

(출판용 논문작성양식) * 저자정보 포함

부산멀티미디어 원격제어 시스템 구현

이민경^o 조동섭

이화여자대학교 컴퓨터공학과

mg0426@gmail.com, dscho@ewha.ac.kr

An implementation of distributed multimedia remote control system

Minkyung Lee^o Dongsu Cho

Dept. of Computer Engineering, Ewha womans Univ.

요 약

통신기술의 발전은 네트워크상에 분산되어 있는 시스템 간의 다양한 서비스를 가능하게 하였다. 특히 많은 사용자가 사용하는 미디어플레이어는 보다 더 쉽고 빠르게 제어할 수 있어야 한다. 본 논문에서는 소켓을 이용해 분산지의 미디어플레이어를 원격지에서 쉽게 제어할 수 있는 방법을 고찰한다. 소켓은 속도가 빠르고 사용자의 제어가 가능하다는 장점을 가지고 있다. 따라서 사용자의 요구에 따른 다양한 서비스를 보다 빠르고 쉽게 제공할 수 있다. 본 논문에서는 분산된 환경의 멀티미디어 시스템을 중앙의 사용자가 좀 더 효율적으로 관리하기 위한 소켓통신을 이용한 부산멀티미디어 원격제어 시스템을 설계, 구현한다.

1. 서 론

인터넷과 통신기술의 발달은 분산환경에서의 멀티미디어 서비스 기술개발의 다양한 변화를 촉진하고 있다. 사용자는 좀 더 빠른 전송속도와 응답시간, 지연시간, 지터의 제한, 동기화, 다양한 서비스 품질, 세션기능들을 요구한다.

우리가 사용하는 인터넷은 최선형(Bset of Quality)서비스를 제공함으로써 전송경로상의 라우터와 네트워크의 혼잡상태에 따라 변화한다. 한정되어 있는 네트워크 대역폭에서 많은 데이터의 이동은 지연과 지터, 데이터의 손실을 유발한다. 특히 비디오와 오디오, 영상이 합쳐진 멀티미디어데이터는 그 용량이 클 수 밖에 없다. 멀티미디어 데이터의 특성을 고려하지 않은 지금까지의 연구는 통신에 오버헤드가 컸다.

분산환경에서 멀티미디어 데이터를 좀 더 빠르고 효율적으로 이용하기 위해서는 네트워크를 통해 오고 가는 데이터의 양을 최소화하고 중앙의 제어자가 각 분산지의 클라이언트들을 쉽게 제어할 수 있어야 한다. 그러기 위해 필요한 기술이 원격제어이다.

인터넷의 발달은 먼 거리에 있는 컴퓨터를 자신의 컴퓨터를 이용해 제어할 수 있는 원격제어 기능을 가능하게 하였다. 예를 들어 서울의 보호자는 자신의 서울 집에 있는 컴퓨터를 통해 부산에 있는 환자의 병실 컴퓨터를 제어할 수 있다. 보호자는 부산에 있는 환자의 병실에 가지 않고도 환자의 컴퓨터를 제어해 환자의 필요에 따라 실시간으로 영화나 음악파일 등을 바꾸어 실행시켜 줄 수 있도록 한다.

원격제어 기술의 발달로 제어자는 원격지까지 이동하는 수고를 하지 않고도 자신의 편의에 따른 장소에서 쉽게 원격지의 컴퓨터에 접근할 수 있다.

본 논문에서는 다양한 원격제어 방법을 알아보고, 부산 미디어플레이어 원격제어 시스템을 설계, 구현하고 논문의 결론 및 앞으로의 연구계획을 기술한다.

2. 관련연구

- CORBA (Common Object Request Broker Architecture)

CORBA는 여러 회사와 조직들로 구성된 OMG(object Management Group) 컨소시엄에서 개발하고 발전시켜 나가고 있는 개방형표준이다. 객체지향, 분산 애플리케이션 개발을 하기 위한 프레임워크를 정의하는 것으로 이 구조는 분산 애플리케이션을 마치 하나의 언어로 구현되어, 하나의 컴퓨터에서 상호 작용하는 것처럼 네트워크 프로그래밍을 훨씬 쉽게 이용할 수 있도록 해준다. 이는 객체들간의 통신을 가능하게 해주는 객체지향 기반의 미들웨어를 통해 이루어진다.

