

모바일 온라인 게임을 위한 Proxy 구조의 설계

¹⁾ 서원일 이은정
경기대학교 일반대학원 게임웨어전공
manofone@kyonggi.ac.kr

Design of Proxy Architecture for Mobile Online Game

W.Seo, E.Lee
Dept. of Computer Science, Kyonggi Univ.

요 약

최근 모바일 단말기의 보급이 증가함에 따라 이를 이용한 다양한 어플리케이션이 개발되고 있다. 특히 게임 분야는 기존의 Stand-Alone 방식에서 모바일 단말기의 무선통신 기능을 이용한 네트워크 온라인 게임으로 전환이 이루어지고 있다. 하지만, 네트워크 온라인 게임은 게임 로직의 처리, 렌더링 처리 같은 복잡한 계산과 비동기적인 네트워크 송·수신이 빈번하게 이루어지기 때문에 모바일 단말기에서 모두 처리하기에는 많은 문제점이 있다. 따라서, 본 논문에서는 모바일 단말기가 처리해야 할 계산 및 통신기능을 대신하여 처리할 수 있는 프록시 구조를 제안한다.

1. 서 론

최근 모바일 단말기의 보급이 증가함에 따라 이를 이용한 여러 가지 어플리케이션이 개발되고 있다. 그 중 게임 분야에서는 기존의 Stand-Alone 방식에서 모바일 단말기의 무선 통신 기능을 활용한 간단한 네트워크 게임으로의 전환이 이루어지고 있다[1][2].

하지만, 핸드폰과 스마트 폰 그리고 PDA와 같은 모바일 단말기는 컴퓨팅 능력의 제한, 작은 메모리 공간, 디스플레이 크기의 제약 등의 단점을 가지고 있다. 무선 통신 역시 안정적으로 연결이 지속되기 힘들다는 단점을 가지고 있다. 이러한 하드웨어의 제약은 모바일 단말기를 이용한 네트워크 온라인 게임 개발에 걸림돌이 되고 있다.

본 연구에서는 이동 단말에서의 무선 통신 네트워크 온라인 게임지원을 위한 프록시 구조를 제안한다. 일반적으로 프록시는 서버와 클라이언트 사이에서 중간 게이트웨이 역할을 담당하는 소프트웨어 시스템이다[3]. 이러한 프록시는 현재 Web-Access나 Multimedia Content 분산에서 성공적으로 사용되고 있다. 이러한 프록시 구조를 응용하여 모바일 단말기를 이용한 네트워크 온라인 게임을 위한 프록시 구조를 개발하는 것이 이 논문의 목표이다.

본 논문에서 제안한 프록시 구조는 전체적인 게임을 진행하는 중앙 서버와 모바일 단말 사이에 프록시가 위치한다. 프록시는 중앙 서버로부터 전송 받은 데이터를 처리하여 모바일 클라이언트로 전송함으로써 모바일 클라이언트가 처리해야 할 컴퓨팅 기능 및 서버와의 통신 기능을 대신하게 된다. 또, 모바일 클라이언트로부터 수신 받은 데이터를 처리하여 중앙서버로 송신하는 역할을 담당하게 된다.

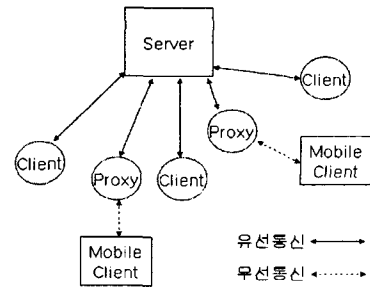


그림 1 프록시 시스템 구조도

본 논문의 구성은 다음과 같다. 제2절에서는 프록시 시스템의 역할에 대하여 설명하고, 제3절에서는 프록시를 적용하여 구현한 시스템에 대해 설명하고, 마지막으로 제4절에서 결론을 맺는다.

2. 프록시 시스템의 역할

2.1 모바일 클라이언트의 특징

일반적으로 모바일 단말기는 다음과 같은 하드웨어의 제약을 가지고 있다.

- ① 컴퓨팅 능력의 제약
- ② 무선통신의 제약 - 낮은 대역폭, 불안정한 연결
- ③ 메모리 및 해상도의 제약

위와 같은 모바일 클라이언트가 가지는 단점을 극복하기 위해 본 논문에서는 모바일 온라인 게임을 위한 프록시 시스템을 제안한다.

2.2 컴퓨팅 기능의 대리

일반적인 온라인 게임에서 클라이언트가 계산을 통해 처리해야 하는 것들은 다음과 같다.

- ① 게임 로직의 계산
- ② 송·수신 이벤트의 처리
- ③ 화면 출력을 위한 렌더링 계산

모바일 클라이언트의 컴퓨팅 능력으로는 온라인 게임 진행을 위한 모든 데이터를 처리하기에는 부족한 면이 있다. 그러므로 프록시에서 게임 진행에 따른 상태변화의 계산은 물론이고, 송·수신 이벤트의 처리와 화면 렌더링 계산까지 수행하는 방법을 생각해 볼 수 있다.

