

# 분산형 전략 시뮬레이션 게임을 위한 통신모듈

노기남\*, 심중표, 송동호  
한국항공대학교 전자계산학과

## A Communication Module for Distributed Strategic Games

GiNam Rho\*, JungPyo Shim, DongHo Song  
Dept. of Computer Science, Hankuk Aviation University

### 요 약

응용프로그램과 통신처리 프로그램간의 접속에 있어서는 응용 프로그래머가 통신을 고려하지 않고 개발해야 하며 중앙집중식 다자간 통신을 분산식 다자간 통신으로 지원할 수 있는 통신 모듈이 이상적이다. 중앙집중식 통신을 사용시에는 반드시 서버의 역할을 하는 중계시스템을 거쳐야 하지만 분산식 통신을 한다면 중계자를 거치지 않고 통신게임을 원하는 사용자간의 의사에 의해서 게임을 할 수 있다. 본 논문에서는 클라이언트와 서버의 기능을 통합한 단일모듈을 이용해서 분산통신 모듈이 가져야할 편의성, 범용성, 이벤트 동기화, 다중성등에 관하여 논의한다. 그리고 전략 시뮬레이션의 형태에 따른 세가지 유형의 예를 설정한다. 첫째는 웹 브라우저의 형태를 갖추면서 정해져 있는 시나리오를 따르는 게임이다. 둘째는 바둑과 같은 형태의 토큰을 통해서 제어권을 행사하는 것이다. 셋째는 토큰을 갖지 않고도 제어권을 행사하는 것이다. 마지막으로 본 논문에서 제안하는 통신모듈을 이용해서 개발 중에 있는 Death Flower라는 게임을 소개한다.

### 1. 서론

정보통신용 응용소프트웨어의 개발은 국내외에서 활발히 진행되고 있는 연구분야이다. 그중 게임분야에서는 하나의 시스템을 이용하던 종래의 제약을 벗어나서 통신을 이용한 게임 소프트웨어를 만들고 있는 추세에 있다. 그러나 대부분이 국외에서 제작된 제품이 대부분의 시장을 점유하고 있다. 특히 분산통신을 지원하는 게임은 아직 국내에서는 거의 응용되지 않고 있는 실정이다.

TCP / IP를 근간으로 하고 있는 소켓 인터페이스는 상위 응용프로그램에서 직접 사용하기에는 몇 가지의 제약사항을 갖고 있다. 첫째는 다중성을 지원하지 않기 때문에 통신을 지원하는 모듈에서 지원해야한다. 둘째는 편의성이라는 점에서 응용 프로그래머가 통신에 관한 것을 고려하지 않도록 해야한다. 셋째는 이벤트 처리에 있어서의 동기화와 동일 시점에서 서로 다른 호스트들간에 이벤트 처리를 지원하지 않고 있다.

본 논문의 2장에서는 클라이언트와 서버 통합에 대하여 논하며 3장에서는 통합된 모듈의 동적 재조합을 논한다. 4장에서는 호스트간의 이벤트 동기화를, 그리고 5장에서는 제안된 통신 모듈을 이용한 분산 시뮬레이션 게임의 엔진 개발에 대하여 논한다.

### 2. 기능통합 통신모듈

클라이언트와 서버는 자신들이 갖는 고유한 기능을 갖는다. 예를들면 네트워크 접속(Connect)이나 네트워크 수락(Accept)을 들 수 있다. 또한 공통적으로 사용하는 네트워크 단절(Disconnect)등의 기능도 갖고 있다. 본 논문에서는 입차적으로 공통기능을 통합하고 이차적으로 각기 갖고 있는 고유기능을 통합한다.

