

컴퓨터 영상회의 시스템을 위한 분산형과 집중형 스트림 연결 구조 비교

이 경 희[†] · 김 두 현[†] · 임 헌 규^{††} · 임 영 환^{†††}

요 약

MuX서버는 다양한 멀티미디어 응용 프로그램 개발을 지원하기 위하여 멀티미디어 스트림의 생성, 복제, 분할, 합성, 집합등의 멀티미디어 입출력을 위한 객체지향적이고 일관적인 인터페이스를 제공한다. 본 논문에서는 MuX의 요소객체를 이용하여 컴퓨터 영상회의 프로그램을 개발할 경우에 가능한 분산형과 집중형 연결 구조에 대하여 지연, 동기 등 멀티미디어 관련 성능에 따른 장단점을 비교 분석한다.

Comparisons between Distributed Connections and Centralized Connections of Multimedia Streams for Computer-based Audio-Video Teleconferences

Kyung Hee Lee[†] · Doohyun Kim[†] · Hun Gyoo Lim^{††} · Young-Hwan Lim^{†††}

ABSTRACT

To support various multimedia applications, MuX server produces object-oriented and consistent interfaces for creation, copying, splitting, mixing and interleaving of streams. In this paper, we describes distributed connection structures and centralized connection structures which can be used in building a teleconferencing system using basic objects of MuX and compares merits and demerits of each structure from the viewpoint of multimedia related performance like delay and synchronization.

1. 서 론

중앙처리장치의 연산능력 향상, 컴퓨터 네트워크의 발전 등으로 인해 문자, 숫자위주로 사용되던 종래의 컴퓨터 시스템의 응용이 음성, 영상, 데이터를 종합적으로 이용하는 멀티미디어 응용 서비스로 발전하고 있다. 특히 컴퓨터 시스템을 이용한 멀티미디어

어 통신용 단말기는 기존의 음성통신만을 위한 단말 장치에서 벗어나 멀티미디어 통신 뿐만 아니라 각종의 멀티미디어 데이터 처리도 가능한 멀티미디어 컴퓨터로 대체되어 갈 전망이다. 이러한 추세 속에서 영상 회의 시스템도 전용 케이블 채널을 이용한 아날로그 통신에서 벗어나 디지털 통신과 데스크탑 컴퓨터를 이용한 컴퓨터 영상 회의 시스템으로 변모하고 있다. 이러한 컴퓨터 영상 회의 시스템은 데스크탑 컴퓨팅 환경 내에 존재함으로써 컴퓨터의 응용 영역을 확장하는 계기가 되었다[1, 2, 3].

그러나 디지털화된 오디오, 비디오 등의 데이터를

[†] 정희원: 한국전자통신연구소 분산멀티미디어연구실

^{††} 정희원: 한국전자통신연구소 시각언어연구실

^{†††} 종신회원: 숭실대학교 정보과학대학 컴퓨터학부

논문접수: 1995년 12월 13일, 심사완료: 1996년 3월 15일

이용하여 영상회의시 기존의 전화에서와 같은 아날로그방식에서 발생되지 않았던 지연(delay), 불연속 등의 문제가 발생하고 회의 참석자의 선택, 다자간 영상회의등의 새로운 요구가 등장하고 있다. 이에 따라 동일 데이터를 회의에 참석하는 다른 영상회의의 단말기들에 전달하여야 하는 데이터 분배 기능과 여러 단말기로 부터 전송되어 오는 데이터를 선택, 혼합등의 기능 그리고 원활한 회의를 위한 회의관리기능, 데이터의 연속성 유지, 데이터 전송 지연의 최소화등의 기능등이 필요하게 되었다.

컴퓨터 시스템의 성능이 발달하였다고는 하나 컴퓨터 시스템의 자원에는 한계가 있으며 컴퓨터 시스템을 연결하는 통신망에도 전송의 한계가 있으므로 컴퓨터를 이용한 영상회의의 구현시 영상회의에 참석하는 참석자의 수와 회의방식등과 같은 제한사항들을 고려하여야 한다. 본 논문에서는 멀티미디어 처리를 위한 모델인 MuX모델을 이용한 컴퓨터 영상회의 시스템 구현 방법, 영상회의의 구현시 고려하여야 할 사항들과 영상회의의 분산 스트림 연결 구조들에 대해 기술한다.

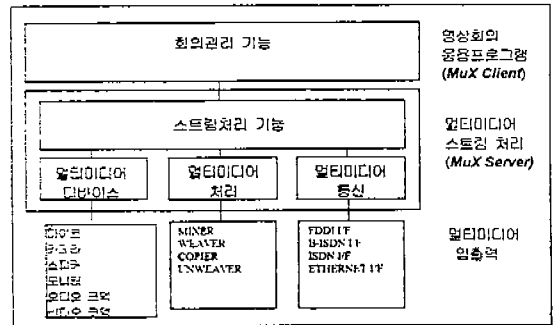
본 논문의 구성은 제 2장에서 멀티미디어 처리 서버인 MuX의 기본 객체와 상속관계를 정의하고, 이들을 이용하여 참석자의 이참석에 따른 동적인 미디어 흐름의 구조 변화가 가능한 다자간 멀티미디어 영상회의의 구조를 구현하는 방법에 대하여 설명하고, 제 3장에서는 이러한 객체를 이용한 다자간 영상회의의 프로그램 구성 방법에 대하여 논한다. 제 4장에서는 영상회의의 시스템구현시 고려해야할 사항과 MuX를 이용한 가능한 영상회의의 시스템의 구조를 비교하였고 제 5장에서 결론을 기술하였다.

2. 멀티미디어 데이터 처리 모델

본 장에서는 분산 멀티미디어 처리기 MuX의 멀티미디어 데이터 처리 모델(이하 MuX모델)에 대하여 설명한다. MuX를 이용한 영상회의의 시스템 설계의 접근 방법은 (그림 1)에 나타낸 바와 같이 클라이언트 서버 개념을 추구한다. 참석자의 관리 및 인증 그리고 사용자 인터페이스 부분은 MuX의 클라이언트로서 존재하고 실질적인 음성 및 영상 데이터는 MuX의 서버에 의하여 처리 된다. 특히 MuX 서버의 스트

림기능 부분은 각종의 멀티미디어 디바이스, 멀티미디어 프로세서, 멀티미디어 통신 인터페이스 등을 연결하는 다중의 스트림 객체를 병렬적으로 처리함과 아울러 다중 스트림의 분배 및 합성, 그리고 다중 스트림의 동적 접속 등의 기능을 제공함으로써 영상회의시 참석자의 동적 이참석에 따른 스트림의 구조 변화 또한 동적으로 유지될 수 있도록 한다.

본 방식은 ITU-T T.120 시리즈에서 GCC와 MCS를 분리하는 접근 방법과 유사하나 클라이언트 서버 개념의 접근 방법을 추구하여 영상회의의 만을 위한 수직적인 시스템을 구축하지 않고 다양한 응용 프로그램을 클라이언트로서 일관성있게 개발할 수 있도록 한다. 본 논문의 접근 방법에 전주어 볼 때 ITU-T의 MCS 기능은 MuX 서버의 일부 기능이 되고 GCC는 영상회의 및 공동작업 관련 기능으로서 서버를 사용하는 클라이언트 프로그램으로 볼 수 있다.



(그림 1) 영상회의 시스템의 개념도
(Fig. 1) The concept of teleconferencing system

2.1 미디어와 프레임

본 논문에서 취급하는 미디어는 주로 컴퓨터 영상회의에서 사용되는 실시간 미디어인 비디오와 오디오로 제한한다. 실시간 미디어의 특징은 재생이 불가능한 물론이고 프레임율(frame rate)이나 샘플링율(sampling rate) 등 QoS(Quality of Service)가 실시간 입출력 단말장치의 성능에 일차적으로 좌우된다.

프레임은 시간축 상에서 임의의 시간간격에 해당되는 미디어 데이터를 포함하고 있는 구조체로서, 미디어의 종류나 임의의 시점의 QoS에 따라 그 크기가 가변적일 수도 있다. 본 논문에서 논하는 프레임은 입출

력시 사용자가 느끼는 감각적 단위와 다를 수도 있다. 예를 들어 비디오의 경우 한 프레임은 사용자가 시각적으로 느끼는 정지화면 한장을 한 프레임으로 삼을 수 있으나 압축방법에 따라서는 그렇지 않을 수도 있다. 또한 오디오의 경우에는 기본적으로 프레임 개념이 없기 때문에 가상의 프레임(만들어야 할 경우도 있다. 따라서 프레임은 연속 미디어의 컴퓨터 처리를 위하여 사용하는 내부 구조체이다.

