

# 멀티플레이어를 위한 실시간 네트워크 게임 시스템

김성후\*, 박규석\*\*  
\*창신대학 컴퓨터정보과, \*\*경남대학교 컴퓨터공학과

## Real Time Network Game System for Multiplayer

Seong-Hoo Kim\*, Kyoo-Seok Park\*\*  
\*Dept. of Computer Infomantion, ChangShin clege  
\*\*Dept. of Computer Engineering, Kyungnam University

### 요 약

기존의 비디오 게임이나 오락실 게임은 PC상에서 다양한 에뮬레이터 상에서 실행되고 있으며 단일 사용자 게임으로 진행되고 있다. 그러나 통신기술이 발전되고 게임 기술이 발전하면서 많은 부분에 변화가 일어났다. 이러한 변화는 곧 많은 게임 개발자와 게임 개발 회사들의 관심을 단일 사용자 게임에서 다중 사용자 게임으로 옮기게 되었다. 본 논문에서는 비디오 게임을 지원하는 에뮬레이터 게임에 네트워크 모듈을 지원함으로써 네트워크 게임을 할 수 있는 플랫폼을 구성하고자 하며, 서버의 부하를 분산시키기 위하여 분산 로비시스템을 도입한 peer to peer 방식의 네트워크 게임을 실행 할 수 있는 메카니즘 및 실시간 게임을 위한 동기화 방안을 제안한다.

### 1. 서론

많은 사람들이 컴퓨터가 없던 시절부터 지금까지 여러 가지 형태의 게임을 만들어 왔고 즐겨왔다. 컴퓨터의 등장 이후 수많은 컴퓨터 게임이 나왔지만 대부분 혼자서 즐기는 게임의 형태이다. 그러나 통신기술이 발전되고 게임 기술이 발전하면서 많은 부분에 변화가 일어나게 되었으며, 이러한 변화는 곧 많은 게임 개발자와 게임 개발 회사들의 관심을 단일 사용자 게임에서 다중 사용자 게임으로 옮기게 하였다.

단일 사용자 게임은 크게 PC게임 비디오 게임으로 나누어지며 다중참여 게임은 온라인 게임과 네트워크 게임으로 나누어진다. 비디오 게임을 제작하는 콘솔 업체도 현 추세에 맞추어 MS사는 X Alive를 출시하여 네트워크 게임을 지원하고 있으며, PC 게임을 온라인화한 게임으로는 대표적으로 Counter striker를 들 수 있다. 이러한 게임들은 한 종류의 게임을 온라인화한 경우이다.

기존의 비디오 게임이나 오락실 게임은 PC상에서 다양한 에뮬레이터 상에서 실행되고 있으며, 단일 사

용자 게임으로 실행되고 있다. 업계에서는 단일사용자 게임을 온라인화 시켜 서비스 하고 있으나, 기존의 다양한 게임이 활용 되지 못하고 있다. 본 논문에서는 비디오 게임을 지원하는 에뮬레이터 게임에 네트워크 모듈을 지원함으로써 기존의 모든 게임을 네트워크 게임으로 실행할 수 있도록 플랫폼을 구성하였으며, 서버의 부하를 분산시키기 위하여 분산 로비시스템 운영 방식을 도입하여 참가자들이 peer to peer 방식의 네트워크 게임을 실행 할 수 있도록 제안하였다. 그리고 실시간 게임을 실행할 수 있는 동기화 방안을 제안한다.

### 2. 관련 연구

본 절에서는 게임을 위한 네트워크 위상 모델과 기존 게임의 수행 방식에 대해 알아본다.

#### 2.1 멀티플레이어 게임의 네트워크 위상

멀티플레이어게임 네트워크 위상은 그림1과 같이

게임플레이에 지대한 영향을 끼칠 수 있는 네 가지의 일반적인 모델이 있다.