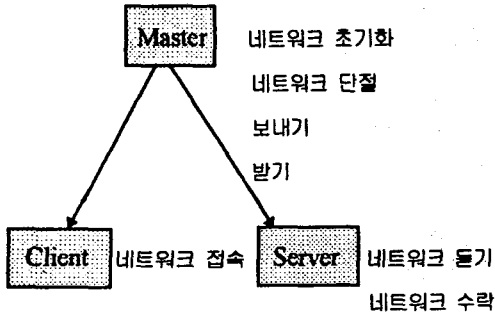


그림 1. 일차적 통합 구조

<그림1>에서는 일차적으로 마스터 클래스를 사용해서 클라이언트와 서버를 통합을 했다. 이는 통신상에서 같이 사용되는 네트워크 초기화, 네트워크 단절, 보내기, 받기들을 통합한 상태이다. 그리고 <그림2>에서는 클라이언트와 서버의 고유기능을 통합한 단일 통신모듈을 보이고 있다.

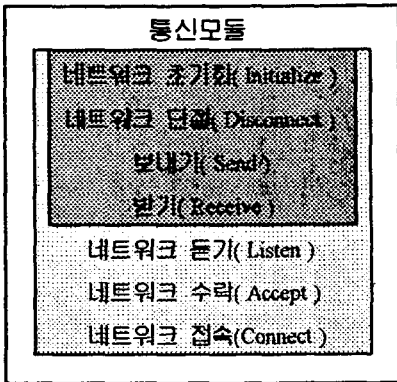


그림 2. 이차적 통합 단일 모듈

통신이 설립되는 과정에서 고려되어야 하는 클라이언트와 서버의 역할을 응용 프로그램에서는 고려할 필요가 없다. 또한 소프트웨어 운영중에 클라이언트와 서버의 공통기능인 보내기, 받기하는데 통신상의 역할을 구별할 필요도 없다. 이러한 제약에 대해서 단일모듈을 제안하고 있다. 클라이언트 측면에서든 서버측면에서든 통신이 설립된 이후에는 각자의 역할을 구별할 필요가 없게 된다. <그림2>의 모듈은 본 논문에서 분산통신을 지원하는 모듈로 제안하고 있다.

<그림3>에서는 <그림2>에서 제안한 클라이언트와 서버를 통합한 모듈을 통해서 각자의 기능을 수행하고 있다. 통신이 설립되는 시점에서는 네트워크 설립을 시도하는 클라이언트가 있고 이를 수락하는 서버가 있다. 그러나 네트워크가 설립되고 있는 호스트들은 서버와 클라이언트 기능을 같이 수행할 수 있으며 필요에 따라서 두가지의 기능중에서 하나를 수행할 수 있다.

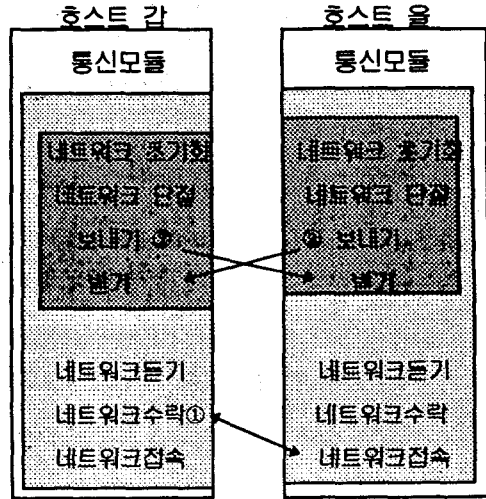


그림 3. 통합된 모듈을 통한 통신

### 3. 통신모듈의 동적 재조합

본 논문에서는 통신응용 프로그램이 갖어야 하는 기능에 대해서 편의성, 범용성, 이벤트 동기화, 다중성을 지원해야 한다는 것을 전제하고 있다.

첫째는 편의성이다. 통신상에서의 여러가지의 문제물 응용 프로그래머 직접 판여하지 않도록 해야한다. 통신에 관련된 부분은 독립된 클래스로 제안하고 응용 프로그래머는 응용 소프트웨어에서 보내고 받는 시점만을 결정하게 된다.

