

## 온라인 게임 엔진 기술 동향

아주대학교 이만재\*

### 1. 서론

1997년 8월호의 정보과학회 특집으로 “컴퓨터 게임”을 다룬 적이 있다. 그 당시에 온라인 게임인 “바람의 나라”에 대한 소개가 이루어졌으며[1] 온라인 게임이 게임 산업에 있어 새로운 발전방향으로 제시되고 있었다. 그리고 4년이 지난 지금 온라인 게임은 예상대로 게임 산업을 주도하는 원동력이 되고 있다. 온라인 게임은 오프라인 게임에 네트워크라는 새로운 기술적 개념이 추가되어 게임 프로그램에 담겨진 인공지능과의 대결이 아닌 인간간의 게임을 가능케 하여 이전의 게임에서 다룰 수 있는 내용이나 게임을 접하는 게이머에게 새로운 경험을 제공하고 있다.

이 글에서는 온라인 게임의 발전과 함께 온라인 게임에서 사용되는 기술을 온라인 게임 엔진과 온라인 게임의 시스템 아키텍처 중심으로 살펴보도록 한다.

### 2. 온라인 게임의 발전

온라인 게임이 등장하기 이전의 컴퓨터 게임은 아케이드 게임, 콘솔 게임, PC게임으로 분류되었다. 이 중 온라인 기능은 PC게임을 기본으로 출발하였다. 그러나 온라인 기능이 지원된 것은 장르에 따라 다르게 진행되었다. 이 중 두드러진 것은 RPG(Role Playing Game), 액션 게임, 실시간 전략(RTS; Real Time Strategy) 게임의 세 가지를 들 수 있다. 장르 별로 온라인 기능에 대한 요구는 서로 달랐으며 이에 대해 설명한다.

#### 2.1 온라인 RPG 게임

RPG에서 게이머는 게임에 등장하는 캐릭터의 역할을 수행하게 되며 주어진 환경에서 홀로 또는 동료

의 도움을 받아 게임을 진행시킨다. 게이머는 주어진 시나리오에서 주인공의 역할을 맡게 되나 대부분의 경우 그룹이 함께 게임을 진행시켜야 하기 때문에 당연히 여러 게이머가 함께 참여하기 위한 온라인 월드의 필요성이 제기된다.

초기의 RPG는 PC의 성능 제한 때문에 텍스트를 중심으로 하는 MUD(Multi User Dungeon 또는 Multi User Dimension)가 사용되었다. 게임 월드는 서버에서 유지하며 게이머는 자신의 PC에서 게이머의 행동을 키보드로 입력하며 이에 대한 결과가 텍스트로 설명되는 방법이다. 그래픽 기능이 없어도 여러 사람이 참여하는 데에 새로운 게임의 즐거움이 있어 매니아 층을 구성하여 발전하였다. MUD는 PC기술의 발전으로 그래픽과 사운드를 지원하는 MUG(Multi user Graphic) 게임으로 발전하게 되며 이는 최근의 MMORPG(Massively Multi-Player Online Role Playing Game)라는 새로운 장르로 발전된다.

MMORPG의 대표적인 게임으로 울티마 온라인(Ultima Online)을 들 수 있다. 가상의 게임 월드를 만들고 게이머는 이 월드에서 자기의 분신인 아바타(avatar)를 만들어 이를 주인공으로 하여 게임 월드를 탐험하고 다른 게이머와 함께 모험을 즐길 수 있다.

MUD에서 출발한 온라인 게임은 기본적으로 하나의 서버를 유지하며 사용자의 PC는 클라이언트 역할만을 수행한다. 따라서 게임 개발을 위해서는 서버 측 소프트웨어와 클라이언트 측 소프트웨어를 별도로 취급한다. 이러한 접근 방법은 클라이언트의 수가 증가할 경우에 서버의 처리능력이나 저장능력을 늘리면 부분적으로 해결할 수 있다.