- RMI (Remote Method Invocation)

자바프로그램을 실행할 수 있는 환경, 자바가상기계 (Java Virtual Machine)에서 쉽게 이용할 수 있다. JDK1.1 이상의 버전에 모두 포함되어있고 자바로 이루어진 개발환경에서는 자바언어의 특징인 이식성, 재사용성 등의 다양한 이점을 이용할 수 있다. 그렇지만 대규모시스템에서의 인프라텍스처에 필요한 요소들

만족시키는 데는 한계가 있다.

• 소켓 (socket)

소켓은 애플리케이션이 네트워크에 플러그인하여 동일한 네트워크에 플러그인 된 다른 애플리케이션과 통신할 수 있도록 한다.

소켓통신을 위해서 TCP, UDP 프로토콜이 대표적이다. TCP는 데이터의 유실이 없는 신뢰적 전송 프로토콜이고 UDP는 비신뢰적 전송프로토콜이다. 소켓은 구현이 쉽고 전송속도가 빠르다는 장점을 가진다.

• 차이점

CORBA와 RMI, 소켓은 네트워크상에서 원격제어를 할 수 있는 방법들이지만 각기 다른 차이를 가진다.

	장점	단점
CORBA	• 다른언어와 상호작용 • 다양한 서비스 제공	• 미들웨어의 구매
RMI	• JVM 환경에서 실행가능 • JAVA 언어의 특징을 지님	• 인프라구조에 필수요소의 불만족 • 오버헤드가 크고 전송속도 느림
소켓	• 전송속도가 빠름 • 다양한 언어, 다양한 환경에서 사용	• 프로그래머에 의한 구현

<표1> CORBA, RMI, 소켓의 장단점 비교

CORBA와 RMI는 분산환경에서의 원격호출에 대한 ORB(Object Request Broker)로 많은 유사점이 있다.

CORBA는 객체지향, 분산애플리케이션 개발을 하기 위한 프레임워크를 정의하는 것으로 마치 하나의 언어로 구현되어, 하나의 컴퓨터에서 상호작용하는 것처럼 네트워크 프로그래밍을 쉽게 하고 분산환경 트랜잭션 서비스와 같은 인프라구조에 필수요소를 만족시키기 위한 다양한 서비스를 제공한다.

자바의 RMI는 객체직렬화, 다운로드 가능한 객체실행, 자바인터페이스 등의 자바언어의 특성에 의존하는 프로그램으로, 자바언어가 가지고 있는 상속이나 재사용성 등의 다양한 이점을 이용할 수 있다. 또한 미들웨어의 구매 없이 JVM에 RMI 구현 환경이 구현되어 있어 누구나 쉽게 사용할 수 있다. 그렇지만 대규모시스템에서의 인프라구조에 필요한 요소를 만족하는 데는 한계가 있다. 또한 객체단위 직렬화를 통해 데이터를 넘겨주고, 받는 쪽에서도 직렬화 된 스트림을 복구해서 다시 객체로 변환해야 하므로 오버헤드가 크고 전송속도도 느리다.

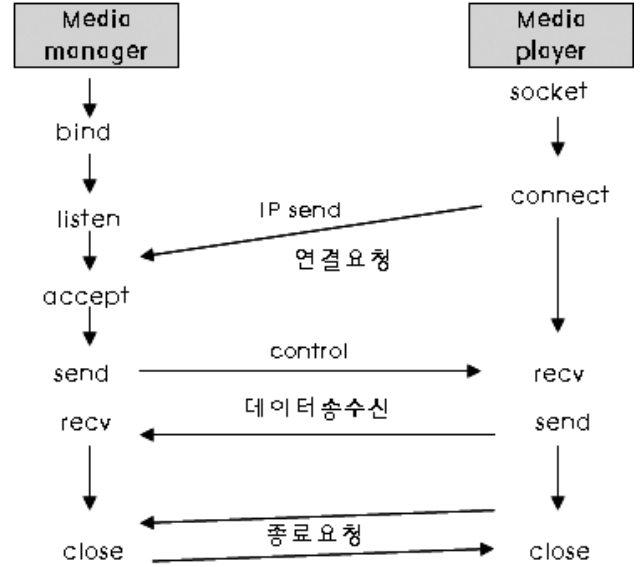
소켓의 경우 사용자가 직접 프로그래밍 할 수 있기 때문에 필요한 데이터에 대해서만 넘겨줄 수 있고, 다양한 환경에서의 전송이 가능하지만 프로그래머가 직접 구현해야 한다는 어려움이 있다.