모든 프레임은 입력단말장치에 의하여 부여된 시간표지(time tag)를 갖고 있다고 가정한다. 시간 표지에 기록된 시각은 SMPTE 등 입력장치의 실시간일 수도 있고 일련번호를 나타내는 정수일 수도 있으나 이들은 프레임들과 시작시간만 알고 있으면 서로 변환이 가능한 것이기 때문에 본 모델에서는 프레임들과 시작시간을 알고 있다는 가정하에 일련번호를 나타내는 정수 시간 표지를 사용한다.

2.1.1 완전주기적 연속 미디어

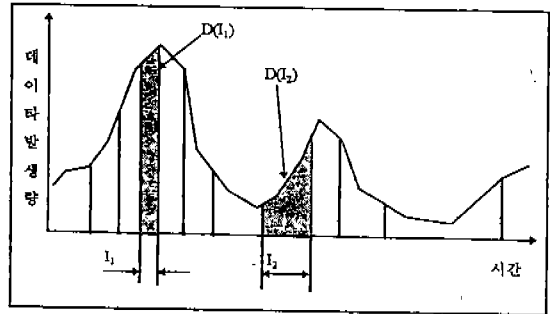
완전주기적 연속 미디어의 특징은 일정량 또는 가변량의 데이터가 일정한 주기로 발생된다는 점이다. 이러한 미디어는 각종 디지털izer에 의하여 일정한 프레임율로 샘플링되는 경우 또는 데이터 모델에 따라 합성(synthesize) 또는 생성되어지는 경우에 속한다. 스트림은 완전주기적 연속 미디어의 흐름을 실현함에 있어서 데이터 발생 당시부터 내재되어 있는 주기를 최대한 유지시켜 주어야 한다.

2.1.2 부분주기적 연속 미디어

부분주기적 연속 미디어의 특징은 주기적 연속 미디어와 같이 일정량 또는 가변량의 데이터가 지속적으로 발생되기는 하나 그 주기가 약간의 가변성을 갖는 경우이다. 이러한 미디어는 주로 차분적 코딩(differential coding)을 하는 경우 코덱 장치가 데이터를 발생시키는 시간 간격이 일정하지 않은 경우에 속한다. 스트림은 부분주기적 연속 미디어의 흐름을 실현함에 있어서 데이터 발생 당시부터 내재되어 있는 데이터 간의 시간 간격을 최대한 유지시켜 주어야 한다.

현재 MuX 서버는 윈도우 NT 3.5에서 H.261 압축형식의 비디오와 8비트 샘플 크기, 11.025Khz 샘플링율의 PCM(pulse code modulation) 형식의 오디오를 사용하고 있다. 따라서 비디오는 부분주기적 연속 미

디어이고 오디오는 완전주기적 연속 미디어이다. 이를 보다 자세히 설명하면 H.261은 기본적으로 차분정보를 이용한 압축 방식이므로 데이터 발생량이 시간에 따라 가변적이다. 또한 H.261 압축 디바이스가 압축한 데이터를 매번 메모리로 전달하는 크기가 고정되어 있으므로 압축 디바이스가 중앙처리장치에 인터럽트를 보내주는 주기도 가변적이 된다. 이러한 상황이 (그림 2)에 예시되어 있다. 즉 그림에서 나타냈듯이 시간 간격 I 동안의 데이터 발생량을 D(I)라 할 때 $I_1 \neq I_2$ 인 반면 $D(I_1) = D(I_2)$ 이다. 한편 PCM 방식의 오디오는 차분정보를 이용하지 않으므로 $I_1 = I_2$ 임과 아울러 $D(I_1) = D(I_2)$ 이 된다.



(그림 2) 부분주기적 연속 미디어의 데이터 발생 예시 (Fig. 2) A data generation example of partially periodic continuous media

2.1.3 이벤트형 미디어

이벤트형 미디어의 특징은 주로 마우스 입력과 같이 비주기적이면서 불연속적이라는 점이다. 엄밀히 말해서 이벤트형 미디어는 멀티미디어 본연의 데이터는 아니나 멀티미디어와 함께 사용되는 데이터이므로 스트림이 처리하여야 할 대상에 속한다. 스트림은 이벤트형 미디어를 처리함에 있어서 이벤트 처리의 실시간성을 최대한 만족시켜 주어야 한다.

2.2 스트림(Stream)

스트림은 단일 미디어 흐름으로서 미디어 데이터 입출력의 가장 기본적인 구조이다. 스트림이 처리하는 미디어의 종류는 크게 3가지 즉 1)완전주기적 연속 미디어(completely periodic continuous media), 2)부분주기적 연속 미디어(partially periodic continuous

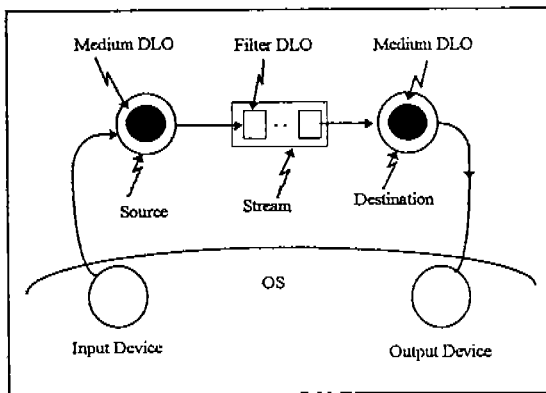
media), 3) 이벤트형 미디어 등이 있을 수 있다. 미디어 스트림은 상기의 세가지 미디어에 대하여 화일, 디바이스, 네트워크, IPC, 분배기(Copier), 접합기(Weaver), 분리기(Unweaver), 합성기(Mixer)와의 입출력 관계를 맺어 줌으로서 미디어의 흐름이 형성되도록 한다.

2.2.1 스트림 기능

- 화일, 디바이스, 통신 등 입출력 장치와의 완전주기적 연속 미디어, 부분주기적 연속 미디어, 이벤트형 미디어에 대한 입출력 기능을 수행한다.
- 미디어 데이터를 분배 또는 합성하기 위한 Mixer, Weaver, Copier, Unweaver 등과 접속하고 이들과 데이터를 입출력하는 기능을 수행한다.
- 미디어 데이터를 입력하여 출력 시키는 과정에서 데이터에 대한 필터링 기능을 수행한다.
- 다른 스트림과의 동기화를 위해 스트림으로 들어오는 데이터의 시간표지를 기록한다.

2.2.2 스트림 처리 요소

스트림은 상기의 서비스들을 제공하기 위하여 Stream, Source, Destination, Filter, Medium 등의 객체가 필요하며 이들의 관계가 (그림 3)에 나타나 있다.



(그림 3) 스트림 개념도
(Fig. 3) The concept of a stream

2.2.2.1 Medium 객체

Medium 객체는 각종 멀티미디어 디바이스, 화일

포맷, 통신 프로토콜과의 입출력을 일관성 있게 하기 위하여 사용되는 객체로서 DLO(dynamic linking object)로 구현되어 진다. 미디어용 DLO는 각종 포맷의 멀티미디어 화일 및 멀티미디어 디바이스와의 인터페이스를 위한 DLO와, 통신을 위한 DLO 등이 있는데, 대표적인 예로 JPEG 비디오 화일 입출력 DLO, H.261 비디오 입력 DLO, UDP 통신용 DLO, TCP 통신용 DLO 등을 들 수 있다.

2.2.2.2 Source 객체

Source 객체는 입력용 Medium을 소유하고 있는 객체로서 스트림에 흐르는 단위 데이터인 프레임의 시발점이 된다. 따라서 Source 객체에서는 Source 객체가 소유한 DLO를 통하여 각종 입력 장치로부터 프레임 단위의 정보가 읽혀 들어간다. 따라서 Source 객체에서는 Medium에서 프레임을 읽어 Destination으로 전달하는 주기적 동작을 위한 쓰레드의 출발점이며, Source 객체는 프레임의 흐름을 제어하기 위한 Play(), Stop(), Pause(), Jump(), SetFrameRate(), SetSpeed(), SetDuration() 등의 함수를 제공한다.

2.2.2.3 Destination 객체

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

2.2.2.4 Stream 객체

Stream 객체는 (그림 3)과 같이 Source 객체와 Destination 객체의 연결로서, Source 객체가 Medium 객체로부터 읽어들이 생성된 프레임을 Destination 객체가 소유하고 있는 출력용 Medium 객체로 전달하는 기능을 수행한다. 특히 Stream 객체는 이러한 전달의 과정에 있어서 자신에게 등록된 Filter 객체에 매 프레임을 적용함으로써 Stream 객체를 통과하는 프레임에 적절한 변화를 줄 수 있도록 해준다.

2.2.2.5 Filter 객체

필터는 DLO로 구현되는데, Stream 객체가 생성된 후 응용 프로그래머가 원하는 필터의 DLO 명칭을 Stream 객체에 API를 통하여 지정할 수 있다.