둘째는 범용성이다. 통신 모듈은 다양한 응용 소프트웨어를 지원해야한다. <표1>에서는 소켓 상태에 따른 발생가능한 메시지를 통해서 현재의 소켓 상태 다음에 발생 가능한 상태를 정의하고 있다.

속 성	발생가능 Message
READ	WM_RECV, WM_SEND
WRITE	WM_RECV, WM_SEND
OOB	WM_RECV, WM_SEND
CONNECT	WM_CONNECT
ACCEPT	WM_SEND, WM_RECV, WM_LISTEN
CLOSE	WM_SEND

표 1. 사용자 정의 메시지

<표 1>은 대화 프로그램과 웹브 화일 서버, 전략 시뮬레이션 게임 구현을 통해서 정리한 것이다. 이는 다양한 소프트웨어를 통해서 좀더 범용성을 갖기 위함이다. <표 1>에 의거한 두가지 형태의 사용 예를 들어 본다. <그림 4>에서는 웹브화일프서버를 보이고 있다. 여기에서는 여러 형태의 정보가 사용자 즉, 클라이언트에 의해서 정보 분류, 선택되고 있다. 수행상의 모든 제어는 클라이언트에서 하고 있다. 그러나 <그림 5>에서는 두 호스트간에 동등한 제어권을 행하고 있다. 두 가지의 응용에서는 데이터를 처리하고 보내고 받는 과정만 다르다. 이러한 비교를 바탕으로 클라이언트와 서버를 통합한 통신모듈을 구현하였다.

그러나 위의 응용 소프트웨어들은 직접 설계해서 만들었다는 점이 완전한 범용성을 보장할 수는 없다. 예를 들어서 화일전송을 위해서는 ABOR, LIST, PASS 등의 FTP 프로토콜에서 지정하고 있는 명령어를 교환한 후에 데이터 전송이 이루어지는 FTP 프로토콜을 따라야 한다는 것이다. [1]

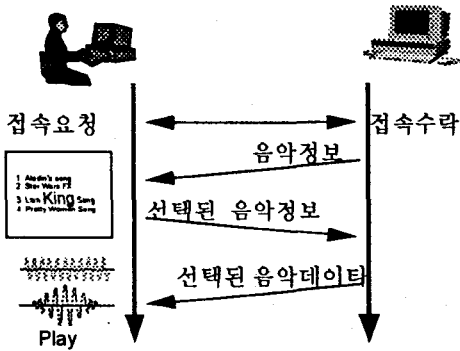


그림4 웹브화일 서버

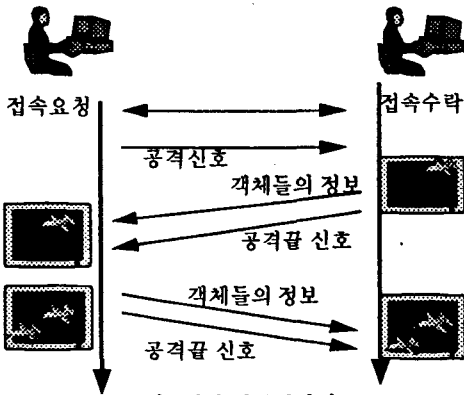


그림5 전략 시뮬레이션

셋째는 이벤트 동기화이다. 서로 다른 네트워크 거리를 갖는 호스트들 간에 발생하는 이벤트들에 대해서 동기화를 지원해야 한다. 또한 서로 다른 호스트들 간에 발생된 이벤트를 동일 시점에서 처리할 수도 있어야 한다. 이 문제는 4장에서 구체적으로 다루기로

한다.

넷째는 다중성이다. 여러 호스트들 간에 분산식 다자간 통신을 지원해야한다. 이를 위한 통신 구조는 멀티캐스트 및 MBone 등에서 제시되고 있지만 본 논문에서는 언급을 하지 않기로 한다.