3. 분산 멀티미디어 원격제어 시스템 구현

3.1 분산 미디어플레이어 원격제어 시스템 설계

물리적으로 분리되어 있는 두 시스템이 있다. Media player를 제어할 수 있는 Media manager 시스템과 media가 실행되는 환경의 Media player 시스템으로 정의한다.

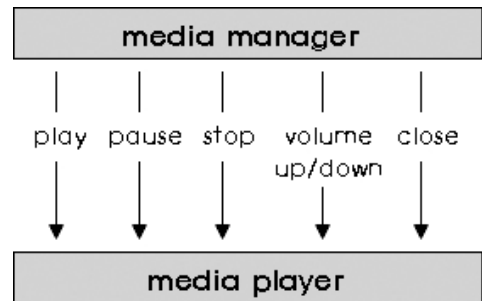
Media manager는 소켓을 이용해 분산되어 있는 시스템의 Media player를 제어할 수 있다



<그림 1> media manager와 player간 통신 과정

위의 그림처럼 두 객체간의 통신은 Media player의 연결요청으로부터 시작된다. 보낸 소켓에 대한 응답이 올 때까지 Media Player는 모든 준비를 마치고 대기 상태가 된다. Media manager로부터 응답이 오면 두 객체간에 통신을 시작한다. Media manager와 Media player가 서로의 데이터 수신에 관한 상태를 점검하며 데이터를 주고 받을 수 있게 한다.

분산 멀티미디어 원격제어 시스템은 미디어플레이어로서의 기본적인 기능을 갖추어야 한다.



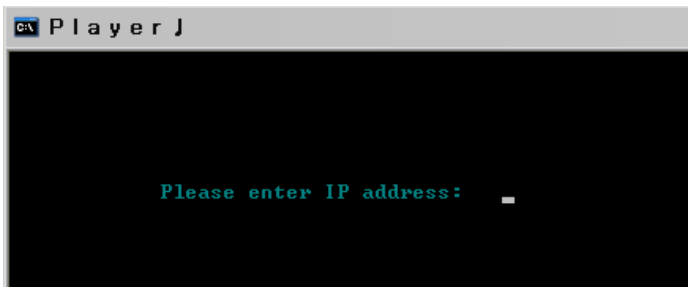
<그림 2> media manager와 player의 동작

- Play - 미디어 플레이어의 음악을 재생시킨다.
- Pause - 미디어 플레이어의 음악을 일시 정지 할 수 있다.

- Stop - 미디어 플레이어의 음악을 정지한다.
- Volume up/down - 재생되는 음악의 크기를 줄이거나 크게 할 수 있다.
- Close - 모든 통신을 마치고 소켓을 닫는다.

3.2 분산 미디어플레이어 원격제어 시스템 구현

본 논문에서 제안된 시스템은 purebasic을 사용하여 구현되었다.

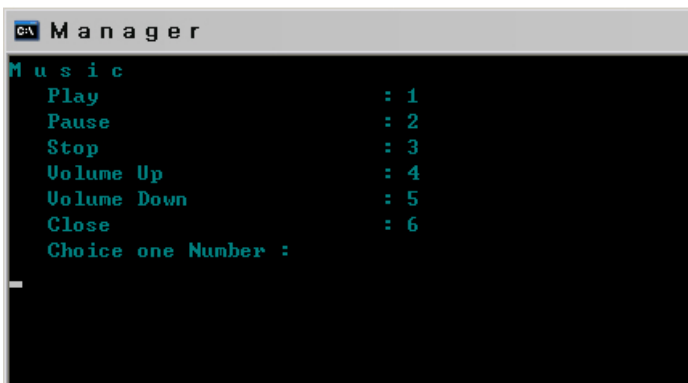


<그림 3> client측 연결 요청 화면

Media player는 manager에게 IP address를 입력하고 통신을 요청하는 소켓을 보낸다. Manager는 player측으로부터 전송된 IP를 확인하고, 통신을 시작한다

이 시스템은 TCP 프로토콜을 사용한 제어를 한다. 데이터의 전송상태를 확인하면서 통신을 하기 때문에, 네트워크의 혼잡상태를 예측한 통신을 제공할 수 있다. 이는 통신의 과부하를 줄이고, 예상되는 시간에 적절한 전송을 할 수 있는 서비스를 제공할 수 있어, 데이터 전송시에 발생할 수 있는 많은 문제들을 최소화 할 수 있다.