2.3 채널(Channel)

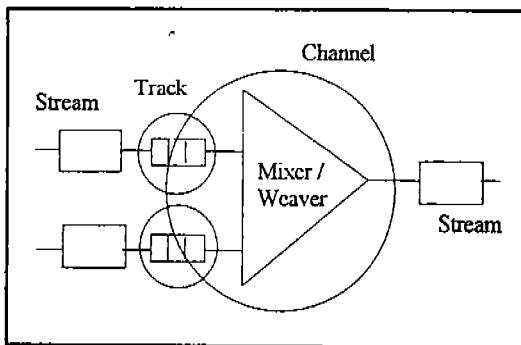
스트림이 미디어 입출력의 가장 기본적인 독립구조라 할 수 있다면, 채널은 여러개의 독립적인 스트림을 하나의 스트림으로 합치는 작업을 한다.

2.3.1 채널의 기능

- 스트림 시작시점 및 종료시점 동기화(end-point synchronization): Channel 객체에 등록된 Track 이나 Source에 지정된 시작 시점과 종료 시점에 근거하여 Channel 자체의 시간 축 상에서 해당 시점에 시작 및 종료를 시킨다.
- 동종 미디어 스트림의 통합: 예를 들어 여러 비디오 신호를 혼합하여 하나의 그래픽 표면에 매핑하거나, 다중 오디오 스트림을 하나의 스트림으로 혼합하는 등의 기능을 수행한다.
- 이종 미디어 스트림의 접합: 예를 들어 오디오 스트림과 비디오 스트림을 접합(interleave)하여 단일 스트림으로 만들어주는 기능을 수행한다.
- 스트림 간의 동기화(interstream synchronization): 상기 항목의 통합 및 접합 기능을 수행하면서 프레임에 부착된 시간표지를 이용하여 프레임 간의 동기화를 시킨다.

2.3.2 채널 처리 요소

채널은 상기의 서비스들을 제공하기 위하여 트랙 객체, 채널객체, 합성기(Mixer), 접합기(Weaver) 등의 요소가 필요하며 이들의 관계가 (그림 4)에 나타나 있다.



(그림 4) 스트림 합성 및 접합 개념도

(Fig. 4) The concept of stream mixing and stream interleaving

2.3.2.1 트랙 객체

트랙 객체는 일종의 Destination 객체로서 Stream 객체를 통하여 전달되어 오는 프레임을 큐에 입력하고, 채널의 요구에 따라 큐로부터 프레임을 인출하여 주는 역할을 한다. 트랙 객체가 사용하는 큐는 프레임에 부착된 시간표지를 이용하는 시간큐(time-based queue)로서 채널이 dequeue시에 채널 객체의 시점을 파라미터로 넘겨주면 시간큐는 그 시간에 가장 근사한 시간 표지를 갖는 프레임을 dequeue 시키고 그 이전시간의 프레임 들을 소멸시킨다. 이러한 시간큐의 기능에 의해서 채널 객체가 프레임의 합성이나 접합시에 동기화 기능까지 수행할 수 있게 한다.

2.3.2.2 채널 객체

채널 객체는 일종의 Source 객체로서 자신에게 접속된 트랙 객체들로부터 프레임을 인출하여 이들을 합성기(Mixer) 또는 접합기(Weaver)를 이용하여 합성 또는 접합하여 출력측 Stream 객체에게 전달하는 기능을 한다. 합성기나 접합기는 DLO로 구현되어 있는데 채널의 생성시 파라미터로 지정하게 되어 있다. 채널의 생성시 어느 DLO를 지정하느냐에 따라 트랙 객체로부터 인출한 프레임들의 데이터 자체를 믹싱하여 한 프레임으로 만드는 합성을 할 수도 있고, 프레임들을 단지 인터리브시키는 접합을 할 수도 있다. 믹싱을 시키는 경우는 주로 음악과 음성 등과 같이 동종 미디어의 스트림이 트랙을 입력으로 받는 경우이고, 인터리브시키는 경우는 주로 비디오와 오디오 등 이종 미디어일지라도 동기정보를 유지한채 네트워크 전송하기 위한 경우에 주로 사용한다. 또한 채널 객체가 수행 중이라도 트랙 객체의 접속이 가능하다. 즉 예를 들어 오디오 스트림 하나와 음성 스트림 하나를 믹싱하고 있는 도중에라도 세번째 오디오 스트림이 접속되는 것이 가능한데 세번째 스트림이 접속되면 접속되는 순간 부터 세개의 스트림이 합성되게 된다.

2.3.2.3 합성기(Mixer)

합성기의 주요 기능은 각 트랙으로부터 채널 시점에 가장 근사한 시간표지를 갖는 프레임을 인출하여 하나의 합성된 프레임을 생성하고 이를 출력 스트림으로 출력시키는 기능을 한다. 또한 합성기 DLO를

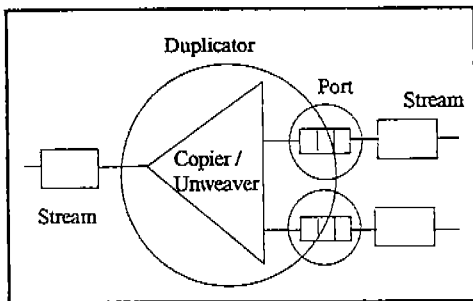
만들기 여부에 따라서는 몇 개의 트랙들이 합성되어 내부적인 스트림을 형성하고 이러한 내부 스트림 몇 개가 다시 합성되어 새로운 스트림을 형성하는 등 계층적인 형태도 가능하다.

2.3.2.4 접합기(Weaver)

스트림 통합과 스트림간 동기화는 합성기에 의하여 주로 이루어 지지만 때로는 합성을 하지않고 동기를 맞추어 단순히 접합만을 하여 인터리브 시켜야할 경우가 있다. 이를 위한 것이 접합기이다. 접합기는 합성기와 마찬가지로 각 트랙으로부터 채널 시점에 가장 근사한 시간표지를 갖는 프레임을 인출하여 복합적인 구조의 하나의 접합 프레임을 생성하고 이를 출력 스트림으로 출력시키는 기능을 수행한다. 이렇게 만들어진 복합 프레임은 Weaver의 짝인 Unweave에 의하여 다시 단위의 프레임으로 분리된다. Unweave는 Duplicator를 통하여 사용할 수 있다.

2.4 Duplicator

Duplicator는 채널 객체와 대칭되는 개념으로서 (그림 5)와 같이 하나의 스트림을 여러개의 스트림으로 복제하거나 접합되어 있는 프레임을 분리하는 기능을 한다.



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

2.4.1 Duplicator의 기능

- 입력 스트림을 통하여 전달되어 오는 프레임이 하나 이상의 출력 스트림으로 복제하여 분배하는 기능을 제공한다.

- 입력 스트림을 통하여 전달되어 오는 프레임이 채널의 접합기(Weaver)에 의하여 여러 개의 프레임을 내포하고 있는 경우 이를 분리하여 각 해당 출력 스트림으로 분리하는 기능을 제공한다.

2.4.2 Duplicator

2.4.2.1 Duplicator 객체

Duplicator 객체는 일종의 Destination 객체로 스트림 객체를 통하여 전달되어 오는 프레임을 복제기(Copier)나 분리기(Unweaver)를 통하여 여러개의 프레임으로 복제 또는 분리하여 이들을 자신에게 접속된 포트객체에게 전달해주는 기능을 한다. 복제기나 분리기는 DLO로 구현되어 있는데 Duplicator의 생성자를 호출할 때 반드시 DLO의 이름을 파라미터로 제공하도록 되어 있다. 어느 DLO를 지정하느냐에 따라 프레임을 복제할 수도 있고 인터리브된 프레임을 분리할 수도 있다.

2.4.2.2 포트(Port) 객체

포트 객체는 일종의 Source 객체로서 Duplicator 객체로부터 프레임을 전달받아 이를 자신의 내부에 갖고 있는 큐에 넣고, 출력 스트림이 큐로부터 프레임을 인출해 갈 수 있도록 하는 기능을 한다. 포트 객체는 Duplicator객체가 수행 중이라도 Duplicator 객체에 접속이 가능하다. 단 수행 중 접속된 포트에는 복제 또는 분리된 프레임이 접속된 순간부터 포트객체로 출력되기 시작한다.

2.4.2.3 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 MPEG과 같은 압축된 스트림으로 출력할 수도 있다.

