분산 게임 서버 시스템 설계에 관한 연구

정회원 배 재 환*

A Study on Design of a Distributed Game Server System

Jae-Hwan Bae* ***Regular Member***

요 약

네트워크 게임의 특성상 다수의 이용자들이 서버에 집중되게 되면 서버의 부하가 걸려 이용자들의 불편함을 초래함은 물론, 서비스 제공자 측면에서도 많은 비용을 들여서 서버를 증설하여야 하는 문제가 발생하게 된다. 본 논문은 클라이언트의 서버에의 의존성을 최소화시키기 위한 복합 분산 서버 시스템을 제안하고자 한다. 이 시스템은 서버에 의존하여 메시지를 송·수신 하는 기존 방법을 탈피하여 정보갱신 및 저장의 경우에만 서버에 의존하고, 대부분의 메시지 처리는 클라이언트 간에 이루어 질수 있도록 설계하였다. 제안하는 설계 방법은 다음과 같이 요약된다. 첫 번째로 메시지 등급화 처리방법을 제안하여 메시지의 중요성에 따라 서버나 클라이언트가 관리를 할 수 있도록 하였으며, 두 번째로 Client-Server 방식과 Pear-to-Pear 방식을 병행하여 효율성을 확보할 수 있도록 하였다.

ABSTRACT

As the Internet continues to grow, network games are widely spreaded. For most network games, many users' meet on a server causes a heavy load to the server, which in turn brings inconvenience to the user. Moreover, it demands increased expense to the service provider for deploying additional servers. In this paper, we propose a hybrid distributed system for network games. In our proposed system, a client is independents of the server and exchanges information directly with other clients. The client depends on servers only for the update information. The proposed methodology classifies messages according to the characteristics of information that the message handles and applies either client-to-server and pear-to-pear communication for processing messages which increases the efficiency of systems.

Ⅰ. 서 론

미래의 게임을 발전시킬 가장 큰 원동력의 세 가지 요소는[12] 3차원 그래픽, 네트워크, 가상현실 기술로 집약되는데, 이중 네트워크 기술은 차세대 네트워크 게임의 필수적인 요소기술로서 현재 많이 연구되고 있다. 초고속정보통신망의 확충 및 보급에 따라서 인터넷을 통한 온라인 게임이 널리 이용되고 있다. 하지만 인터넷 게임의 특성상 다수의 이용자들이 서버에 집중되게 되면 서버의 부하가 걸려 이용자들의 불편함을 초래함은 물론, 서비스 제공자 측면에서도 많은 비용을 들여서 서버를 증설하여야 하는 문제가 발생하게 된다[1,2]. 이에 본 논문은 기존의 네트워크 구성 시스템에서 일어나는 병목현상을 효율적인 메시지 관리 및 전송을 통해 서버의 부하를 줄이고 클라이언트의 부하를 높이는 복합 분산 서버 시스템을 제안 하고자 한다. 제안하는 복합 분산 서버 시스템은 비대칭 분산 서버 시스템에서의 동기화 문제, 통신 속도 및 비용문제를 해결하기 위한 것으로써 일대 일 연결이 아닌 다중 연결 구조를 취해 메시지를 송· 수신하게 되며, 메시지 비동기화 처리 문제를 위해 멀티 메시지 큐(Multi

*탐라대학교 정보출판미디어학부 컴퓨터게임학전공(bjhmail@korea.com)
논문번호 : 030239-0602, 접수일자 : 2003년

Message Que)를 운영하고, 속도향상을 위해 중요도에 따라 나눈 3 레벨 메시지 처리 방법을 제안하고 있다. 이 방법의 변화로 시간적 동기화나 무결성의 문제, 동시성의 문제 등을 상당량 해결할 수 있다.

본 논문의 구성은 2절에서는 네트워크게임의 현황에 대해서 살펴보고, 3절에서는 일반적인 네트워크 게임 서버 구조, 4절에서는 제안하는 분산 서버 시스템에 대해 각각 설명하고, 마지막으로 5절에서 결론을 맺고자 한다.