#### 2.2 온라인 액션 및 RTS 게임

RPG 게임의 경우 여러 게이머가 참여할 수 있는

\* 중신회원

게임 월드를 구성하기 위한 것이 온라인화의 목적이었다면 액션이나 RTS 게임의 경우에는 게임의 상대를 게임 내부의 인공지능이 아닌 생각하는 인간으로 바꾸기 위해 시작되었다. 액션 게임이나 RTS 게임의 경우 게임 프로그램에서 제공되는 인공지능 기능은 어느 정도 예측이 가능하여 일정 시간이 지나면 흥미를 잃게 된다. 게임에 있어 인공지능을 사람으로 대신하면 이러한 문제를 해결할 수 있다.

1996년 id software 사에서 발표한 퀘이크(Quake)는 하나의 PC에 클라이언트 기능과 서버 기능을 함께 포함시켰다. 혼자서 즐기는 솔로(solo) 게임의 경우에는 클라이언트와 서버를 하나의 PC 내부에서 직접 연결하여 사용하며, 다른 사용자와 함께 게임을 할 경우에는 하나의 PC가 서버와 클라이언트의 역할을 하고 다른 PC는 클라이언트만의 기능을 담당하도록 하는 새로운 온라인 게임의 개념을 제시하였다. 이어 등장한 RTS 게임인 Command & Conquer에서는 IPX 프로토콜을 지원하여 LAN에 있는 서로 다른 PC에서 동일한 게임을 서로를 상대로 하여 진행할 수 있도록 하였다.

IPX 프로토콜을 사용할 경우 같은 LAN에 연결된 컴퓨터만이 게임에 참여할 수 있기 때문에 상대를 쉽게 찾기 어려운 게이머에게는 불편한 점이 있었다. 따라서 IPX뿐만 아니라 TCP/IP를 지원함으로써 전세계로 연결된 인터넷에서 상대편을 찾을 수 있도록 하였다.

이후 등장한 대부분의 게임은 온라인 기능이 없는 게임을 찾기 힘들 정도로 게임의 온라인 추세는 급격히 발전하였다. 이와 같은 발전방향은 콘솔 게임에도 영향을 주어 소니의 플레이스테이션 II, 닌텐도의 게임보이, 마이크로소프트의 X-box 모두 네트워크 기능을 지원할 수 있도록 하드웨어를 구성하고 있다.

최근 온라인 게임 기술은 MMORPG를 구현하기 위한 서버 구축 및 게임 월드를 지속적으로 유지하기 위한 기술과 클라이언트에서 2D 및 3D 기술을 사용하여 게임 월드를 렌더링하는 기술, 그리고 게임 월드에 등장하는 캐릭터의 인공지능 처리를 중요 내용으로 볼 수 있다.

이러한 기술은 같은 장르의 경우 유사한 기술이 사용되기 때문에 동일한 기술을 매년 통합하여 사용하는 대신 게임 엔진을 사용하여 기본적인 기능을 통합한 다음 이를 기반으로 게임 시나리오에 따라 필요한 게임 월드를 구성하는 방향으로 게임 개발 전략이 바뀌고 있다. 온라인 게임 엔진의 발전과정을 살펴본다.

### 3. 온라인 게임 엔진

게임 엔진이라는 용어를 보편화한 게임은 퀘이크 엔진이다. 이 엔진은 1999년 소스를 공개함으로써 널리 알려졌으며 이 엔진을 기반으로 한 많은 온라인 액션 게임이 개발되었다.

#### 3.1 퀘이크 게임 엔진

퀘이크 엔진은 클라이언트-서버로 나뉘어진다 [4]. 모든 게임 로직은 서버에서 행해지며 클라이언트는 입출력만을 담당한다. 게임엔진을 구현하는 방법 중 가장 초보적인 방법은 신뢰성 있는 패킷 전송 방법만을 사용하는 것이다. 패킷이 전달되었는지를 확인하는 방법은 ack 정보를 받는 것이며 이 ack 정보를 받지 못할 경우에는 해당 액션이 지연될 수밖에 없다. 퀘이크 엔진에서는 이를 해결하기 위해 게임 진행에 있어 전달되지 않으면 안 되는 점수, 레벨 변화와 같은 중요한 정보는 reliable한 패킷을 사용하며 그 외의 정보는 unreliable한 패킷을 사용하여 보내진다. 패킷이 전달되지 않은 가능성을 감안하여 전달되는 내용은 이전과의 차이를 보내는 것이 아니라 매 번 새로운 전체 정보를 보낸다. 따라서 하나의 패킷이 전달되지 않을 경우 한 프레임 정도 화면에 변화를 줄 수 없게 되나 일반적으로 게임에서 큰 문제가 되지 않는다.