2.4.2.4 분리기(Unweaver)

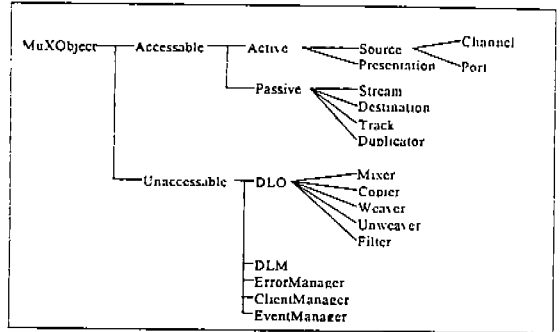
스트림 분리는 입력되는 스트림이 하나 이상의 미디어를 접합(interleaving)하고 있을 경우 각 접합된 미디어들을 분리하여 각각을 해당 출력측 스트림으로

출력한다. 일례로 영상회의시 사정에 의하여 PCM방식의 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.

2.5 상속체계 및 컨커런시

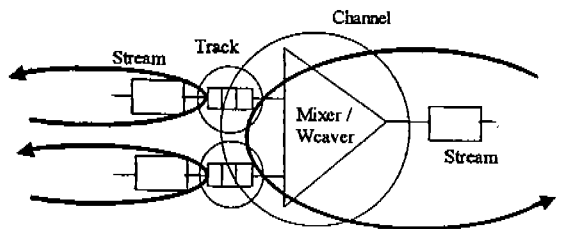
2.4절까지 본 논문에서 제시한 스트림, 채널, Duplicator 등 멀티미디어 요소 객체는 개별적으로 동작이 가능하지만 다양한 응용 프로그램을 만들어내기 위하여 이러한 요소 객체들이 연결될 수 있다. 이러한 연결은 우선 (그림 6)에 나타낸 바와 같이 상기의 요소 객체가 서로 간에 상속체계를 가지므로써 가능해진다. 상속체계의 가장 상위 클래스는 MuXObject 클래스이다. MuXObject 클래스는 상기에서 설명된 모든 객체에 상속되며 객체관리에 필요한 기능을 수행한다. Accessable 클래스는 MuX 서버내의 모든 객체가 네트워크 환경에서 투명한 접근이 가능하도록 객체의 생성과 소멸시 이름관리 및 디렉토리 관리 부분에 해당 객체를 등록 또는 삭제시키는 관리기능을 한다. MuX API를 통하여 응용 프로그램이 사용할 수 있는 모든 클래스는 바로 이 Accessable 클래스를 상속 받는다. Accessable 클래스 중에서 Active 클래스는 동적인 역할을 하는 객체의 클래스로서 각자 쓰레드 하나씩을 갖고 미디어 처리에 필요한 단위 작업을 동적으로 수행한다. Active 클래스에는 상기에서 설명한 클래스 중 Source클래스, 채널 클래스, 프리젠테이션 클래스 등이 있다. 반면 Passive 클래스는 Active 클래스에 접속되어 접속된 Active 클래스를 수행하는 쓰레드의 지배하에 수행된다. Unaccessable 클래스는 클라이언트 프로그램이 접근할 수 없는 서버 내부 클래스로서 주로 멀티플렉싱 기능을 만들기 위한 DLO의 서브클래스인 Medium, Mixer, Copier, Weaver, Unweaver 등이 이에 해당된다. 또한 DLO를 관리하기 위한 DLM 클래스와 ErrorManager, EventManager, ClientManager 등도 모두 Unaccessable 클래스에 속한다.

데이터 플로우는 각 Active 객체의 쓰레드 구동에 의하여 일어난다. 특히 Source 객체는 스트림 중에서 유일한 Active 객체로서 스트림이 개념적 뿐만아니라 실제의 동작에서도 컨커런시의 기본 단위 객체라 할

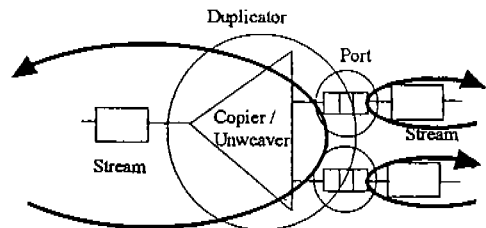


(그림 6) 클래스 상속체계 (Fig. 6) Class hierarchy

수 있다. 이뿐만아니라 Source 객체는 다른 Active 객체인 Channel과 Port에 상속되어 이들 역시 동시에 동작될 수 있도록 한다. 따라서 전송된 스트림, 채널, Duplicator와 관련된 (그림 4), (그림 5)의 개념도는 사실상 (그림 7)와 (그림 8)에 굵은 화살표로 나타낸 바와 같이 여러 개의 동시 동작 스트림(concurrent stream)이 접속되어 있는 것이다.



(그림 7) 채널관련 동시 동작 스트림의 접속 (Fig. 7) A connection of concurrent stream related to a channel



(그림 8) Duplicator 관련 동시 스트림의 접속 (Fig. 8) A connection of concurrent streams related to a duplicator

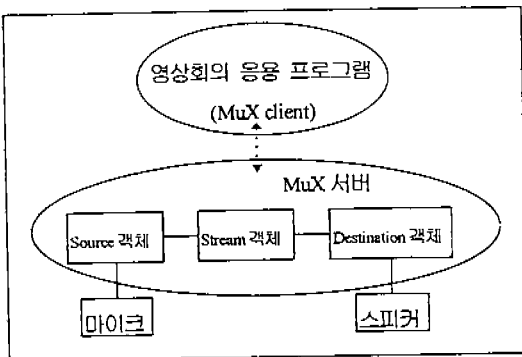
3. 멀티미디어 입출력

본 장에서는 영상회의를 위한 응용 프로그램의 구성과 MuX서버와의 관계에 대해 기술한다. 본 논문에서 제시하는 모델은 각 시스템마다 MuX 서버를 두고 영상회의를 위한 응용 프로그램에서 멀티미디어 입출력 요구를 MuX 서버에 전달, 처리하게 한다.

3.1 MuX 서버

MuX는 멀티미디어 입출력을 담당하는 서버이다. MuX 클라이언트 프로그램에서 멀티미디어 입출력을 위한 객체들을 직접 생성, 연결하는 것은 아니다. MuX 클라이언트 프로그램은 2장에 기술된 Source, Destination, Medium, Stream, Channel등과 같은 멀티미디어 입출력을 위한 객체들의 생성, 연결을 MuX 서버에게 요청만 한다. MuX 서버는 MuX 클라이언트 프로그램의 요구를 받아 실제적으로 각 객체들을 생성, 연결한다. 그러므로 MuX의 클라이언트 프로그램으로 영상회의 프로그램을 작성하는 경우, 영상회의 응용 프로그램에서는 영상회의 소집, 진행, 종료, 관리등의 기능을 수행하고 영상회의에 필요한 오디오, 비디오 데이터를 입력받아 상대방에 전달하거나 상대방으로부터 전달되어 오는 오디오, 비디오 데이터를 출력하는 일은 MuX서버를 이용한다.

MuX서버는 멀티미디어 데이터 스트림을 위한 일관성있는 인터페이스를 제공하므로 멀티미디어 데이터 처리를 필요로 하는 응용 프로그램은 MuX 클라



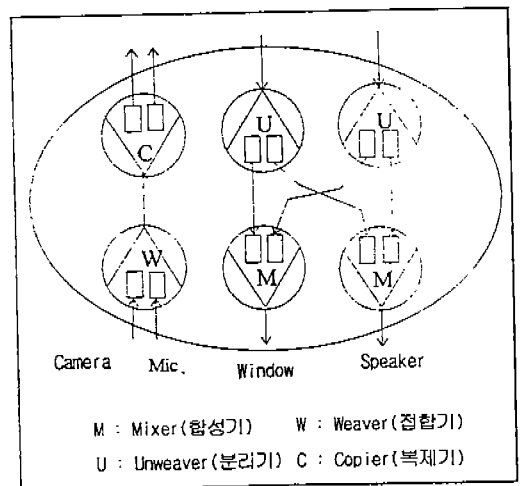
(그림 9) MuX 서버와 클라이언트와의 연결

(Fig. 9) A connection between MuX server and a client

이언트 프로그램으로 MuX의 API(Application Program Interface)를 이용하여 멀티미디어 스트림을 컨트롤할 수 있다.

3.2 MuX 서버내 객체들의 연결 예

본 절에서는 2장에서 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, MuX는 미디엄 DLO를 통하여 이러한 각 멀티미디어 디바이스와 프레임을 입출력한다. 위와 같은 환경에서 3자 오디오 비디오 회의의 경우 한 호스트의 동적 객체와 이들의 연결 관계가 (그림 10)에 나타나 있다. 각 호스트에 구성된 객체들과 연결 구조는 서로 동일하여 각 호스트는 자신의 마이크 입력과 카메라 입력을 접합기를 사용하여 인터리브 시키고 그 결과를 복제기를 사용하여 자신을 제외한 두 상대방에게 전송한다. 반면 두 상대방부터 전달되어 오는 인터리브된 프레임을 먼저 분리기를 사용하여 오디오 프레임과 비디오 프레임으로 분리한 후 각각을 오디오 믹서를 사용하는 채널 객체와 비디오 믹서를 사용하는 채널 객체로 전달 한다. 각 채널 객체는 두 상대방부터 오



(그림 10) 3자회의 시 한 호스트의 MuX 서버내 객체 연결도
(Fig. 10) Object connections in MuX server for 3 person conference at a host

는 프레임을 합성하여 각각 스피커와 디스플레이 윈도우로 출력시킨다.