II. 네트워크 게임의 현황

최근 전화 및 인터넷 등의 통신망을 통하여 다수의 사람이 동시에 참여하여 즐길 수 있는 네트워크 게임이 그 기반을 넓혀가고 있다[12,13]. 국내에서의 네트워크 게임은 전화를 통 해 기존 PC 통신망에서 게임을 즐기는 몇 년 전부터 등장한 텍스트 기반 머드(MUD: Multi-User Dungeon) 게임들을 비롯하여 최근의 몇 가지 그래픽 머드 게임에 이르기까지 다른 나라에 비해서 특히 머드게임 분야에서 많이 등장했으며, 그 이외의 네트워크를 이용한 온라인 게임 분야에서는 아직도 미약하다고 볼 수 있다. 하지만 현재의 대부분의 네트워크 게임이 지니는 한계인 사용자 수에서 머드는 타 기술에 비해서 많은 사용자를 수용할 수 있다는 강점을 가진다. 현재 국내외에서 소개되고 있는 온라인 게임은 크게 IPX(Internet Packet Exchange)를 기반과 TCP/IP를 기반 게임으로 분류된다.

1. IPX 기반의 네트워크 게임

IPX 기반의 네트워크 게임은 근거리 통신망을 기반으로 10여명 정도의 사용자를 위한 게임이다. 이러한 게임은 현재 국제적으로 수십 종이 넘게 출시되어 있으나 현재 국내에서는 이 분야에서 두각을 나타내는 게임이 출시되고 있지 않다. 대부분의 IPX 기반의 게임의 경우에는 게임에 참여한 한 사람의 컴퓨터가 서버처럼 동작하며 이 서버에 의해서 전체 게임의 동기가 조절되며 사용자의 수가 제한적인 반면에 전체 네트워크의 부하 가중에 의한 속도저하로부터는 비교적 자유롭다. 따라서 TCP/IP를 기반으로 한 네트워크 게임보다 서비스에 필요한 자원이 적게 소요된다.

2. TCP/IP 기반의 네트워크 게임

TCP/IP를 기반의 네트워크 게임은 인터넷을 기반으로 구축되어 전 세계의 수많은 사람들이 동시에 접속하여 수행할 수 있는 게임이다. 동시 사용자 크기에 의하여 요구되는 서버과 네트워크 용량(Bandwidth)의 크기가 결정된다.

국내외에서 소개된 몇 개의 TCP/IP 기반의 온라인 게임을 들자면 다음과 같다. 국내에서 사용중인 바람의 나라[3], 어둠의 전설[4], 및 영웅문[5]은 그래픽 머드 게임이고, 아크메이지[6]는 웹에 기반 한 온라인 전략 게임이다. 울티마 온라인[7]은 울티마 시리즈의 온라인 게임버전으로 현재 인터넷상에서 서비스중이다. 이 외에도 여러 회사가 인터넷을 통해 온라인 게임을 서비스하고 있다.

근래에 들어 미국에서는 기존 비온라인 게임들을 인터넷을 통해서 제공하기 위한 T[EN[8], Mplayer[9]. Battle.net[10]. 및 Internet Gaming Zone[11] 등의 다수의 게임 서비스 회사들이 등장하였다. 이러한 네트웍 게임 서비스 회사들은 최신 네트웍 게임 타이틀과 다양한 커뮤니티 기능을 통해 3D 액션게임, 실시간 전략게임, 롤플레잉게임, 스포츠게임, 시뮬레이션게임 그리고 각종 카드 게임과 보드 게임들을 서비스하고 있다. 이러한 경향은 기존의 PC 통신망에서 벗어나 인터넷을 통한 직접적인 서비스를 통하여 보다 많은 사용자에게 다양한 게임 환경을 제공하려는 데에서 기인한다.

III. 일반적인 네트워크 게임 서버 구조

1. 단일 게임 서버 구조

하나의 서버가 여러 개의 클라이언트를 지원하는 중앙 집중형의 단순 방식으로, 다수의 클라이언트의 처리에는 적합하지 않으므로 소수의 클라이언트로 구성되는 단순한 게임의 구현에 이용된다. 실제로 하나의 서버가 서버기능을 수행하므로 서버 간의 시간적 동기화문제(Clock Synchronization)나 교착상태(Deadlock), 동시성(Concurrency), 무결성(Integrity) 등의 문제가 쉽게 해결될 수는 있지만, 서버가 데이터베이스관리, 소켓관리, 인증처리, 게임 콘텐츠 서비스 등 모든 부분을 맡아서 처리 하고 있기 때문에 다수의 클라이언트를 감당하는데 문제가 있다.