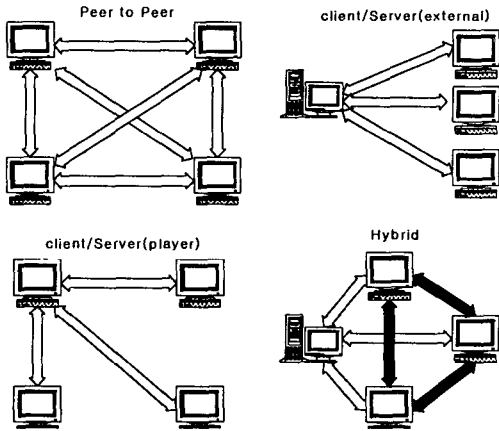


그림 1. 멀티플레이어 네트워크 게임 위상

◆ peer to peer방식의 경우, 각 플레이어는 게임의 모든 다른 플레이어들과 통신한다. 보통 외부의 서버는 따로 존재하여 대전이나 새로운 콘텐츠를 다운로드하기 위한 용도로 사용된다. peer to peer게임의 장점은 게임 서버의 디자인 및 코딩, 유지 관리가 필요 없다 것이다. 각 피어들은 서버의 중계 없이 직접 서로 통신하게 되며, 랙도 최소화된 모델이다. 단점은 대역폭의 제한으로 플레이어의 수가 제한된다.

◆ Clinet/Server(Player)방식의 경우, 게임 참가자가 서버가 되어 실행되는 방식이다. 그 참가자는 중계를 위한 전용서버가 아니며 적극적으로 게임에 관여하지도 않는다. peer to peer 모델처럼 어떤 외부서버도 유지될 필요는 없다. 그러나 peer to peer 모델과는 또 다르게 각 참가자는 연결되어진 서버(참가자)하고만 통신하기 때문에 대역폭에 대한 문제는 적다. 그러나 서버가 단절되거나 게임 종료 시 호스트를 대체하여야 하며, 이 경우에 대역폭을 최대화하고 랙을 최소화시킬 수 있는 서버를 선택하는 것이 중요하다.

◆ Clinet/Server(External)방식은 전형적으로 서버를 높은 대역폭으로 연결된 데이터센터에 두기 때문에 이상적인 대역폭을 제공할 수 있다. 그러나 이와 같은 네트워크 위상은 서버개발이나 테스트 뿐 아니라 장기간 이루어질 유지보수, 전용선 이용료 측면에서 가장 높은 비용을 소비하게 된다.

◆ Hybrid 방식은 부하분산 방식과 웹 서버 방식의 두 가지 방식이 사용된다. 부하분산 방식의 경우에는 참가자의 서버 접속을 여러 대의 접속 서버가 담당하며 메인 서버가 이를 제어하는 방식이다. 부하 분산 방식은 비록 접속 기능을 여러 대의 서버에 나누었지만 하나의 메인 서버에 의해 게임이 실행되기 때문에 참가자의 수가 증가할 경우 새로운 한계에 부딪히게 된다.

## 2.2 기존 게임의 수행 방식

◆ 비 네트워크 게임은 단독으로 게임을 하게 되는 단일 사용자 게임이다. 이러한 구조는 시나리오 상의 제약을 많이 받지만, 네트워크 부하와 같은 다른 제약 조건은 없다. 즉 탄탄한 시나리오 구성이 이루어진다면 원활히 서비스하는 데 문제가 없다.

◆ 비디오 게임은 CPU의 명령어를 패치하기 위한 하나의 디코드와 타이밍 제어를 위해 사용되는 타이밍 제너레이터, 사운드와 비디오 출력부, 그리고 조정기의 입력을 위한 I/O 인터페이스로 제공된다. 다양한 비디오 게임은 모두 이런 방식으로 게임을 실행하고 있다.

## 3. 네트워크 게임 시스템 구축

PC게임이나 비디오 게임에 네트워크 모듈을 지원하여 게임별 실행 가능한 로비 서버와 게임 클라이언트를 구축한다. 기존의 단일 사용자용 게임이 네트워크 게임의 실행을 위해서는 네트워크 모듈의 추가 가능성과 게임 실행의 일률성을 가지고 있어야 하며, 실질적인 동기화 처리 메카니즘이 지원되어야 실시간 게임이 실행될 수 있다. 비디오 게임은 콘솔게임과 오락실 게임이 주축으로 되어 있으며, 여러 에뮬레이터로 구현되고 있기 때문에 다양한 장르의 게임의 실현이 가능하다. 이러한 여러 플랫폼의 게임을 지원하여 플레이어들이 인터넷상에서 게임을 실행하기 위해서는 선택적인 게임에 따른 로비 시스템을 필요로 한다.