또한 사용자의 취향에 따라서는 자신의 음성과 자신의 모습을 자신도 듣고 보기를 원하는 경우가 있을 수 있는데 이러한 경우의 구조는 더 복잡하여 자신의 마이크와 카메라로부터 들어오는 프레임을 인터리브를 위한 채널에 입력하기 전에 각각을 먼저 복제하여 음성 프레임은 상대방의 음성을 믹싱하는 채널객체로 입력시키고, 비디오 프레임은 상대방의 비디오를 오버레이시키는 채널 객체로 입력시킨다. 이 경우 음성합성용 채널 객체와 비디오 오버레이용 채널 객체가 각각 3개씩 필요하게 된다.

인터리브된 프레임을 전송하는 이유는 인터리브하지 않을 경우 delay, jitter, skew 등 전송 상의 갖가지 이유로 오디오와 비디오 간의 전송 속도의 상대적인 전송 속도의 차이가 발생하여 수신측에서 오디오와 비디오의 동기가 손상되어 이를 복구하기가 어렵게 되는 경우가 발생하는 것을 방지하기 위함이다.

이와 같이 영상회의와 같은 멀티미디어 응용 프로그램은 MuX서버를 이용하여 Source, Destination, Channel, Duplicator 등의 객체를 조합하여 원하는 데이터를 입출력할 수 있다.

4. 영상 회의

4.1 구조 결정 요소

4.1.1 회의 형태

회의 진행에 있어 모든 참석자들이 동등한 권한을 가지고 참석자가 원하는 시기에 누구나 발언을 할 수 있는 형태와 회의를 주재하는 주재자가 존재하고 발언권을 획득하여 발언을 하는 형태가 있을 수 있다. 전자의 경우는 회의를 진행하는데 별다른 회의 중재 기능이 없는 것으로 구성은 간단하나 여러 참석자가 동시에 발언을 하는 경우 원활한 회의 진행이 어려워질 수 있는 문제가 있고, 후자의 경우에는 회의를 중재하는 기능을 구현해주어야 하나 원활한 회의를 할 수 있다는 장점이 있다.

4.1.2 이참석의 유연성

영상회의를 시작하는 회의 소집방법에는 크게 두 가지가 있다. 첫째는 회의를 소집하는 소집자가 참석

자를 결정, 소집하는 소집자 기반 방식과 회의에 참가를 원하는 멤버들이 회의에 참석하는 형식의 참가자 기반 방식이 있을 수 있다. 그리고 회의가 진행되는 동안 참석자들의 이참석이 자유로운 방식이 있을 수 있고 회의종료시까지 이참석을 불허하는 방식이 있을 수 있다.

회의 이참석이 고정적인 형에는 회의를 시작할 때 이루어진 오디오, 비디오 통신채널등의 연결이 변하지 않으므로 회의 초기시에만 연결 경로를 만들어 놓으면 된다. 그러나 회의 멤버들의 이참석이 자유로운 경우 멤버들의 이참석에 따라 오디오, 비디오, 통신채널들의 연결을 동적으로 매번 조정해 주어야 한다. 이는 영상회의 시스템의 성능에 영향을 미칠 수 있으므로 영상회의 시스템의 성능이 허용하는 범위 내에서 이루어져야 한다. 그러므로 영상회의시 이참석의 유연성은 영상회의 시스템의 성능을 고려하여야 한다.

4.1.3 단말기 성능

영상회의를 위해서 단말기는 기본적으로 오디오와 비디오 데이터를 입출력할 수 있고 이를 네트워크 등의 장치를 통해 다른 단말기로 전송할 수 있어야 한다. 그리고 경우에 따라서는 데이터의 변환, 역변환 혹은 다중화 역다중화등의 기능을 제공하여야 한다.

오디오, 비디오 등의 멀티미디어 데이터를 처리하는 데 단말기가 지원하는 서비스의 질이 각 단말기 상호간에 다를 수 있다. 송신측에서 양질의 데이터를 송신한다고 하더라도 수신측이 이를 소화해낼 능력이 없으면 문제가 될 수 있으므로 단말의 성능차이로 인해 서비스의 질을 조정해 주어야 하는 일도 고려하여야 한다.

또한 성능이 우수한 컴퓨터 시스템이라 하더라도 지원이 유한하기 때문에 이용가능한 자원에는 한계가 있다. 즉 예로 단말기의 성능에 따라 연결을 맺을 수 있는 최대 스트림의 수는 제한될 수 밖에 없다. 그러므로 단말기에서 처리할 수 있는 오디오 입출력 스트림의 수 혹은 비디오 입출력 스트림의 수 등 단말의 성능에 따라 회의에 참석할 수 있는 멤버의 수 등은 제한될 수 있다.

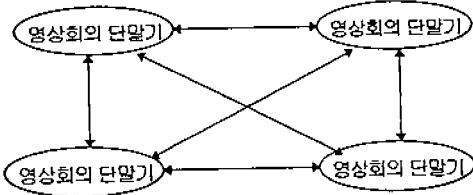
4.1.4 서버의 존재

영상회의를 진행하는데 회의진행을 관리해 주거나

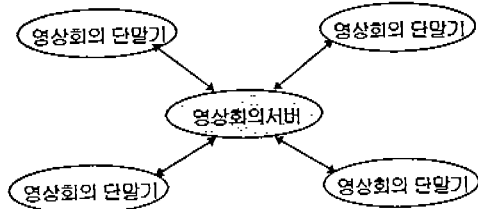
회의에 필요한 여러가지 종류의 데이터를 송수신하는 등의 기능을 각 단말기들이 분산하여 처리하는 분산처리형의 방식이 있을 수 있고 이러한 기능을 통괄하여 주는 서버를 두는 집중형이 있을 수 있다.

분산처리형은 회의 참가중인 단말들과 접속을 만드는 일과 접속 후 데이터를 전송하는 일 등을 각 단말기들이 처리해야 한다. 또 다수의 통신채널을 형성해야 하고 정보의 처리량이 많아 지게 되는 등 단말기의 구조가 복잡해 질 수 있다.

집중형은 각 단말기들이 중앙의 서버에 종속되는 형태로 각 단말기의 구조를 간단히 할 수 있고 회의 중 이참석시 연결된 기존의 접속을 크게 바꾸지 않고서도 새로운 접속을 생성할 수 있다. 그러나 서버에서 많은 일들을 행하기 때문에 서버에 부하가 많이 걸리므로 서버의 연산능력에 의존적이 된다.



(그림 11) 분산 처리형 영상회의의 연결의 예
(Fig. 11) A connection example of distributed teleconference



(그림 12) 집중형 영상회의의 연결의 예
(Fig. 12) A connection example of centralized teleconference

4.1.5 네트워크 구조와 성능

분산된 단말기들을 연결해주는 네트워크의 구성은 링(ring)형, 버스(bus)형, 성(star)형, 나무(tree)형 등의 다양한 방식이 있을 수 있다. 이는 사용하는 망의 종류, 단말기들의 물리적 위치 등에 의존적일 수 있다.

네트워크상에서 이루어지는 회의의 수 혹은 회의에 참석하는 참석자의 수가 증가하면 네트워크에서 전송되어 지는 데이터의 량은 증가한다. 각 단말기의 성능이 우수하여 양질의 오디오, 비디오 데이터를 생성, 처리할 수 있다고 해도 데이터를 전송하는 네트워크의 성능이 이를 지원하지 못 한다면 네트워크에서 병목현상이 생겨 네트워크상에서 이루어지는 영상회의의 수 혹은 회의의 참석자의 수, 영상이나 음성에 대한 서비스의 질은 낮아 질 수 밖에 없다. 그러므로 영상회의의 시스템을 구축하려면 영상회의를 위한 단말기의 성능과 아울러 각 단말기들이 접속되어 있는 네트워크의 성능도 고려해야 한다.

4.2 영상회의의 연결구조 분류

영상회의를 위해 단말기들간의 접속을 만드는 방법으로는 크게 두가지가 있을 수 있다. 그 중 하나는 영상회의를 위한 서버를 두고 각 단말기들은 이 서버와 접속을 생성하는 집중형이고, 다른 하나는 특별한 서버를 두지 않고 상호 필요한 접속을 각 단말기들이 원하는 단말기들과 오디오, 비디오등의 데이터 경로나 콘트롤 신호를 송수신하기 위한 제어 경로를 생성하는 분산형이 있을 수 있다. 분산형은 통신경로의 수가 많으며 각 단말기들의 구조가 복잡해 질 수 있다. 반면 집중형은 통신경로의 수가 분산형보다 적으며 각 단말기의 구조를 간단히 할 수 있으나 서버에 부하가 편중되며 데이터 전송시 시간지연이 생길 수 있다.