2. 멀티 게임 서버 구조

멀티게임서버구조는 서버들이 일의 역할별로 나누어 처리하는 형태로써, 로그인서버, 업데이트서버, 게임콘텐츠 서비스서버, 자동갱신서버, 트랜잭션서

버, 데이터베이스서버, 웹서버, 모니터링서버, 빌링 서버등 세부적으로 분리되어 처리될수록 서버간의 전송 트래픽은 높아지지만 그만큼 많은 수의 클라이언트를 수용할 수 있다[14,15]. 멀티게임서버구조는 분산 네트워크 서버라고도 불리 우며, 분산 시스템 환경으로 인해서 발생할 수 있는 문제점들이 많으므로 많은 기술이 적용되어야 한다. 멀티게임서버구조는 의존성에 따라 대칭 서버(Replicated Server)와 비대칭 서버(Non Replicated Server)로 구분 할 수 있다.

2.1 대칭 서버 구조(Replicated Server)

대칭 서버 구조는 서로 다른 서버의 플레이어들 간의 의존성이 없을 때 사용하는 구조이다(그림 1). 보드게임이나 턴 방식의 네트워크 게임에 사용된다. 대칭 서버 구조는 최대 수용할 수 있는 클라이언트 수 만큼만을 할당받아 관리하므로 서로 다른 서버에 접속한 플레이어들은 정보를 주고받기가 힘들다.

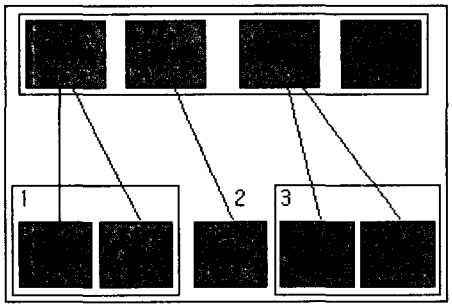

그림 1. 분산 네트워크 대칭 서버 시스템

그림 1에서 1번, 2번, 3번 그룹의 클라이언트들은 서로간의 연결성이 없다. 1번 그룹의 클라이언트 사이에만 서로 의존성을 갖게 된다. 이것은 서버간의 정보 공유를 하지 않기 때문이다.

2.2 비대칭 서버 구조(Non-Replicated Server)

비대칭 서버 구조는 서로 다른 서버의 플레이어들 간의 의존성이 존재할 경우에 사용하는 구조이다(그림 2). 대부분의 머그 게임이 이 방식을 따르고 있으며, 클라이언트의 영역 위치에 따라 서버가 담당하여 처리하는 방식이 가장 많이 사용되고 있다. 하지만 같은 영역에 많은 플레이어가 몰리면 그 영역을 담당하는 서버에 많은 부하가 발생되는 문제점이 있다. 이러한 비대칭 서버 구조에서 해결해야할 주요 문제점들은 아래와 같다.

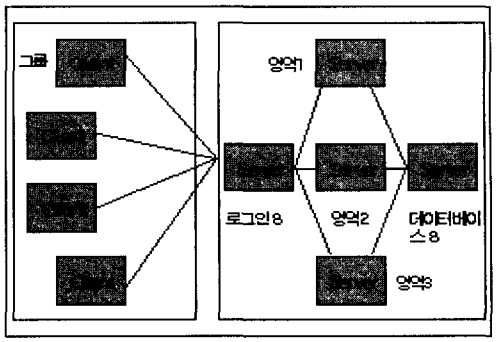

그림 2. 분산 네트워크 비대칭 서버 시스템

1) 시간적 동기화

실시간 네트워크 게임에서 가장 중요한 것은 동기화 문제이다. 같은 시간대에서 A라는 클라이언트보다 B라는 클라이언트가 약 3초정도 반응이 늦게 온다면 큰 문제가 발생하게 된다. 시간적 동기화는 매우 중요하므로 서버는 이를 받아서 틱(Tick)을 이용하게 되는데 완전히 해결할 수는 없다. 그나마 도움이 되는 대표적인 방법은 데드 레커닝(Dead Reckoning) 기법을 적용하는 것이다. 그리고 Ping을 이용하여 평균시간을 계산하고 지연(Latency)을 고려하여 설계하면 오차를 줄일 수 있다. 하지만 클라이언트 서버시스템은 서버가 소켓관리와 DB관리를 맡게 되므로 소켓관리를 하면서 시간동기화를 위한 계산을 같이 해야만 한다. 이것은 서버에게 많은 부하를 일으키게 하고 그만큼 많은 수의 클라이언트를 감당하기 어렵게 만든다.