기존의 게임을 다중 플랫폼 방식의 네트워크 게임화 시키기 위해서는 네트워크 엔진과 서버 엔진을 SDK로 구현하여 범용화가 요구되며 네트워크 지연, 그리고 패킷 손실에 따른 적합한 네트워크 게임모델의 지원을 필요로 한다.

### 3.1 네트워크 게임을 지원하기 위한 기존의 게임 선정

기존의 PC 게임은 다양한 형식의 게임 구현으로 인하여 일률적으로 실행된다는 보장이 없다. 또한, 공개된 소스가 없다면 동기화에 무리가 따르며, 네트워크 게임을 지원하기에는 어려운 문제가 발생할 수 있다.

비디오 게임은 타이밍을 이용하는 인터럽트 개념을 사용하여 1/60초 마다 CRT칩에서 CPU쪽으로 내보내는 요구가 들어오는 형식으로, 일률적인 재 요구 처리가 가능하며, 비디오 게임을 지원하는 에뮬레이터의 소스 공개로 인해 네트워크 엔진의 탑재가 가능하다. 직접 동기화 매카니즘을 지원하여 실시간 게임 실행이 가능하며, 장르별 게임의 분류에 따라 네트워크 지연에 따른 적응형 동기화 기법을 적용 시킬 수도 있다.

### 3.2 분산 네트워크 게임 시스템 구조

본 논문에서 제안한 시스템 구조는 그림2 와 같다. 미들웨어는 게임 서버를 중재하는 역할로서 클라이언트와 TCP 컨넥션으로 정보교환과 세션 연결 처리되며, 게임서버는 각각의 다른 에뮬레이터 플랫폼을 지원하는 모델이다.

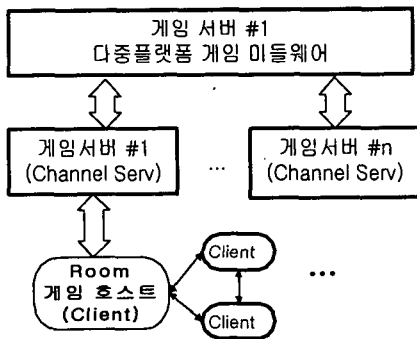


그림 2. 분산 네트워크 게임 시스템 구조

각각의 클라이언트는 게임서버에서 게임 룸을 생성함과 동시에 클라이언트가 호스트 컴퓨터로 지정되어 게임룸을 관리하게 된다.

### 3.3 네트워크 게임서버 중재

다중 플랫폼 게임 미들웨어는 사용자와 게임을 중재하는 게임서버 사이에 존재하며, 참가자는 게임 플랫폼(서버에 지원하는 기존의 1인용 게임들)을 선정하

게 되고, 사용자 로그인을 하게 된다. 사용자 로그인 처리는 인증과정을 먼저 거치게 되며, 인증된 사용자는 자기 자신의 게임 자원(게임 롬(Rom)) 이미지 리스트를 서버에 등록하게 되며, 이는 사용자간의 자원을 공유하게 될 게임 롬 이미지들이다. 또한 인증된 사용자는 다중 플랫폼 미들웨어에 등록된 게임 서버(하나의 플랫폼 기준)들의 부하 상태를 검사하여 게임 서버 리스트를 사용자에게 전송하여 주고, 사용자는 그 중 적절한 게임 서버로 연결하게 된다.

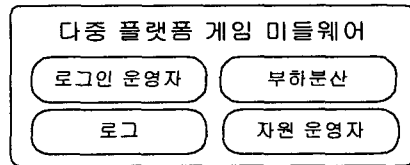


그림 3. 다중 플랫폼 게임 미들웨어 구조

자원 운영자는 게임 롬 이미지 리스트를 관리하며, 게임하고자하는 사용자가 해당되는 게임 롬이 없을 때 게임서버 또는 게임 실행 중인 사용자를 제외한 모든 사용자에게서 peer to peer 방식으로 자원을 공유하도록 관리한다.