온라인 게임에서의 또 하나의 문제는 게임 입력과 이에 따른 캐릭터의 움직임에 지연이 있을 수 있다는 것이다. 예를 들어 점프 액션을 지시할 경우 이 결과를 보여 주기 위해서는 서버의 해당 계산 결과를 패킷으로 받아 클라이언트에서 처리하여야 한다. 게이머가 함정을 발견하고 점프했다는 정보가 느리게 전달된다면 함정에 빠진 것으로 보일 수도 있기 때문이다. 이런 문제를 해결하기 위해 게임 로직은 서버와 클라이언트에서 동시에 수행한다. 둘 사이의 결과가 일치할 경우에는 레그(lag)가 발생하지 않는 효과를 얻을 수 있지만 두 결과가 다를 경우에는 명중해서 죽은 적이 다시 살아나는 것처럼 보이는 모순이 있을 수 있다.

이와 같은 메시지 처리를 위해 모든 메시지는 UDP를 사용한다. 따라서 reliable 한 메시지를 만드는 작업은 게임 로직에서 처리한다. 퀘이크 엔진의 특이한 점은 네트워크 메시지 처리의 일부를 Quake-C라는 스크립트 언어에 맡긴다는 점이다. 대부분의

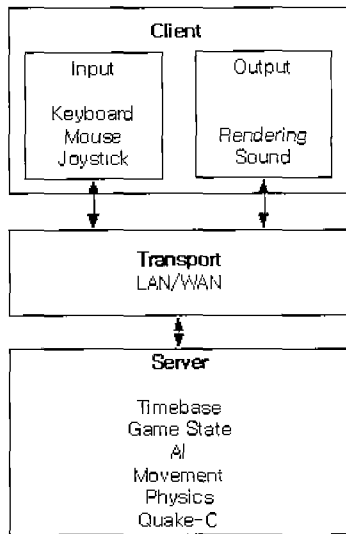


그림 1 Quake 아키텍처

기능은 엔진을 코딩한 C 언어에서 처리하지만 게임 레벨과 관련있는 메시지의 처리는 Quake-C에서 처리하도록 하여 게임 제작의 편의를 제공한다.

### 3.2 언리얼 게임 엔진

언리얼(Unreal) 게임은 웨이크 엔진에서 제시한 온라인 게임 엔진의 기본적인 특성을 크게 개선하였다[5]. 웨이크 엔진에서는 새로 렌더링해야 할 오브젝트의 리스트와 움직임에 따른 궤적의 정보를 보내도록 되어 있다. 그러나 이러한 것만으로는 클라이언트와 서버의 로직이 서로 다를 가능성이 있기 때문에 스테이트(state)라는 개념을 추가하였다.

언리얼 엔진의 클라이언트는 게임 스테이트의 서브세트(subset)를 유지하고 있어 해당되는 스테이트의 로직을 수행함으로써 클라이언트와 동일한 루틴을 수행할 수 있도록 한다. 이는 객체지향 프로그램의 장점을 활용한 것이다. 오브젝트의 스테이트는 객체지향 개념을 최대한 살린 언리얼스크립트(Unreal Script)라는 스크립트 언어를 통하여 구현함으로써 게임 로직과 게임 엔진을 분리하도록 하였다.