2) 메시지 전달의 비동기화

좀더 사실적인 게임형태를 갖추기 위해 리얼타임을 요구하는 게임을 개발하기 위해서는 메시지 전달이 동기화 되어서는 안 된다. 메시지를 송·수신했다면 시스템은 바로 다른 일을 하면서 메시지 큐(Message Que)를 관리하도록 설계되어야만 한다. 하지만 메시지 큐를 이용한 비 동기 통신 방법은 여러 가지 상황을 모두 고려하여 구현하여야 하므로 쉽지 않은 문제점이 있다. 또한 메시지 큐의 경우도 메시지가 언제 만들어지고 보내진 메시지인지 분명히 하기 위해 타임스탬프를 적용해야 오차를 줄 일수 있다. 이부분에 대해서는 각 회사마다 다양한 방식을 적용하고 있으며, 심지어 클라이언트가 서버로부터 응답 메시지를 받지 못해도 받은 걸로 가정하고 실시간으로 진행하게끔 만들기도 한다.

3) 물리적 통신 환경

분산 네트워크 서버 환경을 구축하게 되면, 서버 시스템에는 많은 대역폭을 가진 선로가 필요하게 된다. 또한 서버간의 내부 네트워크를 위해 내부 선로도 고속화되어야 하며, 서버의 확장 및 관리와 유지하는데 있어서도 많은 비용을 무시할 수 없다. 실제적인 해결 방법은 수시로 초고속 대역폭을 가진 선로로 업데이트 하고, 서버의 수를 확장해 계속해서 분산화 하는 것뿐이다.

Ⅳ. 제안하는 분산 서버 구조

1. 분산 서버 구조

제안하는 분산 서버 구조는 비대칭 서버 구조와 병목현상도 줄일 수 있도록 Pear-to-Server 형태를 갖는다. 서버는 사용자인증 서버 및 DB 등의 주요 서버만 존재하며, 그 외의 게임 서버 등은 클라이언트 내에서 자체 해결하도록 설계 된다(그림 3). 본 구수의 클라이언트는 위치한 영역에 대해서 서버가 되며 주변 클라이언트들에게 메시지를 송, 수신하고 소켓관리 및 데이터베이스 관리까지 맡게 된다. 이런 방법을 적용함으로써 최대한 서버의 부하를 줄이고 클라이언트간의 연결망에서 메시지 처리의 해결을 볼 수 있다.

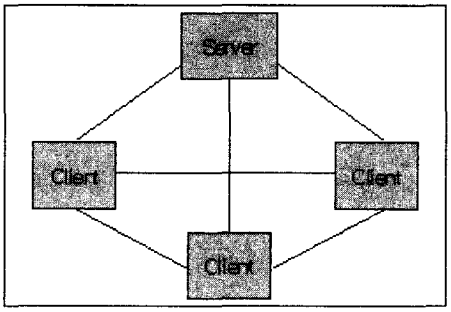

그림 3. 분산 서버 시스템 구조

2. 클라이언트 소켓 연결구조

제안하는 구조는 TCP/IP 기반 프로토콜에서 최적화되어 있으며, 메인 서버와의 연결포트를 기본으로, 자신이 위치한 영역을 중심으로 주변의 클라이언트들과의 연결을 수행한다. 주변 클라이언트들과의 연결을 수행하기 위해서는 주변 클라이언트의 IP가 필요한데 이는 서버가 제공해 주며, 같은 영역에 있는 하나의 클라이언트들에게만 전송해주게 된다. 이 IP테이블을 받은 클라이언트는 주변 클라이언트들에게 같은 IP테이블을 모두 전송해 준다.

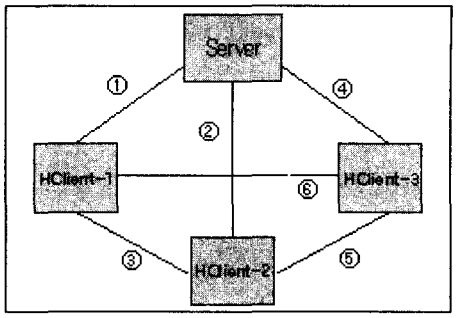

그림 4. 접속한 클라이언트들에 대한 연결순서

2.1 클라이언트 접속 처리

클라이언트가 메인 서버에 접속 할 때마다 메인 서버는 IP주소 테이블을 이용하여 클라이언트 스스로 연결망을 구성하도록 유도한다(그림 4).

- HClient-1의 접속

① 서버로부터 접속하여 인증절차를 마친 후, 자신의 정보를 획득한다.