### 3.3 네트워크 게임 서버

네트워크 게임서버는 로비(Lobby) 운영자, DB 서버, 게임 채널 운영자, Chat 운영자로 구성된다.

네트워크 게임 서버는 네트워크 상에서 게임의 상대방을 찾도록 하며, 대화 채널과 초청 기능을 가지고 있으며 게임 룸의 관리 및 생성이 가능하다. 참가자가 게임진행을 위해서는 게임 룸에 참가해야하며, 참가한 사용자는 하고자하는 게임 롬 이미지를 가지고 있지 않으면 서버 또는 사용자로부터 peer to peer 방식으로 다운로드 받게 된다. 게임 룸에 생성 또는 참가하게 되면 서버의 게임 없이 두 사람 또는 다중 참가자들 간에 서로 메시지를 주고받는 방식이다. 여기서 상대방을 찾고 게임 룸에 참가하기 위하여 로비 역할이 필요하며 담당 컴퓨터를 지정하여 로비에서 만나는 방법을 취한다.

#### ◆ 게임 채널 운영자

- 롬 이미지를 peer to peer 방식으로 다운로드 할 수 있는 자원 관리 기능을 가지고 있으며, 사용자의 자원을 통합 관리한다.
- 게임 롬 이미지에 따른 채널로 구성된다.
- 선택된 채널은 로비 운영자로 연결 설정된다.

◆ 로비 운영자

- 게임 서버를 통제하거나 또는 서버 부하를 모니터링 할 수 있다.
- 게임에 참가하고자 하는 컴퓨터의 주소 또는 id를 알 수 있다
- 사용 세션 처리를 담당한다.
- 게임의 주최자는 호스트가 되며 하나의 세션을 연다.
- 로비 역할을 담당하는 컴퓨터를 로비 서버라고 하며, 로비 서버는 중개 역할만을 담당한다.

◆ Chat 운영자

- 사용자간의 커뮤니케이션 관리를 담당한다.

그림4는 네트워크 게임의 실행 흐름도이다.

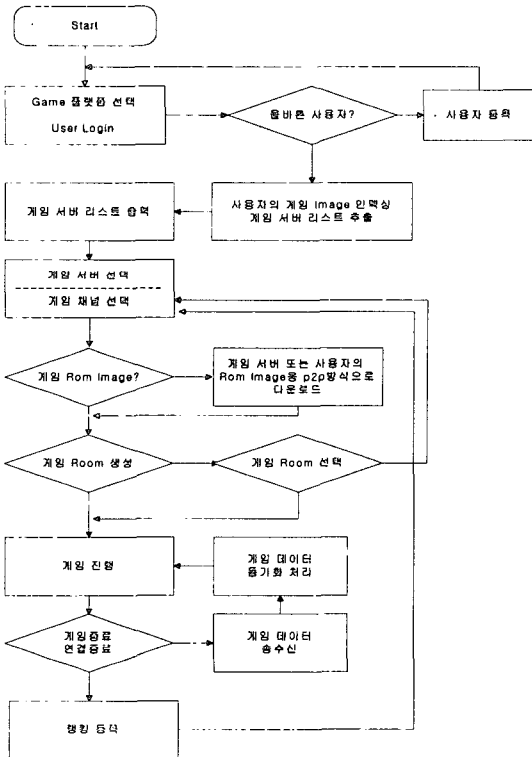


그림 4. 네트워크 게임의 실행 흐름도

Peer to peer 아키텍처인 경우 하나의 참여자가 호스트 역할을 담당하게 되며, 다른 쪽은 클라이언트가 된다. 클라이언트는 모든 입력 정보를 호스트를 포함한 다른 클라이언트에 보내게 되며 각 참여자의 게임 결과는 개별적으로 계산하여 진행된다.