온라인 게임에서 서버와 클라이언트 간의 메시지 전달을 위한 메시지의 양은 최대한 줄이도록 해야 한다. 이를 위해 클라이언트는 자체로 보유하고 있는 게임 스테이트를 사용하여 최소한의 서버의 도움을 받는 상태에서 게임 로직을 구현한다. 따라서 서

버의 상태는 클라이언트에 복제(replication)하여 사용한다. 게임에 등장하는 이동이 가능한 오브젝트를 액터(actor)라 정의하였으며 클라이언트는 서버에서 유지하는 액터의 일부를 복제할 수 있도록 한다. 또한 서버에서 유지하는 액터 변수도 부분적으로 클라이언트에 복제되어 사용하도록 한다. 마지막으로 서버에서 수행되는 함수도 부분적으로 클라이언트에서 복제되어 수행한다. 이러한 복제기능을 통하여 메시지의 전송을 최대한으로 줄이도록 하여 게임의 진행을 빠르게 한다.

표 1 언리얼 엔진의 액터 우선순위

액터	우선순위
적 캐릭터	8.0
탈것	7.0
탄환	6.0
보조 캐릭터	4.0
장식적인 캐릭터(예: 물고기)	2.0
장식	0.5

또한 게임에 등장하는 액터에 우선순위를 부여하여 우선순위가 높은 액터는 보다 자주 데이터를 수정하도록 한다. 구체적인 액터의 우선순위는 표 1과 같다. 표에서 우선순위 2.0인 액터는 우선순위 1.0인 액터보다 2배 이상 자주 상태를 업데이트하도록 한다.

언리얼 엔진에서 사용하는 언리얼스크립트 언어는 객체지향언어로 스크립트 언어로 표시하는 게임에 등장하는 캐릭터의 인공지능 기능을 프로그램으로 구현할 수 있도록 한다. 스크립트 언어이지만 고급 객체지향 언어에서나 지원하는 상속 개념을 사용하며 앞서 설명한 스테이트 개념을 언어에서 직접 지원하기 때문에 게임 로직에 충실할 수 있다.

### 3.3 토르크 게임 엔진

토르크(Torque) 엔진은 Dynamics사의 Tribes II 게임[6]에서 사용된 엔진으로 2001년 Garagegames를 통해 \$100이라는 가격으로 공급되고 있다. 이러한 저렴한 가격정책의 뒷면에는 게임 엔진을 사용하여 개발한 게임의 판매는 Garagegames를 통해 서반 가능하다는 조건을 달고 있다. 이전 게임 엔진보다 게임 참여자의 수를 128명으로 높여 액션 게임에서 대규모의 그룹 플레이를 가능하게 하였다. 이를 위해서는 네트워크 기능에서 보다 효율적인 프로토

를 설계를 필요로 하였으며 다음과 같은 네 가지 데이터를 정의하였다. 이는 언리얼 엔진에서 사용된 액터의 스테이트와 우선 순위와 유사한 개념으로 볼 수 있다.

표 2 토르크 엔진의 메시지

구분	설명
Non-guaranted data	unreliable 메시지와 동일함
Guaranted data	reliable 메시지와 동일함
Most recent state data	항상 최근의 데이터만이 의미를 갖는 패킷으로 패킷 손실의 경우 최근의 데이터를 재전송
Guaranted Quickest data	가장 빠른 시간에 도달해야 하는 패킷

클라이언트와 서버는 그림 2와 같은 관리자료를 사용한다. 이 중 고스트(ghost) 관리자는 호스트의 오브젝트를 클라이언트에 복사한 내용을 말하며 Most Recent state data를 사용하여 서버와 데이터를 동일하게 유지한다. 네트워크 메시지를 줄이기 위해 고스트는 스코프(scope) 규칙을 가지고 있어 고스트가 가까이 접근할 경우에만 해당 메시지를 교환하도록 한다.

동작 스트림 관리자(Move Stream manager)는 기본적으로 Guaranteed quickest data를 사용하여 해당 플레이어의 위치를 표시하도록 하나 네트워크의 특성에 따라 guaranteed를 사용하기도 한다.

데이터블록(Datablock) 관리자는 비교적 자주 바뀌지 않는 정적인 오브젝트를 담당하며 Most recent data를 사용한다. 일반적으로 데이터블록은 클라이언트에 한 번만 전송되고 거의 변화가 없을 경우가 많다. 데이터블록의 전송은 시뮬레이션이 거의 정지된 상태를 사용하기 때문에 네트워크에 부담이 거의 없다.

