

원격 제어 프로그램의 설계 및 구현

이 철⁰ 김상철 김종환
 한국외국어대학교 컴퓨터공학과
 (zippy, kimsa, jhkim)@hufs.ac.kr

Design and Implementation of a Remote Control Program

Chul Lee⁰ Sang-Chul Kim Joong-Hwan Kim
 Dept. of Computer Science & Engineering, Hankuk University of Foreign Studies

요 약

원격 제어 프로그램은 파일전송 관리, 고장난 컴퓨터의 원격 진단 및 A/S, 시스템 관리자의 원격지 컴퓨터 관리, 원격 강의 시스템 등 다양한 곳에 이용되고 있다. 본 논문에서는 일대다 접속모델을 지원하는 원격 제어 프로그램의 구조와 동작원리를 기술한다. 여기서 일대다 접속이란 한 컴퓨터의 화면을 원격지에 있는 다수 컴퓨터들이 공유 및 제어하도록 하는 것을 말한다. 또한, 프로그램의 성능에 주요한 요인이 되는 화면 갱신 이벤트 처리와 마우스 움직임 이벤트 처리에 대한 효율적인 방법을 제안한다.

1. 서 론

원격 제어 프로그램은 원격지에서 자신의 컴퓨터 환경을 이용하거나, 컴퓨터들을 관리하기 위해 개발, 발전되어 왔다. 현재는 파일전송 관리, 고장난 컴퓨터의 원격 진단 및 A/S, 시스템 관리자의 클라이언트 원격 관리, 원격 강의 시스템 등 다양한 곳에 이용되고 있다[1][2][5].

원격 제어 프로그램은 크게 화면공유와 원격제어의 기능을 갖는다. 화면공유 기능은 호스트의 화면을 클라이언트에게 제공한다. 클라이언트는 공유되는 화면을 보며 호스트를 원격 제어 할 수 있게 된다. 원격제어 기능은 클라이언트에서 발생시킨 이벤트를 호스트에서 처리 되도록 하여 클라이언트가 실질적으로 호스트를 조작 할 수 있도록 만들어 주는 기능을 한다.

윈도우 기반 원격 제어 프로그램도 다른 윈도우 응용 프로그램과 마찬가지로 윈도우의 이벤트에 의해서 동작하게 되며, 이러한 이벤트에 의해 화면의 갱신(Update)과 마우스 및 키보드의 제어가 이루어지게 된다[3]. 순간적 또는 연속적으로 무수히 발생하는 이러한 이벤트들을 여과 없이 처리하게 되면 너무 잦은 화면갱신과 불필요한 이벤트 전달이 발생하게 되어 네트워크 트래픽의 증가와 이벤트 처리에 대한 오버헤드의 증가를 가져옴으로써 전체적인 성능저하를 가져오게 된다. 이러한 성능저하는 일대일(1:1) 접속모델의 프로그램에서 보다 원격 강의 시스템과 같은 다수의 클라이언트가 존재하는 일대다(1:n), 다대다(m:n) 접속모델에서 전체적인 성능에 더욱 큰 영향을 주게 된다.

따라서 본 논문에서는 일대다 접속모델을 지원하는 원격 제어 프로그램의 구조와 동작원리를 기술한다. 또한, 연속적으로 발생하는 마우스 움직임 이벤트와 화면갱신 이벤트를 효율적으로 처리함으로써 원격 제어 프로그램의 성능을 향상 시키는 방법을 제안하고자 한다.

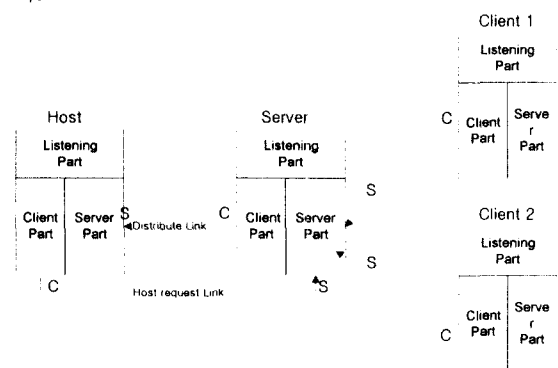
2. 원격 제어 프로그램의 전반

본 장은 우리가 개발한 원격제어 프로그램의 전반적인 구조와 동작원리를 기술한다.