따라서, 네트워크와 메시지 트래픽은 클라이언트의 수가 증가하면 기하급수적으로 증가하기 때문에 클라

이언트의 수는 자연적으로 제한된다. 호스트 역할을 하는 참여자는 현재 몇 명의 참가자가 연결되어 있는가와 같은 기본적인 매니지먼트 기능을 수행한다. 호스트의 역할을 하는 참여자가 게임에서 떠날 경우 타 참여자로 호스트의 역할을 넘기는 것도 가능하다.

4. 네트워크 전송 처리

4.1 네트워크 전송 모델

액션이나 슈팅 같은 실시간 게임을 실행하기 위하여 TCP를 이용하는 경우 상대방의 네트워크 부하나 지연이 클 경우 블로킹 현상이 발생하게 되어 매끄러운 실행을 하기가 어려우며, UDP 방식으로 게임 데이터를 전송할 경우 비동기 방식의 메시지 전달로 최대한의 속도를 유지할 수 있다. 그러나 UDP 방식의 전송은 패킷 손실 가능성이 있으며, 데이터 변형에도 문제가 있다. 이를 보완 하기위해 게임 데이터의 재전송이 가능한 히스토리 전송 매커니즘을 제안하며 CRC32 체크 방식을 지원하도록 하여 패킷 손실에 대한 재전송을 최소화 하였다.

제안 시스템은 게임 중에서 필요한 여러 가지 데이터들 중에서 반드시 중앙의 서버를 통해서만이 아닌, 특정 클라이언트에게 직접 보낼 수 있도록 하여 서버의 부하를 줄여 안정성을 보장하며, 또한 빠른 데이터 전송이 가능하다.

4.2 네트워크 게임 동기화 처리

동기화 처리는 네트워크 지연과 패킷손실에 대한 재전송 처리, 그리고 표현될 시점 등을 다음과 같이 고려해야 한다.

◆ 지연(latency)은 한 컴퓨터에서 다른 컴퓨터로 네트워크를 통해 정보의 패킷을 얻는데 걸리는 시간이며, 컴퓨터 사이를 지나가는 패킷에 의해 이용되는 경로의 함수이다. 지연시간은 네트워크 게임서버에서 참가자들의 패킷전송시간을 측정하여 2로 나눈 값을 기본으로 하여 게임실행 호스트가 sync 신호를 보내어 응답 받는 시간을 고려하여 가장 늦게 도착한 메시지에 대해 지연를 결정하게 된다.

◆ 비디오 게임을 위한 에뮬레이터의 경우는 타이밍 문제를 적절히 조절하기 위해 인터럽트 개념을 도입하여 사용하고 있다. 이 인터럽트 중 타이밍에 필요한 것은 주로 CRT칩에서 나오는 주사선 인터럽트이다.

주사선 인터럽트는 화면을 주사할 때가 되었을 때 CRT칩에서 CPU 쪽으로 내보내는 요구로서 보통 1/60초에 1번씩 발생하게 된다. 이 주사선을 사용하여 타이밍을 조절한다면 네트워크 지연에 의해 적절한 표현이 가능하다.

◆ 동기화에 처리되는 메시지의 내용은 입출력 데이터로 구성되어진다. 키보드, 조이스틱 입력은 2byte의 상태 정보로 충분하며, 동기화하기 위한 일련번호가 필요하며 패킷손실을 막기 위해 CRC32 체크섬 방식의 패킷을 보호한다.

◆ 패킷 손실 또는 변경에 대해서는 재전송으로 다시 처리해야 하는 난점이 있다. 본 논문에서 이러한 난점을 극복하기 위하여 현 데이터와 이전 데이터를 묶어서 보내는 방식으로 처리하였다. 실제 전송되는 데이터는 키보드나 조이스틱의 컨트롤 데이터가 전송되기 때문에 2byte의 상태 정보가 전송 된다. 현재 데이터 2byte와 이전 데이터 2byte를 전송하게 되면 네트워크 부하에는 별 영향을 미치지 않는다. 이러한 데이터는 패킷 손실 때 이전 데이터로 대체하게 되고, CRC32 방식으로 패킷 오류를 체크하지만 오류가 체크 되었을 경우 이전 데이터로 대체하게 된다. 이러한 방식이 효율적인 알고리즘을 가지기 위해서는 패킷 손실률에 대해 적용적으로 처리하는 방법을 고려할 필요가 있다.