위에 본 몇 가지 예를 본다면 토르크 게임에서는 게임에 사용되는 오브젝트 간의 메시지를 최대한 줄이도록 여러 가지 노력을 한 것을 볼 수 있다.

토르크 엔진의 뛰어난 점은 게임전체의 편집을 하나의 편집기에서 모두 처리할 수 있도록 한 기능이다. 이 편집기는 게임 진행사에서 바로 열 수 있다. 이 편집기에는 지형의 변화, 인공지능 기능의 변경 등 이전의 편집기와 다른 발전된 모습을 보여 준다.

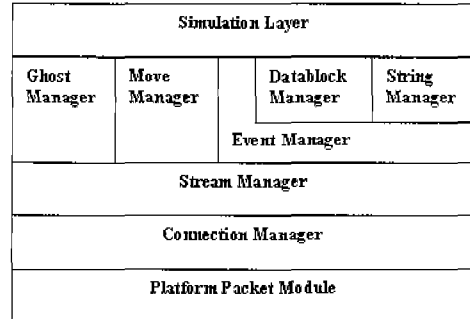


그림 2 토르크 엔진의 네트워크 연결

### 3.4 터바인 게임 엔진

앞서의 게임 엔진들이 액션 게임 엔진인데 반해 터바인 엔터테인먼트사에서 개발한 터바인(Turbine) 엔진[7]은 MMORPG를 위한 게임 엔진이다. 이 엔진은 애쉬론즈 콜(Asheron's call)이라는 게임에 사용되었다. MMORPG는 넓은 면적에서 여러 게이머가 참여하여 발생하는 가상공간의 시뮬레이션으로 볼 수 있다. 따라서 게임 엔진의 가장 핵심이 되는 기능은 넓은 면적을 어떻게 구현할 수 있는냐 하는 것이다. 터바인 엔진에서는 수백 평방 마일의 면적을 하나의 지역(region)으로 정의하고 게임 월드는 이러한 지역의 연결로 구현한다. 한 지역에서 다른 지역으로의 이동은 다른 서버로 해당 캐릭터를 옮기는 작업으로 수초 정도의 지연시간을 필요로 한다.

터바인 엔진에서 클라이언트는 지속적으로 변화할 수 있는 게임 월드를 구현하기 위해 DDD(Dynamic Data Downloading)라는 기법을 사용한다. 이 방법을 사용하기 이전의 온라인 게임은 게임 내에 새로운 임무가 주어질 경우 이를 패치(patch)를 사용하여 게임월드와 변화를 클라이언트에 구현하도록 하였으나 DDD기법을 사용할 경우 게이머는 이러한 패치작업이 보이지 않게 이루어진다. 이러한 기법을 사용할 경우 게임 내에서 특별한 이벤트를 만들어 내는 것이 가능하다. DDD기술의 장점은 게임의 판매에 있어 새로운 모델을 제시한다. 현재 게임은 수백 MB 이상을 차지하기 때문에 CD 형태로 판매하고 이를 바탕으로 시작한다. DDD를 사용할 경우 아주 작은 용량으로 출발하는 지역에 대한 데이터만 제공하고 탐험 과정에서 추가적인 데이터를 제공한다면 CD 구입 없이 직접 온라인 게임에 참여하는 것이 가능해진다.

## 4. 온라인 게임 시스템 아키텍처

온라인 게임의 정의와 일반적인 분류에 대해서는 정보과학회지 온라인 3D 기술 특집[2]에서 이미 자세히 정리한 바 있다. 여기서는 온라인 시스템의 아키텍처에 따른 분류를 제시하고 각각의 특징을 살펴 도록 한다.