지금까지의 원격 제어 프로그램은 보통 서버(server)와 클라이언트(client) 형태를 가진 일대일(1:1) 접속모델이 많았다[1].

반면에 우리의 원격 제어 프로그램 일대다 접속모델을 지원한다. 이 모델은 3종류의 컴퓨터로 구성된다. 화면을 공유하는 컴퓨터를 호스트, 호스트의 화면을 원격지에서 보거나 호스트에 입력이벤트를 발생시키는 컴퓨터를 클라이언트, 호스트의 화면을 클라이언트로 또는 클라이언트의 입력 이벤트를 호스트에 전달하는 컴퓨터인 서버가 있다.

호스트와 서버는 한 개인 반면에 클라이언트는 다수이다.



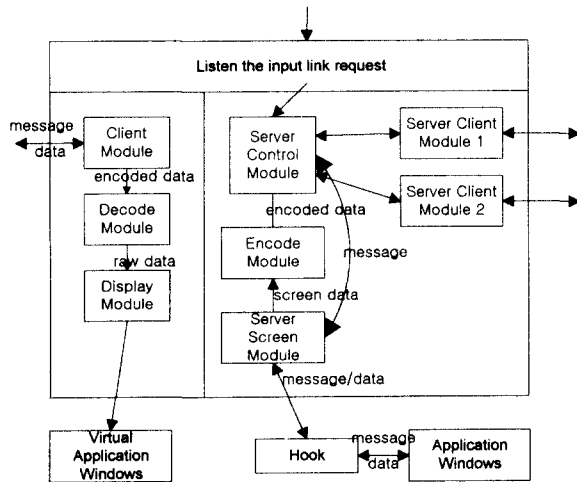
[그림 1] 서버-클라이언트 모델 구조

2.1 구조

일대다 접속모델에서는 호스트와 서버가 독립적으로 존재하여 그 역할을 수행하게 된다.

결국, 원격제어 프로그램은 호스트용, 서버용 및 클라이언트용으로 나뉘게 되고, 각각 자신에 알맞은 기능을 수행하게 된다. 하지만 이들의 구조는 공통적으로 크게 세 부분으로 구분되는 된다. 접속요구(request)를 받아들이는 수신부분(Listen part), 데이터 전달을 받게 되는 클라이언트 부분, 데이터를 공급하는 서버부분으로 구성되어 있다. 서버와 호스트사이에는 호스트가 화면데이터를 공급하고 호스트가 이를 받게 되므로 서버부분과 클라이언트부 모두가 동작하게 된다.

원격 제어의 핵심 기능은 화면 공유 기능과 이벤트 제어 기능이다.



[그림 2] 원격 제어 프로그램의 구조

2.2 화면 공유

화면 공유 부분은 호스트의 화면을 클라이언트에게 제공하는 부분으로 원격 제어 프로그램에서의 가장 큰 데이터 양과 처리시간을 가지게 된다. 따라서 전송되는 데이터의 양을 줄이기 위해서 다양한 이미지 압축 기법들이 이용된다[4]. 클라이언트에게 보여지는 화면은 호스트의 화면 변화가 생길 때 마다 갱신 되는데, 이 화면의 변화(변화되는 시점, 변화된 부분)를 효과적으로 찾아내는 것 역시 원격 제어 프로그램의 성능에 상당한 영향을 미치는 중요한 요소이다.

기존 원격제어 프로그램들은 일정한 주기마다 호스트의 화면을 클라이언트로 전달하거나, 호스트에 화면 갱신 이벤트가 발생되면 변화된 화면을 클라이언트로 전달해서 이용하고 있다. 본 시스템에서는 이 두 가지 방법을 조합한 방법을 이용하고 있다.

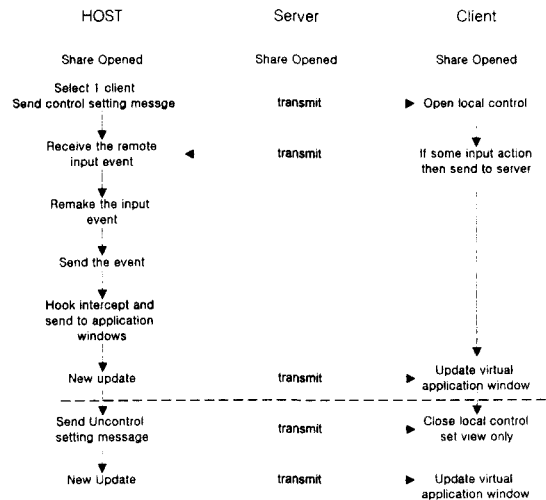
클라이언트의 화면을 주기적으로 갱신하나, 갱신 주기 동안에 호스트에서 발생한 화면 갱신 이벤트 정보를 이용하여 호스트 화면의 변화된 부분을 쉽게 파악할 수 있게 하였다.

