도록 멀티미디어 스트림 처리 쉽게 할 수 있는 메카니즘을 제공한다. 본 논문에서는 Windows NT 운영체제상에서 MuX 모델의 기본 처리 요소들과 동작 메카니즘을 기술하였고 이를 이용한 영상 회의 시스템의 설계와 영상 회의를 위한 하드웨어 플랫폼의 구조를 설명하였다.

영상 회의 시스템을 구현하는데 있어 오디오 비디오 스트림이 시작되는 입력부의 스트림의 동작을 오디오, 비디오 데이타 프레임이 준비되었을 때 동작하도록 하는 것이 시간 지연을 줄일 수 있음을 보였다. 그리고 오디오 비디오 스트림을 접합시키는 채널 객체에서는 오디오 스트림의 주기에 동작되도록 한 경우 오디오 스트림의 전송 지연은 크게 줄일 수 있지만 비디오 스트림에는 크게 영향을 주지 않으며 동기화의 측면에서 Master clock을 이용한 경우보다 유리함을 보였다.

본 논문에서 제시한 영상 회의 모델에서는 독립적인 여러 쓰레드들이 수행된다. 원활한 영상 회의를 위해서는 기존 아날로그 방식의 영상 회의에서보다 늘어 날 수 있는 오디오와 비디오 데이타들의 지연과 지터링을 줄여야 한다. 본 논문에서 제시하는 모델에서는 오디오, 비디오 등 독립적인 여러 쓰레드들이 각각 정해진 시간 내에서 동시에 수행되어야 한다. 지연과 지터링의 축소를 위해 이들 쓰레드들이 수행될 수 있도록 컴퓨터 시스템의 하드웨어 자원들과 운영체제가 지원해 주어야 한다. 그리고 이들 컴퓨터 시스템을 연결하는 통신망의 안정적이고 빠른 전송 속도를 보장하여야 할 것이다.

## 참 고 문 헌


[1] Doohyun Kim, SooHyoung Oh, JinKyung Hwang, YoungHwan Lim and Earl Rennison, An Integration and Synchronization Model for Audio, Video and Time-based Graphics, The 2nd Pacific Rim Conference on Articifial Intelligence, Seoul, pp. 1120-1123.

[2] Earl Rennison, Rusti Baker, Doohyun Kim, Young-Hwan Lim, MuX:An X Co-Existent, Time-Based Multimedia I/O Server, The X Resource 1, Winter 1992, pp. 213-233.

[3] Rusti Baker, Alan Downing, Kate Finn, Earl Rennison, Doohyun Kim, and YoungHwan Lim, Multimedia Processing Model for a Distributed Multimedia I/O Systems, NOSSDAV, 1993, pp. 233-239.

[4] Helen Custer, Inside Windows NT, Microsoft Press, 1992, pp. 241-283.

[5] 전윤호, 송동호, 박치항, DLL에 의한 멀티미디어 데이타 처리 객체의 구현, 한국정보과학회 추계 학술대회, 1993, pp. 455-458.

[6] 김 두현 외, MuX V1 R2 Tutorial, '95 MuX사용자그룹 워크샵, 한국전자통신연구소, 1995. 9. 13.

[7] 김 두현 외, MuX V1 R2 API Reference Manual, '95 MuX사용자그룹 워크샵, 한국전자통신연구소, 1995. 9. 13.

[8] 이경희외, Windows NT용 MPEG오디오 디바이스 드라이버의 구현, 전자공학회 추계 학술대회, 1994, pp. 1080-1083.

[9] 이경희외, PCI버스상에서 Windows NT용 오디오 디바이스 드라이버의 구현, 정보처리응용학회 추계학술대회, 1994, pp. 399-402.

[10] Object Management Group, The Common Architecture Request Broker:Architecture and Specification, OMG Document Number 91.12.1 Revision 1.1

[11] Interactive Multimedia Association, Multimedia System Services, Version 1.0, Contributed by Hewlett-Packard Company, IBM Inc., and SunSoft Inc., June 1, 1993.

[12] ITU-Telecommunication Standardization Sector, T.GAT-Proposed draft Recommandatuin for Generic Application Templete for T.120 Compliant Applications, Geneva, 16-25 November 1993.

[13] Michael Altenhofen, Joachim Schaper, and Susan Thomas, "Then BERKOM Multimedia Teleservices," Proc. of Second International Workshop, IWACA '94, Heidelberg, Germany, September

1994, pp. 237-250.

[14] D. Shepherd, G. Coulson, N. Davies and F. Garcia, Queue Monitoring A Delay Jitter Management Policy, Fourth Intl. Workshop on Network and Operating System Support for Digital Audio and Video, Lancaster, UK, November 1993.

[15] 유상신 외 2, 지연에 민감한 멀티미디어 응용을 위한 재생 동기화 메카니즘, 전자공학회논문지 제33권 A편 제4호, 대한전자공학회, 1996년 4월, pp 57-67.

[16] W. Bunn, Multimedia in interconnected LAN systems, BT Technol J Vol 13 No 4, October 1995, pp 124-126.

이 경 희

1990년 경북대학교 컴퓨터공학과 졸업(공학사)
1992년 경북대학교 대학원 컴퓨터공학과 졸업(공학석사)
1992년~현재 한국전자통신연구원 연구원
관심분야:분산멀티미디어, 가상현실, 실시간 미디어처리

김 두 현

1987년 한국과학기술원 전산학 졸업(이학석사)
1985년 서울대학교 컴퓨터공학과 졸업(공학사)
1987년~현재 한국전자통신연구원, 선임연구원
1991년~1993년 스탠포드연구소 객원연구원
관심분야:분산 멀티미디어 처리, 멀티미디어 운영체제, 멀티미디어 통신프로토콜

강 민 규

1985년 경기대학교 전산학과 졸업(공학사)
1987년 중앙대학교 컴퓨터공학과 졸업(공학석사)
1987년~현재 한국전자통신연구원 선임연구원
관심분야:컴퓨터통신, 멀티미디어

정 찬 근

1979년 항공대학교 컴퓨터공학과 졸업(공학사)
1981년 서울대학교 전자공학과 졸업(공학석사)
1983년~현재 한국전자통신연구소 책임연구원 분산멀티미디어연구실 실장
관심분야:분산 멀티미디어, 고속 통신 프로토콜