### 4.1 피어-투-피어(Peer-to-peer) 아키텍처

실시간 전략게임이나 액션 게임의 경우 게이머는 네트워크 상에서 게임의 상대방을 찾도록 하며 상대방이 발견된 다음에는 다른 컴퓨터의 개입 없이 두 사람의 컴퓨터 간에 서로 메시지를 주고 받는 방식이다. 참여하는 컴퓨터의 수는 꼭 들어야 할 필요는 없으나 대역폭 이하일 경우가 대부분이다. 이 경우 중요한 것은 어떻게 상대를 찾는다는 것이다. 이를 위해서 로비(lobby) 역할을 담당하는 컴퓨터를 지정하여 로비에서 만나는 방법을 취한다. 로비의 주소는 미리 정해져 있어 로비와는 쉽게 연결할 수 있으며 로비에서 게임에 참여하고자 하는 컴퓨터의 주소 또는 id를 알 수 있다. 이 중 게임을 주최하겠다고 선언하면 그 컴퓨터는 호스트(host)가 되며 하나의 연결 세션(session)을 연다. 어느 PC나 호스트가 될 수 있으나 이미 열린 호스트의 세션에 참여할 수도 있다. 일정한 수의 컴퓨터가 세션에 참여하게 되면 호스트는 로비와의 연결을 끊고 게임 세션을 연다. 앞서의 연결 세션에 참여하겠다고 선언한 컴퓨터는 새로운 게임 세션에 참여하는 방식으로 게임이 시작된다. 이 경우 로비 역할을 담당하는 컴퓨터를 로비 서버라고 하나 로비 서버는 중개역할만을 담당할 뿐이다.

대중적인 게임을 예를 들어 아키텍처를 설명하면 다음과 같다. 스타크래프트 게임은 네트워크 플레이 방식은 피어-투-피어 아키텍처이며 로비를 제외하고는 서버의 역할이 불필요하다. 따라서 인원의 컴퓨터 2대만 LAN으로 연결되어 있으면 바로 온라인 게임을 진행할 수 있다. 디아블로 II의 경우에는 오픈 서버와 렐름(Realms) 서버의 두 가지 방식의 서버를 사용할 수 있는데 오픈 서버 방식 역시 피어-투-피어 방식을 사용한다.

피어-투-피어 아키텍처인 경우 하나의 PC가 호스트 역할을 담당하며 다른 컴퓨터는 클라이언트가 된다. 클라이언트는 모든 입력정보를 호스트를 포함한 다른 클라이언트에 보내게 되며 각 PC는 게임의 결

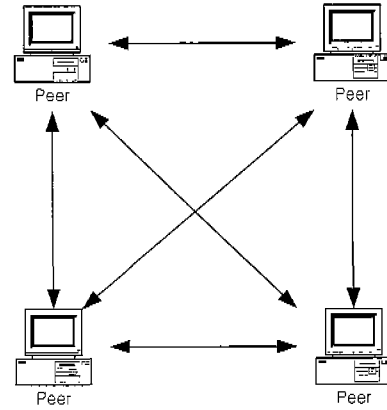


그림 3 피어-투-피어 아키텍처

과를 개별적으로 계산하여 진행한다. 따라서 네트워크의 메시지 트래픽은 클라이언트의 수가 증가하면 기하급수적으로 증가하기 때문에 클라이언트 수는 자연적으로 제한된다. 호스트 역할을 하는 PC는 현재 몇 명의 게이머가 연결되어 있는가와 같은 기본적인 매니지먼트 기능을 수행한다. 호스트의 역할을 하는 PC가 게임에서 떠날 경우 타 PC로 호스트의 역할을 넘기는 것도 구현방법에 따라 가능하다.

### 4.2 클라이언트-서버 아키텍처

클라이언트-서버 아키텍처에서는 서버를 담당하는 컴퓨터는 서버 역할만을 지속적으로 수행한다. 경우에 따라서는 하나의 컴퓨터가 서버와 클라이언트의 역할을 동시에 수행할 수도 있으나 클라이언트의 수가 증가할 경우에는 전용의 서버가 사용되기도 한다. 클라이언트 서버 아키텍처를 사용할 경우 모든 클라이언트는 게이머의 입력을 서버에 보내며 서버는 이러한 입력을 기반으로 게임의 모든 진행 결과를 전체 클라이언트에게 브로드캐스팅한다. 앞서 설명한 피어-투-피어 아키텍처에 비해 메시지의 수는 클라이언트의 수에 비례하기 때문에 비교적 많은 수의 클라이언트를 수용할 수 있다.