변화가 일어난 부분은 몇 개의 블록으로 나눈 후 각 블록에 대해 변화가 발생되었는지를 검사해서 변화가 일어난 부분만을 클라이언트에 전송하는 방법을 사용하여 데이터량을 줄이도록 하였다.

2.3 이벤트의 제어

원격 제어의 핵심 기능인 이벤트 제어는 화면공유 기능과는 반대로 클라이언트에서 호스트쪽으로 일어나게 된다. 클라이언트들 중에서 호스트로부터 제어권을 부여 받은 클라이언트가 마우스, 키보드 입력 같은 이벤트를 발생시키게 되면 이벤트가 서버를 거쳐 호스트로 전송된다. 전달된 이벤트는 호스트에서 로컬 이벤트로 발생되어 이벤트에 대한 처리가 이루어지게 된다. 이에 따른 결과는 모든 클라이언트들에게 전달된다. 이 과정의 반복을 통해 클라이언트는 이벤트 제어를 제공받게 된다.

일반적으로 이벤트 데이터 자체의 크기는 매우 작기 때문에 압축을 하지 않아도 클라이언트가 이벤트들을 호스트로 전송하는 것은 시간상으로 문제가 되지 않는다. 그러나 마우스 움직임 이벤트는 단위 시간에 수많이 발생하므로 호스트쪽에서는 이들 이벤트를 처리하면서 그에 따른 결과인 화면 데이터를 계속해서 서버에게 전송하게 되면 이에 따른 호스트의 부담이 급격히 증가하게 된다. 이러한 부담은 결국 전체적인 시스템 성능 저하를 가져오게 된다.



[그림 3] 이벤트 제어 과정

3. 이벤트의 처리

호스트 자체의 이벤트와 클라이언트로부터 전송되는 이벤트 등 원격 제어 프로그램 내에서의 수많은 이벤트를 여과 없이 모두 처리하게 되면 이벤트 전송에 대한 네트워크 트래픽의 증가와 이벤트 처리에 대한 호스트의 부담이 증가하게 되어 전체적인 성능저하를 가져올 수 있다.

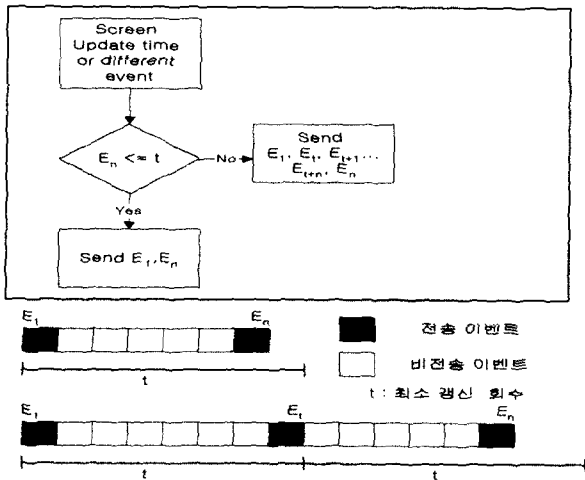
본 논문에서는 마우스 움직임 이벤트에 대한 처리 횟수를 줄이는 방법을 제안하였다.

화면 갱신 주기 T 를 설정하고, T 동안에 최소 갱신 회수 t 를 설정한다. T 동안에 실제 발생된 이벤트를 E_1, E_2, \dots, E_n 이라 하자.

발생된 이벤트의 수 E_n 이 t 보다 적으면 첫번째 이벤트 E_1 과 마지막 이벤트 E_n 을 전송한다. 만약 E_n 이 t 보다 크면 첫번째 이벤트 전송 후 t 마다 이벤트 전송하고 마지막 이벤트를 전송한다. 이렇게 함으로써 연속된 동일 이벤트 중 처리되지 않아도 결과에 영향을 주지 않는 중간 단계를 줄일 수 있게 된다. 그러나 정확한 이벤트 처리를 위해 첫번째와 마지막 이벤트는 반드시 전송되어야 한다.