#### 4. 호스트들간의 이벤트 동기화

서로 다른 네트워크 거리를 갖는 게임에 있어서는 이벤트 동기화 문제를 갖고 있다.

첫째, 네트워크 거리에 따른 전송 지연에 의해서 생기는 이벤트 동기화이다. 먼저 일어난 이벤트가 먼저 처리되어야 한다.

둘째, 상호간의 이벤트 처리에 있어서 다른 시점에서 이벤트 처리가 되는 문제이다. 이벤트를 발생시킨 호스트에서는 이벤트를 받은 호스트와 같은 시점에서 이벤트를 처리되어야 한다.

첫번째의 문제에 대한 예를 <그림 6>에서 보이고 있다. 서울에 있는 갑이 대전에 있는 을과 네트워크 설립되어 있다. 이때 접속을 요청했던 을은 갑의 논리적 타임(4)에 맞춘다. 이후에 부산에 있는 병이 대전에 있는 을에게 네트워크 설립하였을때 이미 맞추어진 을의 논리 타임에 병이 맞춘다. 각각의 호스트간에는 서로 다른 시간을 갖을 수 있으나 타임간격은 같다는 전제를 갖고 있다. 즉, 네트워크 설립이 뒤후의 호스트들은 서로 타임스텝을 갖는 메시지를 주고 받음으로써 전체적인 논리시간을 갖게 된다. <그림 6>에서는 타임 14 시점에서 갑의 이벤트가 발생되었고 타임 15 시점에서 병의 이벤트가 발생되었다. 부산에 있는 병은 타임 15에서 자신의 이벤트를 발생시켰고 타임 16 시점에서 갑이 발생시킨 이벤트를 받았다. 이러한 경우에 병은 먼저 발생된 갑의 이벤트를 처리해야 한다. 이를 위해서 각 호스트들은 이벤트 큐를 갖고서 각 이벤트에 대한 타임스텝을 검색한다. 그리고 타임스텝의 순서에 따라서 처리함으로써 상호간에 이벤트 동기화를 맞출 수 있다.

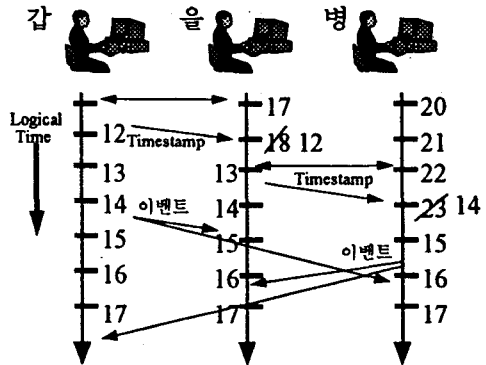


그림 6. 상호간의 이벤트 처리 동기화

예를 들어서 두 사람이 서로를 겨냥해서 총을 쏘는 게임 프로그램에서 한 사람이 먼저 피했는지 아니면 상대가 먼저 피했는지의 이벤트 순서만 유지된다면 호스트들 사이에서는 상대의 화면을 보지 못하므로 발생된 이벤트가 호스트들 간에 동기화되는 것이다.

두번째의 문제는 이벤트가 발생함과 동시에 자신의 다음 대응을 생각하는 경우에 한 쪽이 다른 쪽보다 먼저 이벤트가 발생한다면 먼저 이벤트가 발생한 쪽은 나중에 발생한 쪽보다 우월한 입장에서 게임이 이루어진다.