- HClient-2의 접속

② 서버로부터 접속 후 인증절차를 마친 후, 자신의 정보를 획득한다. 그다음 서버는 HClient-1에게 새로 접속한 클라이언트 IP를 알려준다.

③ HClient-1은 새로 접속한 HClient-2에게 모든 IP주소를 전송한다. 그다음 HClient-2는 받은 IP주소 테이블에 있는 모든 IP와 연결을 한다.

- HClient-3의 접속

④ 서버로부터 접속, 인증절차를 마친후, 자신의 정보를 획득한다. 그 다음 서버는 HClient-2에게 새로 접속한 클라이언트 IP를 알려준다.

⑤ HClient-2는 새로 접속한 HClient-3에게 모든 IP주소를 전송한다. 그다음 HClient-3은 받은 IP주소 테이블에 있는 모든 IP와 연결을 한다.

⑥ HClient-2는 기존에 접속한 HClient-1에게 새로 접속한 IP주소를 전송한다. 그다음 HClient-1은 받은 IP주소의 클라이언트와 연결을 수행 한다.

2.2 클라이언트 연결해제 방법

클라이언트 해제의 경우는 매우 간단히 처리될 수 있으며, 만약 비정상적인 종료로 인해 종료 메시지가 발송되지 않는다면, 비정상적으로 종료된 클라이언트에게 메시지를 보내게 될 첫 번째 시스템이 이를 관련된 모든 클라이언트 및 서버에게 알려 주면 된다.

3. 전송 메시지

제안하는 분산 서버 구조는 네트워크 연결망이 서버간의 연결망과 클라이언트간의 연결망을 직접 가지고 있으므로 하나의 클라이언트는 자신과 관련된 서버나 클라이언트에게 빠른 속도로 의사전달을 할 수 있다. 제안하는 방법에서는 의사전달에도 등급별로 메시지가 분류되어 처리되도록 설계되어 있는데, 이것은 메시지의 중요도에 따라 효율적인 운영을 위한 것이다. 클라이언트가 전송하는 메시지는 중요도에 따라 크게 아래와 같은 3가지로 나뉘어 진다.

• 레벨 3

서버에 직접적으로 전송되는 중요한 메시지로서 여기에서는 서버와 클라이언트간의 동기화나, 시스템 업 데이트 및 오류나 기타 알림 등 여러 가지 시스템 정보를 받아들이는데 사용되며, 받은 즉시 즉각적으로 처리되는 메시지이다. 주로 캐릭터의 중요 정보 저장 등을 할 때 이루어진다.

• 레벨 2

서버에 전송되긴 하지만 일정한 시간을 두고 결과 메시지만 전송되는 메시지로서 IP 테이블을 참조해 주변 클라이언트들에게 알려주기는 하지만 서버에게는 일정한 시간이 지난 후에 전송하거나, 다른 이유로 전송할 필요가 있을 때 사용하는 메시지이다. 이런 메시지는 자주갱신이 필요할 경우에 해당하며 대표적으로 대부분의 행동 모션(움직임, 공격, 방어, 죽음, 기술사용 등등)이 이에 해당한다. 이 메시지는 클라이언트간의 전송은 실시간으로 이루어지나 서버와의 연결을 타이머를 이용한 일정 주기를 가지고 전송된다. 여기서 전송 주기는 접속자수의 증가에 비례한다.

• 레벨 1

서버에 전송되지 않으며 단지 클라이언트간의 정보전달에만 쓰이는 일회용 메시지로서 서버에게 전혀 알릴필요가 없는 클라이언트 간에만 필요한 메시지로써, 클라이언트간의 대화를 위한 메시지(전체대화, 그룹대화, 영역대화, 기밀대화 등)나 행동모션 중 일부 감정표현 메시지 등이 이에 해당하며, 클라이언트간의 연결망을 통하여 직접 전송된다. 거의 대부분의 네트워크 게임에서 가장 많은 메시지 패킷은 행동모션과 대화 메시지이다. 사실상 이런 메시지는 서버에 많은 부하를 줄뿐 중요한 정보는 아니다. 제안하는 본 방법에서는 클라이언트 간 에는 실시간으로 메시지가 전송되고 처리되나, 서버에게는 갱신이 반드시 필요할 때만 전송을 함으로써 통신 속도의 향상을 꾀하고 있다. 그림7은 레벨별로 전송이 되는 통신 구간을 나타낸 것이다. 서버와 가장 밀접한 관계를 갖는 메시지일수록 중요한 메시지이다.