연속된 동일 이벤트의 수 E_n 의 최대크기는 화면 갱신주기에 따라 달라지게 되며, 갱신주기 내에서 다른 이벤트가 발생할 때까지 연속 발생된 이벤트 수를 나타낸다. 만약 연속된 동일 이벤트가 화면 갱신주기 보다 길게 발생하게 되면, 화면 갱신 주기 내에 발생된 이벤트들에 대해 위에 방법을 적용하고 나머지 이벤트는 다음 갱신주기에 처리하게 된다.

- $E_n \leq t$ 이면 : E_1, E_n 전송.
- $E_n < t$ 이면 : $E_1, E_1, \dots, E_{t-m}, E_n$ 전송.
- E_1 과 E_n 은 반드시 전송.
- E_n 크기는 화면 갱신주기 의해 결정.

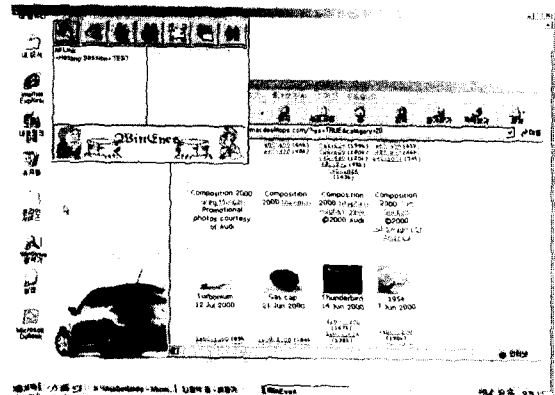


[그림 4] 이벤트 전송주기

4. 구현

본 시스템은 Win32API를 이용하여 Win2000(P3 550Mhz, 256M)과 Win98(P2 333Mhz, 128M)환경에서 구현 되었으며, 로컬 네트워크 상에서 30대의 클라이언트를 연결하여 테스트 하였다.

움직임 이벤트의 전송횟수는 t 에 따라서 결정되나, t 동안에는 한 번의 전송만 일어나기 때문에 움직임 이벤트에 전송 횟수를 줄이는데 상당한 효과를 나타내었다.



클라이언트에 보여지는 호스트 화면의 모습
[그림 5] 실행화면

그러나 t 값을 너무 크게 설정할 경우 마우스 움직임의 표현이 부드럽지 못하게 되며, 또한 움직임 이벤트들 사이에 다른 이벤트들이 자주 발생하게 되면 그 효과가 크게 줄어들게 된다.

5. 결론

본 논문에서는 다수의 클라이언트를 가지는 일대다 원격 제어 프로그램을 설계 및 구현하였다. 프로그램은 크게 서버용, 클라이언트용, 호스트용으로 구분되지만, 이들의 기본적인 구조는 동일하다. 화면 갱신 이벤트의 처리와 움직임 이벤트에 대한 처리를 줄여 이벤트 처리의 효율을 높일 수 있게 하였다. 마우스 움직임 이벤트의 경우 갱신 회수 t 값이 크게 되면 움직임 표현이 부자연스럽게 되고, 너무 작게 되면 전송 횟수가 증가하게 되므로 적절한 갱신주기 값에 선택이 필요하다. 화면 갱신 이벤트는 주기적 갱신으로, 갱신 횟수가 자체는 고정적이 된다.

6. 참고문헌

[1].Tristan Richardson, Quentin Stafford-Fraser, Kenneth R. Wood & Andy Hopper, "Virtual Network Computing", IEEE Internet Computing, Vol2 No.1, Jan/Feb 1998 pp33-38
 [2]. Norbert A.Streitz, Jorg Geibler, Jorg M. Haake, and Jeroen Hol, "DOLPHIN: Integrated Meeting Support across Local and Remote Desktop Enviroments and Live Boards", Proceedings of the conference on Computer supported cooperative work, 1994, Pages 345 - 358
 [3].Charles Petzold, "Programming Windows fifth edition" Microsoft press 1998
 [4].Rafael C. Gonzalez, Richard E. woods, "Digital image processing", addison Wesley longman 1992