그리고 오디오, 비디오 데이터를 전송하는 방식 중 오디오 스트림과 비디오 스트림의 동기를 위해 오디오 스트림과 비디오 스트림을 접합시켜 하나의 스트림을 생성하여 전송하는 접합형의 방법과 오디오 스트림과 비디오 스트림을 독립적으로 전송하는 분리형이 있을 수 있다.

데이터를 전송할 때 데이터를 전달해야 하는 단말기들에 순서적으로 각각 전달하는 방식이 있을 수 있고 같은 데이터를 전송해야하는 단말기들을 하나의 그룹으로 취급하여 그 그룹에 속한 단말기들 전체를 대상으로 한번만 전송하는 멀티캐스팅(multicasting) 방식이 있을 수 있다. 멀티캐스팅 방식은 서버가 존재하는 중앙처리형에도 적용되어 질 수 있고 서버가 없는 분산처리형에도 적용되어 질 수 있다.

4.2.1 분산형 연결

4.2.1.1 분리 분산형

분리 분산형은 오디오, 비디오 데이터 스트림을 결합시키지 않고 독립적인 경로를 통해 연결된 다른 단말기로 전송하는 형태이다. 이 모델은 구현하기가 다른 방식보다는 비교적 쉬운 장점은 있으나 통신을 위한 경로의 수가 많아져 통신망에 부하가 걸린다는 단점과 각 단말기의 구조가 복잡해지는 단점이 있다. 예로 N명의 회의 참석자들이 모두 참석자들의 영상과 음성을 듣고자 하면 $2*N*(N-1)$ 개의 통신경로가 설정되어야 한다. 이와 아울러 오디오와 비디오의 스트림이 각각 독립적인 경로로 전달되므로 오디오 스트림과 비디오 스트림간의 동기화가 복구한계를 초과하여 복구불능에 이르는 문제가 발생할 소지가 있다.

4.2.1.2 결합 분산형

결합 분산형은 오디오, 비디오 데이터 스트림을 결합시켜 연결된 다른 단말기로 전송하는 형태이다. 이 모델은 각 단말기에서 오디오, 비디오 데이터 스트림이 결합된 단일 스트림을 생성하고 오디오, 비디오 데이터 스트림이 결합된 스트림에서 오디오, 비디오 스트림을 분리해내야 하는 문제가 있다. 그러나 오디오, 비디오 스트림의 동기화가 보장되고 통신망의 필요한 경로가 분리 분산형의 경우 보다 절반으로 줄어든다는 장점이 있다.

4.2.1.3 분리 분산 멀티캐스팅형

분리 분산 멀티캐스팅형은 단말기들 사이의 통신 방식으로 멀티캐스팅을 지원하는 프로토콜을 이용함을 의미한다. 분리 멀티캐스팅형 모델은 오디오, 비디오 스트림을 독립적으로 전달하는데 각 단말기마다 통신경로를 설정하여 순서적으로 전달하는 것이 아니라 보내려는 단말기 그룹을 설정하고, 그 그룹에 포함된 단말기들에게 오디오, 비디오 스트림을 한꺼번에 전송하는 방식이다. 멀티캐스팅을 이용하면 통신망의 부하를 줄일 수 있는 장점이 있으나 통신 오류 발생시 이를 복구하는 데 문제가 있을 수 있다. 그리고 분리 분산형에서와 같이 오디오, 비디오 스트림간의 동기화의 문제가 있을 수 있다.

4.2.1.4 결합 분산 멀티캐스팅형

결합 분산 멀티캐스팅형은 분리 멀티캐스팅형과 통신방식에서는 동일하며 전달하는 스트림이 오디오와 비디오가 결합된 스트림이라는 것이 다르다. 그러므로 각 단말기에서는 오디오, 비디오 스트림을 결합하여 하나의 스트림으로 생성하는 기능과 오디오, 비디오 스트림이 결합된 스트림에서 오디오 스트림과 비디오 스트림을 분리해내는 기능을 수행하여야 한다. 그러나 분리형에서 발생할 수 있는 동기화의 문제는 없앨 수 있으나 스트림의 결합 분리시 시간 지연이 생길 수 있다.

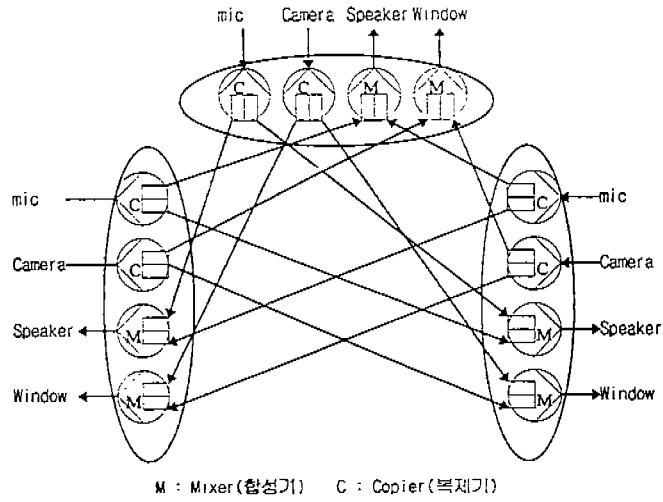
4.2.2 집중형 연결

4.2.2.1 분리 집중형

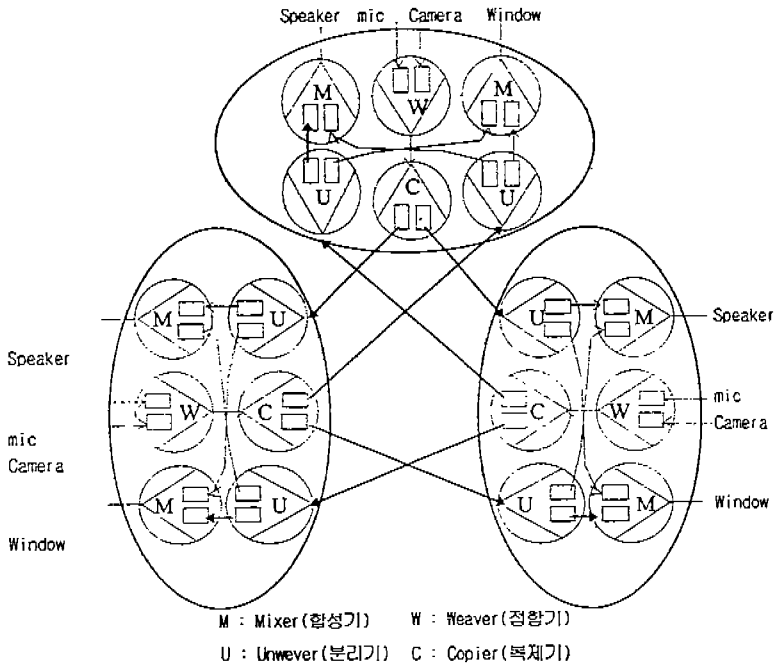
분리 집중형은 각 단말기에서 생성되는 오디오, 비디오 스트림을 각각 영상회의의 서버 시스템에 전달한다. 영상회의의 서버 시스템은 각 단말기들로부터 수신된 오디오, 비디오 스트림들을 혼합하는 등의 가공처리를 하여 각 단말기들에게 필요한 스트림으로 변환한 후 각각 전달하는 방식이다. 각 단말기에서는 영상회의의 서버와 통신을 하여 필요한 스트림을 송수신하면 되므로 통신망상의 연결을 생성하는 부담과 오디오 혹은 비디오 스트림을 믹싱해야 하는 등의 부하를 줄일 수 있으며 전체 통신망의 부하가 줄어드는 장점이 있다. 그러나 영상회의의 서버에서 처리해야 하는 일이 많으므로 영상회의의 서버의 연산능력이 이를 허용해야 한다. 또한 오디오, 비디오 스트림이 독립된 경로로 서버에 전달되므로 오디오, 비디오 스트림간의 동기화에 문제가 있을 수 있으며 스트림의 전달이 서버를 거치게 되므로 분산형보다 스트림의 전달이 지연될 수 있다. 이 모델을 이용하면 새로운 멤버의 추가시 기존의 다른 모든 단말기들과의 통신경로를 설정할 필요가 없이 단지 영상회의의 서버와 접속만 하면 되는 장점이 있다.