히스토리 방식의 재전송 알고리즘(그림5)은 패킷 손실률에 따라 히스토리 저장 개수가 달라진다.

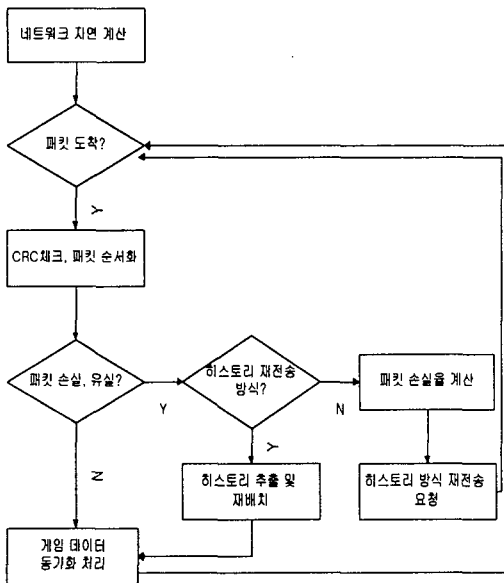


그림 5. 히스토리 방식의 재전송 알고리즘

## 5. 결론

본 논문은 기존의 단일 사용자용 비디오 게임을 네트워크 게임으로 실행 될 수 있도록 분산 서버모델로 설계하였으며, 로비 시스템 방식으로 게임을 실행하도록 하였다, 기존의 게임 애플레이터에 네트워크 모듈을 지원하여 다수 사용자가 네트워크 게임을 할 수 있기 때문에 기존의 다양한 게임의 사용이 가능하다.

Peer to peer 방식으로 네트워크 게임을 실행하도록 하여 참여자간의 실시간 게임을 실행시키기 위해 적응적 히스토리 방식의 재전송 알고리즘의 구현으로 패킷 손실 또는 유실에 따른 재전송을 최소화 한다. 3D 온라인 게임인 경우 많은 양의 데이터를 압축해서 보내기 때문에 패킷 손실 또는 유실에 따른 재전송으로 인한 네트워크 부하가 발생되지만, 제안시스템의 경우 키보드나 조이스틱의 입력 데이터만 전송하므로 히스토리 형식으로 저장 전송하여도 네트워크 부하에는 영향을 미치지 않는다. 이러한 적응적 히스토리 방식의 재전송 알고리즘이 UDP에 대한 비신뢰성 데이터를 더욱 안정화 시키고, 그에 따른 동기화로 실시간 게임이 유지될 수 있다.

향후 각각의 게임분류에 대한 예측 알고리즘이 고려되어야 할 것이며, 네트워크 엔진의 범용화를 지원할 수 있는 SDK의 개발을 위한 연구가 이루어져야 할 것이다.

## [참고문헌]

- [1] Daniel Bauer and Sean Rooney, Paplo Scotton, "Network Infrastructure for Massively Distributed Games", Netgames 2002, April, 2002
- [2] The Design of Synchronization Mechanisms for Peer-to-Peer Massively Multiplayer Games", Netgame , May, 2003
- [3] Katherine Guo and Sarit Mukherjee, "A Fair Message Exchanges Framework for Distributed Multi- Player Games," Netgame 2003 May, pp.29-41, 1998
- [4].Tom jehaes and Danny de vleeschauer, "Access Network Delay in Network Games", NetGame 2003, May, pp63-71, 2003
- [5] 김도우,정민수,이현순. "패밀리 게임기상의 유아교육 프로그램 구현", 정보과학회지, 25(1), pp 739-741, 1998
- [6] Causality and Media Synchronization Cotrol For Network Multimedaí Games,ACM NetGame 2003
- [7] Richard Rouse, "게임디자인 이론과 실제", 정보문화사, 2001