Media player와 통신을 시작한 Media manager는 다음과 같은 기능을 시작한다.



<그림 4> media manager 측 제어 화면

Media manager 측은 간단한 입력만으로 client 측의 media 파일을 제어할 수 있다. Player측이 연결을

마치고 대기 상태가 되면 Manager측의 화면에는 제어를 위한 화면이 출력된다. Manager 측의 사용자는 간단한 번호 입력으로 player 측의 음악 파일을 제어할 수 있다.

Media player는 Media manager의 명령에 따라 데이터파일의 동작을 제어한다. 분산된 시스템에서 관리자는 간단한 명령을 통해서 쉽게 사용자를 제어할 수 있다.

Media player는 통신에서 클라이언트의 역할도 하지만 개별적인 Player로써의 기능도 해야 한다. 즉, Manager의 명령에 의한 동작도 수행해야 하지만 개별적인 작동도 할 수 있어야 한다. Media player는 player측의 버튼 입력 이벤트에도 반응 할 수 있도록 해야 한다.



<그림 5> media player

위 그림은 Media player측의 실행화면이다. Mp3 파일이라서 음악을 들을 수는 없지만 실행되고 있는 파일의 이름을 화면에서 볼 수 있다.

본 논문에서 제안된 시스템은 분산멀티미디어 원격제어 시스템으로 서로 다른 네트워크환경과 OS환경에서의 구현과 사용이 쉽다. 전송상태에 대한 확인을 통해 데이터의 전송이 이루어지기 때문에 불필요하게 낭비되는 오버헤드를 줄일 수 있다. 또한 소켓을 이용한 제어방법은 CORBA나 RMI 등보다 데이터의 전송 속도가 빠르다.

그리고 제어 시 Player측에 저장된 DB 음원의 사용을 우선시 하기 때문에 멀티미디어 데이터의 품질도 우수하고 멀티미디어 데이터에 대한 문단복제나 수정 배포 등의 저작권 문제가 발생할 가능성이 적다.

4. 결론

본 논문에서는 간단한 분산 멀티미디어 원격제어 시스템을 구현하였으며 이를 통해 필요한 관리 방법이 제안되었다. 미디어 데이터의 관리자는 제어를 통해 외부의 사용자들에게 제공된 콘텐츠를 효율적으로 제어함으로써 그 기능을 더욱 극대화 할 수 있다. 또한

간단한 소켓을 이용함으로써 불필요한 오버헤드를 줄이며 불필요한 자원의 낭비 및 효율적인 전송을 할 수 있다.

추후에는 좀 더 다양한 기능을 추가할 수 있겠고 GUI 관점에서도 사용자의 접근을 용이하고 효율적으로 관리할 수 있도록, 관리자와 사용자의 GUI 다수의 사용자에 관한 쉽고 빠른 제어도 필요하다. 앞으로의 구현을 통해 보다 빠르고 편리한 서비스를 제공할 수 있는 제어시스템을 고찰하겠다.

4. 참고문헌

[1] Satoshi Itaya, Naohiro Hayashibara, Tomoya Enokido, Makoto Takiwaqa, "Distributed Coordination for Scalable Multimedia Streaming Model," IEEE2006

[2] Richard M. Adler, "Distributed Coordination Models for Client/Server Computing," IEEE Computer, pp14-22. April, 1995

[3] Robert Orfali, Dan Harkey, "Client/Server Programming with Java and CORVA"-2nd, 1998

[4] 김명호, 이윤준, 정연돈, "멀티미디어 시스템 개론," 홍릉과학출판사, 2006

[5] 한윤기, 구용완, "CORVA-ORB, JAVA-RMI, 소켓을 이용한 그룹 통신의 구현 및 성능분석," 2002 한국인터넷정보학회 (3권1호)

[6] Microsoft PRESS, "Inside Distributed COM," 1998

[7] www.purebaisc.com