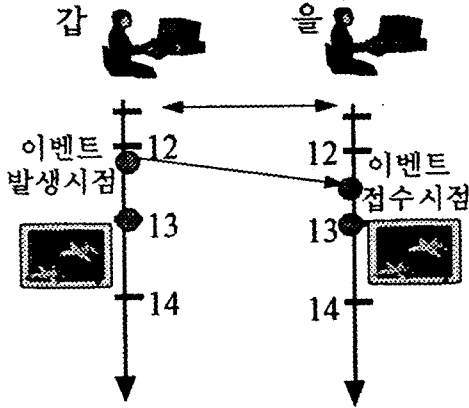


그림 7. 상호간의 이벤트 동일시점 처리

<그림 7>에서는 경우에 따라서 발생할 수 있는 게임의 불합리성을 없애기 위한 예를 보이고 있다. 이 내용은 호스트들 간에 타임스탬프를 갖는 메시지를 주고 받음으로서 전체적인 논리 시간을 형성하고 있다. 갑에서 두 개의 객체가 이동하는 이벤트가 발생했다면 먼저 타임스탬프를 붙인 객체정보를 을에게 전송한다. 을에서는 갑에게서 받은 이벤트에 붙은 타임스탬프를 검색해서 현재의 자신의 논리타임과 비교함으로써 네트워크상의 시간지연을 고려한 이벤트 타임간격을 갑에게 알려줌으로써 발생된 이벤트를 처리하는 상호 지연시간을 정한다. 지연시간이 결정되면 이벤트 처리를 하는 논리 시간을 정해서 갑에게 보내는 방식으로 상호간에 이벤트를 동일 시점에서 처리하게 된다. 지정된 타임간격은 또 다른 호스트의 네트워크 접속시에 최대지연 시간을 기준으로 다시 정한다. 이벤트 처리에 지연 시간을 갖게되지만 동일 시점에서 이벤트를 처리하게 됨으로 게임을 진행할 때 호스트들 간에 생길 수 있는 불합리성을 없앨 수 있다.

## 5. 게임엔진

개발중에 있는 전략 시뮬레이션 게임 DEATH FLOWER는 윈도우용으로 게임의 완성도보다는 윈도우에서 원활히 수행될수 있는 게임엔진 개발에 중점을 두고 작성된 프로그램이다. 현재까지 윈도우게임의 한계였던 느린 그래픽 출력의 극복을 위해 마이크로소프트사의 WinG 라이브러리를 이용하였다. 게임엔진

은 크게 4부분으로 나누어지며 DIB을 출력하기 위한 화면출력 관리자, 배경을 미디화일을 연주하기 위한 음악 관리자, 효과음인 웨이브화일을 연주하기 위한 효과음 관리자, 그리고 상대의 데이터를 관리하기 위한 데이터 관리자가 있다.

화면 출력관리자는 윈도우즈 그래픽 화일 포맷인 DIB 화일을 화면에 디스플레이하며 TILE 방식으로 화면출력 하기위한 함수와 원하는 좌표계(좌측 상단 기준)에 DIB을 디스플레이하는 2개의 출력함수가 있다. 디스플레이상에서 깜박거림을 제거하기 위하여 WinG의 디바이스 컨텍스트 상의 오프 스크린에 더블 버퍼링을 한다. WinG의 디바이스 컨텍스트는 일반 윈도우의 디바이스 컨텍스트와는 달리 메모리를 이용함으로써 비트맵 출력성능을 향상시킬 수 있다.[2]

데이터 관리자는 디스플레이할 오브젝트에 대한 자료를 링크드 리스트로 관리하고 있으며 전투모드는 공격과 수비, 2개의 모드로 구성되어있으므로 공격 모드에서는 아군의 화면출력 관리자, 수비모드에서는 적군의 화면출력 관리자를 이용한다.