4.2.2.2 결합 집중형

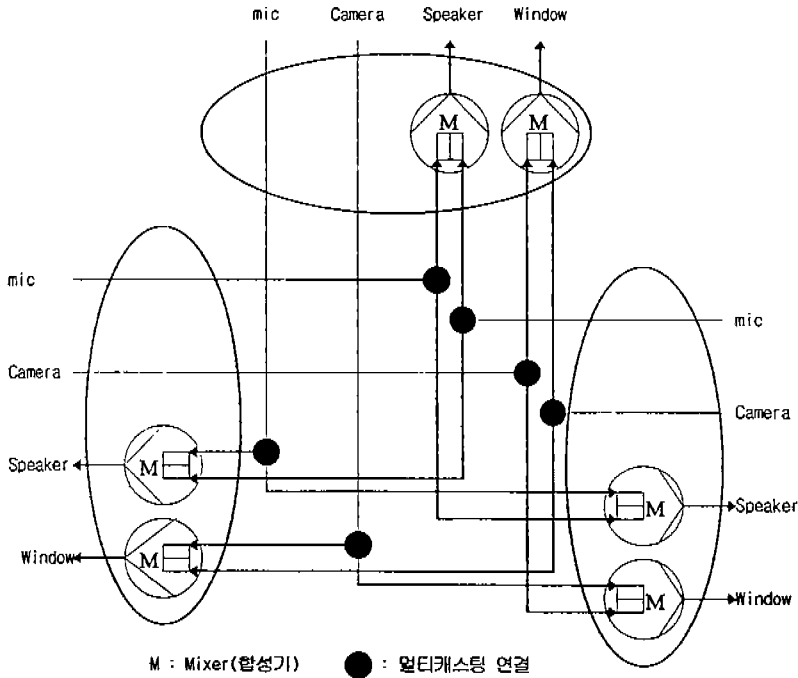
결합 집중형은 각 단말기에서 생성되는 오디오, 비디오 스트림을 결합시켜 하나의 스트림으로 생성한 후 이를 영상회의의 서버에 전달하는 방식이다. 영상회의의 서버는 각 단말들로 부터 받은 각 스트림을 분리하여 혼합한 후 각 단말기가 필요로 하는 정보로 가공한 후 다시 단말기들로 전달하여 주는 방식이다.



(그림 13) 분리 분산형 3자 영상회의 연결의 예
 (Fig. 13) An example of noninterleaved-distributed 3 person teleconference



(그림 14) 점합 분산형 3자 영상회의 연결의 예
 (Fig. 14) An example of interleaved-distributed 3 person teleconference



(그림 15) 분리 분산 멀티캐스트형 3자 영상회의의 연결의 예
 (Fig. 15) An example of noninterleaved-distributed 3-person teleconference

이 모델은 분리 중앙 처리형 모델에서 생길 수 있는 동기화문제는 해결되나 각 단말과 영상회의의 서버에서 오디오, 비디오 스트림을 접합시키는 일과 접합되어 전송된 스트림에서 오디오, 비디오 스트림을 분리해 내는 일을 해야 하는 단점이 있다.

4.2.2.3 분리 집중 멀티캐스트형

분리 집중 멀티캐스트형은 영상회의의 서버가 스트림을 멀티캐스팅을 지원하는 프로토콜을 이용하여 전송하는 방식이다. 분리 분산 멀티캐스트형에서와 같이 통신망의 부하를 줄일 수 있는 장점이 있으나 통신 오류 발생시 이를 복구하는데 문제가 있을 수 있으며 오디오, 비디오 스트림간의 동기화의 문제가 있을 수 있다.

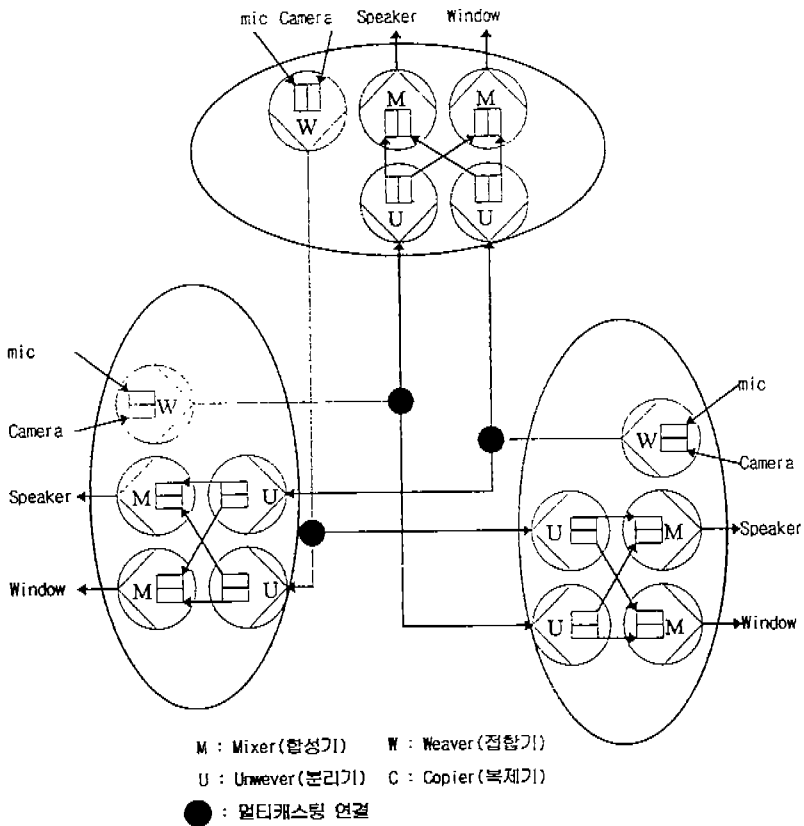
4.2.2.4 접합 집중 멀티캐스트형

접합 집중 멀티캐스팅형은 분리 집중 멀티캐스팅형과 통신방식에서는 동일하며 전달하는 스트림이

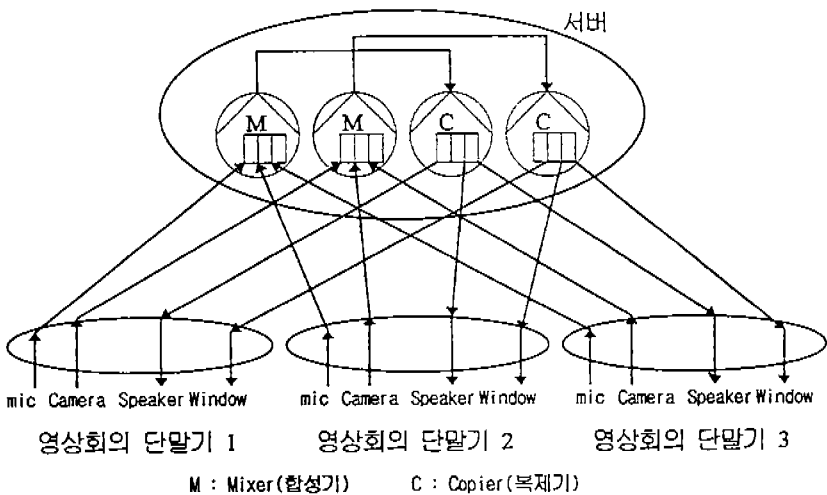
오디오와 비디오가 접합된 스트림이라는 것이 다르다. 오디오 스트림과 비디오 스트림의 동기화는 이를 수 있으나 스트림의 분리, 접합과 서버를 통한 전달 등으로 시간지연이 생길 수 있다.

5. 결 론

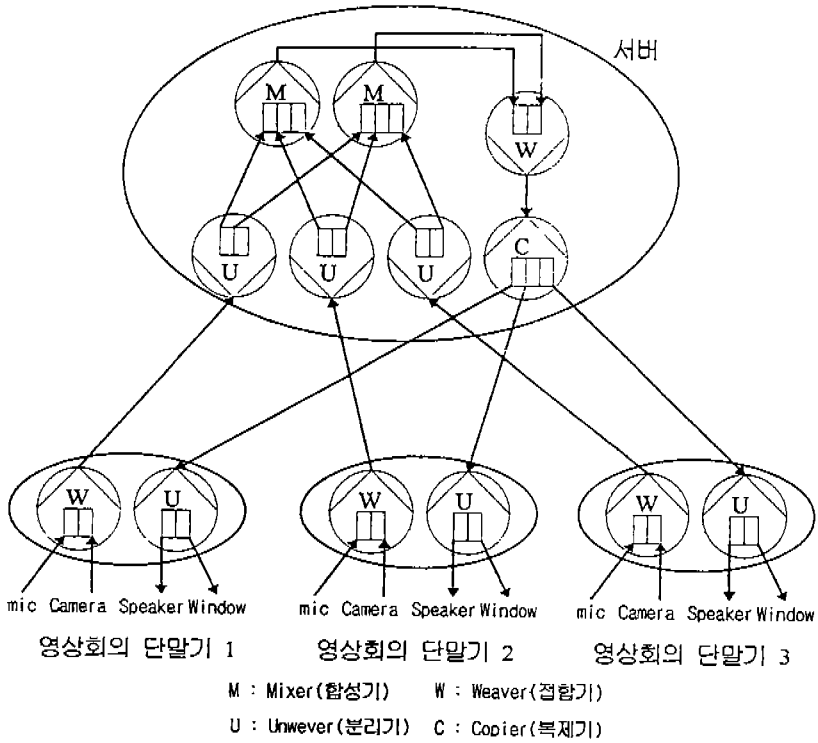
본 논문에서는 멀티미디어 처리 및 입출력 서버인 MuX에서 제공하는 멀티미디어 분산 처리를 위한 기본 객체:1)스트림 관련 객체, 2)스트림의 복제 및 분리에 관련된 객체, 3)스트림의 합성 및 접합에 관한 객체, 4)멀티미디어 프리젠테이션에 관련된 객체들의 개념과 특징 그리고 멀티쓰레드를 이용한 동시 수행 객체로 상기의 MuX 모델을 설계하는 방식에 대하여 설명하였고, 이를 이용한 컴퓨터 영상회의의 구성 방법을 설명하였다. 또한 영상회의의 시스템 구축시 고려하여야 할 사항과 영상회의를 구현시 가능한 스트림들의 연결구조와 각 구조의 장단점에 대해 논하