또한 게임의 상태는 서버에만 보관되기 때문에 RPG와 같이 게임 진행 상황을 여러이 공유해야 하는 경우 적합한 아키텍처이다. 디아블로 II의 경우 렐름 서버는 클라이언트-서버 구조의 서버로 여기에는 온라인 게임에 참여한 클라이언트의 상태가 보관된다. 디아블로 II와 같은 인기가 높은 게임의 경우 하나의 대형 서버에 소 그룹 단위의 여러 게임을 지원

한다. 그러나 대형 서버를 사용하더라도 하나의 서버로 많은 게이머를 만족할 수 없어 여러 개의 랙룸 서버를 사용한다. 이 경우 하나의 랙룸 서버에서 성장한 캐릭터는 다른 랙룸 서버로 이전할 수 없다.

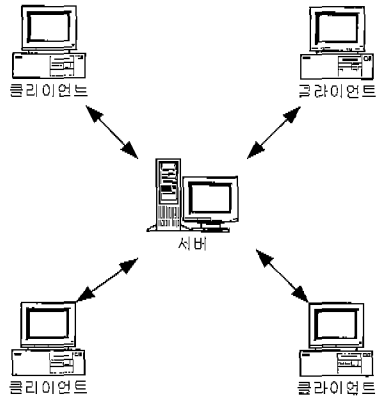


그림 4 클라이언트-서버 아키텍처

### 4.3 분산 서버 아키텍처

클라이언트-서버 아키텍처의 경우 많은 수의 클라이언트가 참여할 수 있으나 MMORPG와 같이 하나의 게임에 수백 내지 수천명이 참여할 경우 하나의 서버로 이를 담당하기 어렵게 된다. 따라서 서버의 작업량을 여러 서버에 나누는 방식이다. 분산 서버 방식에는 부하분산 방식과 맵 서버 방식의 두 가지 방식이 사용된다.

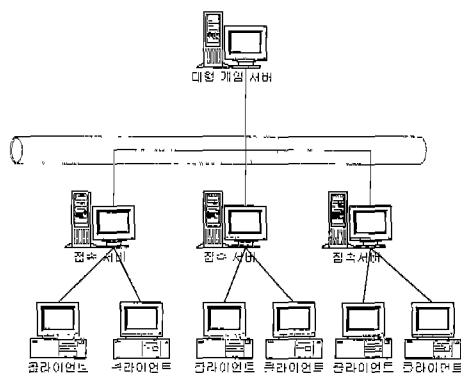


그림 5 부하 분산 방식 서버 아키텍처

부하 분산방식의 경우에는 게이머의 서버 접속을 여러 대의 접속 서버가 담당하며 메인 서버가 이를 제어하는 방식이다. 부하 분산 방식은 비록 접속기능

을 여러 대의 서버에 나누었지만 하나의 메인 서버에 의해 게임이 진행되기 때문에 게이머의 수가 증가할 경우 새로운 한계에 부딪히게 된다.

맵 서버 방식은 게임 월드를 여러 개의 맵으로 나누고 게이머가 어느 맵에 있는가에 따라 해당 맵을 담당하는 서버에 접속하게 하는 방법이다. 게임 월드가 넓어질 경우 서버의 수를 증가시키면 되기 때문에 더 발전된 방식이다. 그러나 맵의 경계를 넘게 되면 해당되는 캐릭터를 다른 서버로 보내 주어야 하기 때문에 게임 진행이 지연될 가능성이 있다.