4. 메시지 비동기화 처리

기존의 분산 네트워크 시스템에서는 영역을 담당하고 있는 서버들이 메시지 큐를 관리하고 운영하지만 제안하는 시스템에서는 클라이언트들도 메시지 큐 기능이 삽입된다. 메시지 비동기화 문제는 위에서 제시한 등급에 따라서 결정된다. 메시지 큐는 멀티구성방식을 따른다. 등급별로 메시지 큐가 존재하며, 메시지의 입력은 레벨이 가장 높은 등급의 큐에서 받는다. 메시지 레벨이 3이 안되는 메시지는 모두 해당 등급 메시지 큐로 옮긴다. 메시지 추출 순서는 레벨 3에서 레벨 2, 그리고 레벨 1의 순서이며, 레벨 3의 메시지가 모두 수행되면, 레벨 2, 레벨 1을 수행한다. 레벨 2와 레벨 1의 메시지는 번갈아가면서 메시지를 추출하게 된다.

V. 결 론

본 논문에서는 복합 분산 서버 시스템 구조를 제안하였다. 기존 분산 네트워크 시스템과의 차이는 서버의 의존도가 매우 작아진다는 점이다. 이는 네트워크 서버 시스템의 운영 및 구입비용 절하를 가져오며, 서버를 거치지 않고 클라이언트간의 직접연결로 인해 빠른 통신 속도를 보장할 수 있는 장점을 갖는다. 시간적 동기화 문제는 클라이언트간의 수시 테스트를 위해 빠른 속도로 보정을 하게 되며, 메시지 비동기화 처리 문제는 메시지의 레벨화 및 멀티 메시지 큐를 운영, 레벨별 전송 통로 결정에 의해 비동기화 채택으로 발생하는 다른 문제 들을 획기적으로 줄일 수 있다. 한편, 제안하는 구조에서는 클라이언트 자체 내에 소켓 관리, 메시지 관리 등 서버가 하는 일들을 맡아서 하게 되므로 클라이언트의 부하량의 증가가 초래된다는 점이다. 향후 연구과제로서 제안하는 복합 분산 서버 시스템의 구조에 대한 구체적인 설계를 바탕으로 한 구현을 통한 검증이 요구된다.

참 고 문 헌

[1] 고욱, 첨단 게임 기술 동향, 정보과학회지, 제15

권 8호, 1997년 8월.
[2] 김형도, 문지영, 멀티미디어 네트워크 게임을 위한 그룹통신 플랫폼, 정보과학회논문지(A), 제24권 제2호, 1997. 2.
[3] 바람의 나라, http://www3.nexon.co.kr/baram/.
[4] 어둠의 전설, http://www3.nexon.co.kr/lod/.
[5] 영웅문, http://www.netgame.com/hero/.
[6] 아크메이지, http://www.magewar.com/archmage/.
[7] Ultima Online, http://www.owo.com/.
[8] TEN, http://www.ten.com/.
[9] MPlayer, http://www.mplayer.chan-neli.net/, http://www.mplayer.com/.
[10] Battle.net, http://www.battle.net/.
[11] Internet Gaming Zone, http://igz.microsoft.com/.
[12] Bishop, L., et al., Designing A PC Game Engine, IEEE Computer Graphics and Applications, Vol 18, Nol, 1997
[13] Park, J. and Park, C., Development of a Multiser & Multimedia Game Engine Based on TCP/IP, Proceedings of the 1999, IEEE Pacific Rim Computers and Singal Processing, August, 1997.
[14] 김미라, 박긴, 곽미라, 조동섭, "APEX 기반 분산 네트워크 게임 설계" 한국게임학회 동계 학술발표대회 논문집, pp.361-363, 2002.
[15] 최우영, 정정민, 김도원, 최성, "네트워크 게임의 다중 시스템 운영체계에 관한 연구", 한국게임학회 동계 학술발표대회 논문집, pp401-406, 2002

배 재 환(Jae-Hwan Bae) 정회원

1992년 2월 : 경일대학교 산업시스템공학과 졸업
1997년 2월 : 대구대학교 산업정보대학원 졸업
2003년~현재 대구대학교 대학원 정보통신공학과 박사과정 수료
2000년 3월~ 현재 : 탐라대학교 정보출판미디어학부 컴퓨터게임학전공 전임강사

<주관심분야> 네트워크게임, 멀티미디어콘텐츠, 분산시스템, 병렬처리