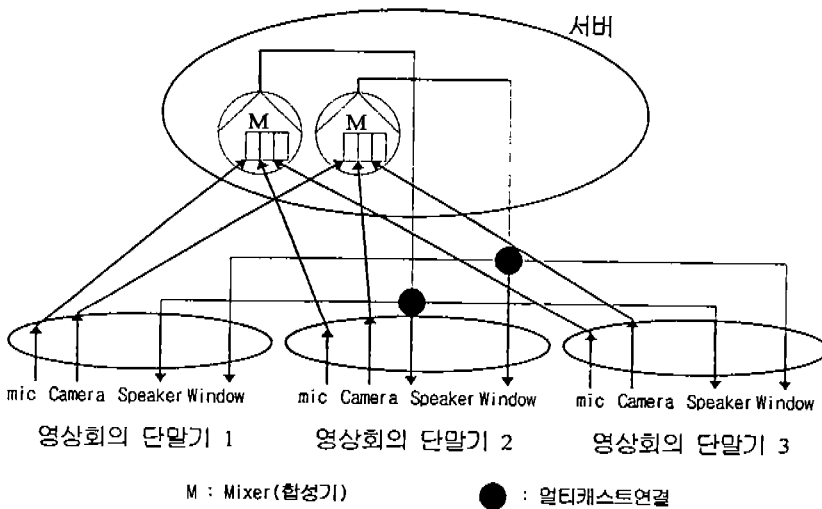
(그림 16) 접합 분산 멀티캐스트형 3자 영상회의의 연결의 예
 (Fig. 16) An example of interleaved-distributed 3 person teleconference using multicast protocol



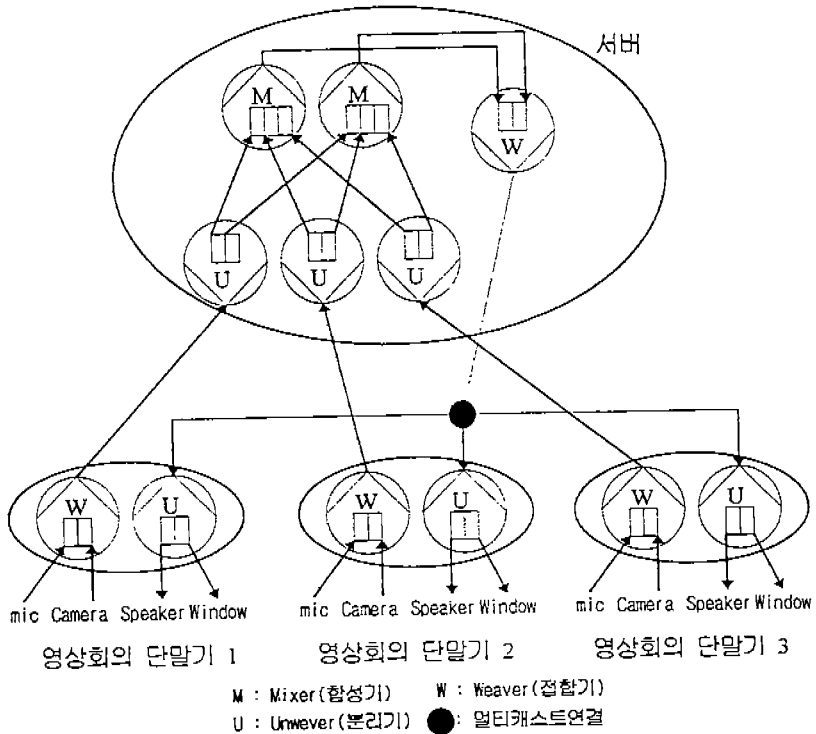
(그림 17) 분리 집중형 3자 영상회의의 연결의 예
 (Fig. 17) An example of noninterleaved-centralized 3 person teleconference



(그림 18) 접합 집중형 3자 영상회의의 연결의 예
 (Fig. 18) An example of interleaved-centralized 3 person teleconference



(그림 19) 분리 집중 멀티캐스트형 3자 영상회의의 연결의 예
 (Fig. 19) An example of noninterleaved-centralized 3 person teleconference



(그림 20) 접합 집중 멀티캐스트형 3자 영상회의의 연결의 예
 (Fig. 20) An example of interleaved-centralized 3 person teleconference using multicast protocol

였다.

오디오 스트림과 비디오 스트림을 분리하여 전송하는 분리형의 구조는 영상회의를 쉽게 구현할 수 있는 장점은 있으나 오디오 스트림과 비디오 스트림의 전달 경로시 동기화가 이루어지지 않을 수 있는 단점이 있다. 이에 반해 오디오 스트림과 비디오 스트림을 접합시켜 전송하는 접합형은 오디오 스트림과 비디오 스트림의 동기화를 이룰 수 있고 데이터를 전달하는 경로의 수를 줄일 수 있으나 오디오 스트림과 비디오 스트림을 접합하고 접합된 단일 스트림에서 오디오 스트림과 비디오 스트림을 분리하여야 하는 오버헤드가 있다. 영상회의의 서버를 두는 집중형은 분산형보다 영상회의의 단말기의 구조를 간단히 할 수 있고 데이터 스트림 전달을 위한 경로의 수를 줄일 수 있지만 영상회의의 서버의 성능이 우수하여야 한다. 데이터를 전송할 때 멀티캐스팅을 지원하는 프로토콜

을 이용한 멀티캐스팅형은 데이터전송시 각 단말기들에 데이터를 순차적으로 각각 전송하여야 하는 부담을 줄일 수 있으나 현재 데이터 전송시에 발생할 수 있는 오류를 복구하는 추가적 기능을 제공하여야 한다. 영상회의의 시스템을 구현할 때 영상회의의 방법, 이참석의 유연성, 단말기의 성능, 서버의 존재유무, 네트워크의 구조와 성능등을 고려하여 필요한 분산 스트림의 연결 구조를 결정하여야 할 것이다.

참고 문헌

[1] Doohyun Kim, SooHyung Oh, JinKyung-Hwang, Young-Hwan Lim and Earl Rennison, "An Integration and Synchronization Model for Audio, Video and Time-based Graphics," The 2nd Pacific Rim Conference on Artificial 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] 전윤호, 송동호, 박치항, "DLL에 의한 멀티미디어 데이터 처리 객체의 구현, 한국정보과학회 추계 학술대회," 1993, pp. 455-458.

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

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

[7] ITU-Telecommunication Standardization Sector, "T.GAT-Proposed draft Recommendation for Generic Application Template for T.120 Compliant Applications," Geneva, 16-25 November 1993.

[8] 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.



이 경 희

1990년 경북대학교 컴퓨터공학과 졸업(공학사)
 1992년 경북대학교 컴퓨터공학과 졸업(공학석사)
 1992년~현재 한국전자통신연구원 연구원
 관심분야: 멀티미디어, 실시간처리



임 현 규

1985년 숭실대학교 컴퓨터공학과 졸업(학사)
 1987년 한국과학기술원 전산학과 졸업(석사)
 1987년~현재 한국전자통신연구원 선임연구원
 한국과학기술원 전산학과 박사과정

관심분야: 그룹웨어, 시각언어, 분산 멀티미디어 처리



임 영 환

1977년 경북대학교 수학과 졸업(이학사)
 1979년 한국과학기술원 전산학과 졸업(이학석사)
 1985년 Northwestern University 졸업(이학박사)
 1979년~현재 한국전자통신연구원 책임연구원

관심분야: 멀티미디어, 초고속 정보 통신 시스템 소프트웨어, 에이전트

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