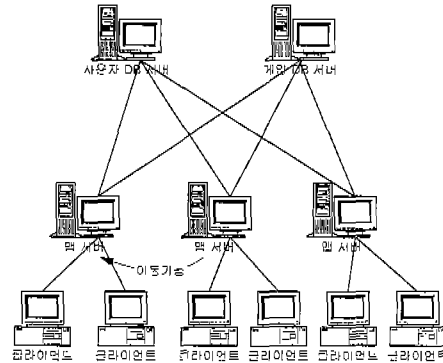


그림 6 맵 서버 방식 분산 아키텍처

## 5. 맺음말

지금까지 온라인 게임의 발전과정을 게임 엔진과 시스템 아키텍처 입장에서 살펴보았다. 온라인 기술의 발전과정에서 볼 수 있듯이 네트워크의 메시지 전송의 효율적인 설계 및 온라인 게임 월드의 구현을 어떻게 편리하게 할 수 있는가 하는 것이 온라인 게임 기술의 핵심이라 할 수 있다.

현재 논의되고 있는 온라인 게임은 피어-투-피어 방식과 클라이언트-서버 아키텍처를 주로 사용하고 있다. 온라인 게임의 중요성이 높아지면서 게임에 필요한 운영체제 기능을 DirectX라는 이름으로 통합하여 지원하는 마이크로소프트는 DirectPlay라는 모듈로 온라인 기능을 부분적으로 지원하고 있다. 그러나 DirectX는 PC의 기능 지원에 한정되기 때문에 피어-투-피어 방식의 온라인 게임에 대한 지원만 가능하며 클라이언트-서버 방식의 서버에 대한 기능은 개발자 스스로가 정의하며 기술을 개발하고 있다.

국내의 온라인 게임 기술은 세계적인 수준으로 알려져 있으나 논문으로 발표되는 등 이를 체계적으로

정리할 기회가 주어지지 않은 상태이다. 이는 게임 관련 학술대회가 활발하게 이루어지는 외국과 경쟁할 경우 장기적으로 취약점이 될 가능성이 높다. 게임에 관련된 컴퓨터 공학자의 게임 관련 연구의 참여가 절실하다.

### 참고문헌

[1] 김정주, "온라인 게임 '바람의 나라'", 정보과학회지, 제 15권 8호, 1997.8, pp33-41.  
 [2] 최진성 외, "온라인 3D 게임 기술", 정보과학회지, 제 19권 5호, 2001.5, pp59-68.  
 [3] Todd Barron & Andre Lamothe, Multiplayer Game Programming, Prima Tech, 2001.  
 [4] Michael Abrash, Quake's 3-D Engine: The Big Picture, <http://www.gamedev.net/reference/articles/article987.asp>  
 [5] Tim Sweeney, Unreal Networking Architecture, <http://www.epicgames.com/Network.htm>  
 [6] Mark Frohnmayer, Tim Gift, "The Tribes

Engine Networking Model", CGDC 2000 Conference Proceedings, March 2000, pp191- 207.  
 [7] The Turbine Engine 2.0, <http://www.turbine-games.com/tech.htm>  
 [8] 노홍식, "온라인 게임 개발에 있어 신생업체의 문제점", 2001년 하계 한국게임학회 학술발표대회 논문집, 2001.7.

### 이 만 재



1970 서울대학교 공과대학 전기공학과 학사  
 1973 한국과학기술원 전자계산실 연구원  
 1982 스탠포드대학 전기공학 석사  
 1986 텍사스 오스틴대학 컴퓨터공학 박사  
 1986 한국전자통신연구소 책임연구원  
 1989 솔빛미디어 대표이사  
 1995 숙명여대 정보방송학과, 전산학과 교수  
 1998~현재 이주대학교 미디어학부 교수

관심분야: 컴퓨터 게임, 디지털 미디어  
 E-mail: manjai@madang.ajou.ac.kr

### ● 제12회 고속통신망 공동워크숍(HSN 2002) ●

- 일 자 : 2002년 2월 20 ~ 22일
- 장 소 : 경주 현대 호텔
- 내 용 : 초청강연, 논문발표 등
- 주 최 : 정보통신연구회, 대한전자공학회, 한국통신학회
- 문 의 처 : 동국대학교 컴퓨터정보통신공학부 이재훈 교수  
Tel. 02-2260-3849