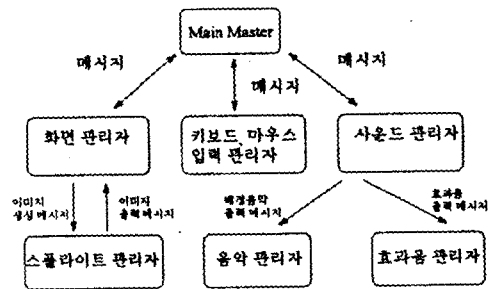


그림 8. 게임엔진의 개념도

<그림 8>은 게임엔진 관리자의 개념도이다. 이 엔진에서는 통신 프로토콜에 상관없이 독립적으로 수행될 수 있도록 각각의 역할을 분담하여 클래스화하였으므로 프로그래머가 어떤 관리자가 사용될 것인지를 판단하여 사용하면 된다. 즉, 메시지가 들어오고 나갈때 이를 판단하여 그에 맞는 관리자를 이용하는 것이다.

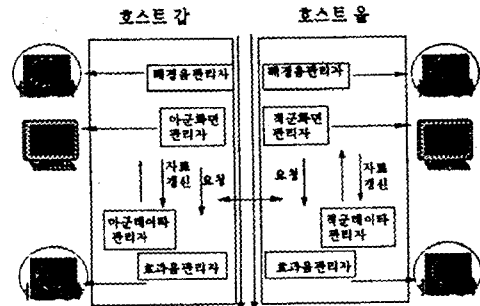


그림 9. 게임 관리자들간의 관계

각 관리자에 대한 연관관계를 보면 <그림 9>와 같다.

게임 엔진이라면 필수적인 관리자 키 입력 관리자는 윈도우환경 특성상 메시지 방식으로 제어되는 관계로 별도의 관리자를 따로 두지 않고 프로그램내의 소스코드에서 제어된다. 이는 윈도우 객체 독립적인 환경에 부합되는 것으로써 모든 제어할 일일이 지정 해주어야하는 도스환경과는 달리 윈도우용 시스템 개발자에게 큰 도움이 되는 것이다. 하지만 이와같은 특성이 때로는 시스템 개발에 불이익을 줄 경우도 있으므로 특별한 용도로 시스템을 제작하기 위해서는 입력관리자를 따로 두는 것이 바람직하다고 여기고 있다.

## 6. 결 론

현재 국내 게임 제작에 있어서는 하나의 시스템을 벗어나는 네트워크 게임이 대중적이지 못하다. 그리고 분산형 게임은 거의 찾아볼 수 없다. 본 논문에서는 클라이언트와 서버를 통합한 단일 모듈을 제안한다. 이를 이용한 분산형 통신 모듈은 아래의 기능들을 지원한다. 상위 응용 프로그래머가 통신에 관한 부분을 최대한 고려하지 않도록 하는 편의성과 발생 가능한 소켓의 상태들을 근거하여서 세가지 유형의 전략 시뮬레이션 형태를 지원가능한 범용성, 그리고 서로 다른 네트워크 거리를 감안한 이벤트 동기화 처리와 다자간 통신을 위한 통신 구조를 제안하고 있다. 이러한 통신모듈을 이용해서 분산 전략 시뮬레이션 게임인 Death Flower 를 작성하고 시험하였다. 이 결과에 관한 상세한 자료는 논문[6]을 참조바란다.

이후 연구 방향은 현재 제안하는 통신 모듈을 이용한 구체적인 다자간 통신에 대한 연구와 다양한 이벤트 처리, 그리고 실시간 처리를 연구할 것이다.

## 참 고 문 헌

- [1] W.Richard Stevens, " Unix Network Programming ", Prentice-Hall, Inc., 1991
- [2] WinG Library Help Document, Microsoft Company, 1995
- [3] W.Richard Stevens, " TCP / IP Illustrated Volume1 ", Addison - Wesley Publishing Company, 1994
- [4] Andrzej Goscinski, " Distributed Operating Systems The Logical Design ", Addison-Wesley Publishing Company, 1992, Reprinted
- [5] Ralph Davis, " Windows NT Network Programming ", Addison-Wesley Publishing Company, 1994
- [6] 노기남, 심중표, "분산형 전략 시뮬레이션 게임", 한국항공대학 이학사 논문, 1995