2.3 통신 흐름의 제어

무선 통신은 장소에 구애를 받지 않는다는 장점이 있지만, 낮은 대역폭과 연결 상태가 고르지 못하다는 단점이 있다. 또, 네트워크 온라인 게임은 데이터의 송수신이 비동기적으로 이루어진다. 수신이 고르지 못한 비동기적 데이터를 수신할 경우 모바일 클라이언트의 능력으로 모두 처리하기 어렵다. 따라서, 유선 통신으로 연결된 프록시에서 비동기적인 데이터를 수신하여 최종 상태값으로 변환하고, 처리된 결과를 동기화된 소량의 통신 이벤트로 모바일 클라이언트로 전송함으로써 무선 통신의 단점을 보완할 수 있다.

2.4 멀티미디어 데이터의 처리

모바일 클라이언트는 별도의 하드 디스크 없이 작은 메모리만 가지고 있다. 반면에 게임 데이터의 대부분을 차지하는 멀티미디어 데이터는 용량이 커서 모바일 클라이언트에 모두 저장하기는 어렵다. 그래픽 이미지의 경우 프록시에서 모바일 클라이언트의 해상도에 맞추도록 변환하여 모바일 클라이언트에 전송할 수 있다. 또, 전체 멀티미디어 데이터를 유선 클라이언트처럼 모두 한꺼번에 설치하는 것이 아니라 몇 개로 쪼개어 필요할 때마다 전송하는 방법으로 메모리 제약을 보완할 수 있을 것이다.

2.5 모바일 클라이언트의 컨트롤 기능

본 논문에서 제안하는 프록시 시스템에서 모바일 클라이언트는 사용자의 입력에 따라 게임 컨트롤과 뷰 컨트롤 기능만을 수행한다.

게임 컨트롤은 사용자가 게임을 진행하기 위해 캐릭터에게 내린 이동, 공격등의 명령을 프록시로 송신하여 게임을 진행하는 기능을 말한다.

뷰 컨트롤은 모바일 클라이언트가 가지는 디스플레이의 제약을 극복하기 위한 기능으로써, 예를 들면 Zoom In/Out, 시점이동과 같은 기능들이다. 이를 지원하기 위해 모바일 클라이언트는 프록시에 가시 영역 및 모드를 등록할 수 있다. 사용자가 모바일 클라이언트 가시영역의 크기 및 위치를 조절하거나 화면 모드를 변경하면, 모바일 클라이언트는 이를 프록시에 등록한다. 프록시는 모바일 클라이언트의 변경된 내용에 맞게 렌더링하여 모바일 클라이언트로 전송을 한다.

본 논문에서 제안하는 프록시 구조에 적용되는 모바일 클라이언트는 터미널과 같이 동작한다. 하지만, 모바일 클라이언트의 사양에 따라 더 많은 기능을 모바일 클라이언트에서 처리할 수도 있을 것이다.

3. 프록시 시스템의 구현

본 연구에서 제안한 프록시를 적용하기 위해 Peer-To-Peer 통신 구조의 복제 객체 기반 데이터 및 제어를 수행하는 분산 가상 환경 플랫폼으로 KGDVE를 개발하였다. 복제 객체에 기반한 Peer-To-Peer 구조는 중앙 집중 클라이언트/서버 구조에 비해 통신 부하를 줄일 수 있고, 물리적인 통신의 제약에 적게 받으며, 호스트의 독립성과 시스템 전체의 안정성을 높일 수 있다[7].

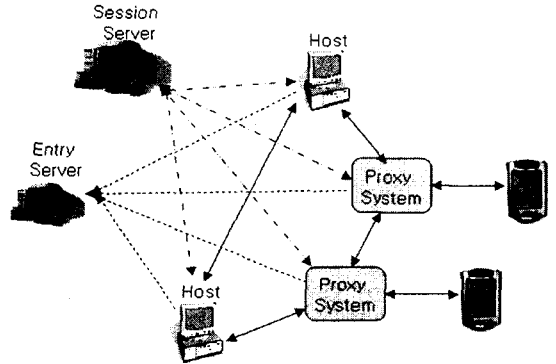


그림 2 전체 시스템의 구조도

KGDVE에서는 연결관리를 담당하는 세션 서버가 있고, 사용자가 시스템에 접속하는 클라이언트는 호스트라 불린다. 이 호스트에서 해당 사용자에 대한 진짜 객체를 생성하고 다른 호스트들은 대응하는 복제 객체를 가진다. 사용자의 액션은 게임 객체를 통해 처리되고 계산되어 복제 객체들에게 전달된다. 또한 다른 호스트에서 발생한 사건은 이 호스트의 복제 객체들에게 전달된다.

호스트 또는 프록시는 게임에서 각 플레이어가 접속하는 시스템이 된다. 시스템은 게임 객체를 가지고 있고, 시스템은 이를 통해 데이터 및 제어 처리를 한다. 한편 시스템은 다른 호스트들의 게임 객체에 대응하는 복제자를 가지고 있다. 각각의 게임 객체는 자신의 상태에 변화가 일어나면 자신의 복제자에게 상태 변경 이벤트를 전달한다. 복제자들은 디스플레이 기능과 통신 기능을 가지고 있다. 그리고, 이벤트는 게임 객체의 단순한 상태 값 변화일 수도 있고 전달 받은 호스트의 게임 객체에게 영향을 미칠 액션일 수도 있다.

이러한 복제자 기반의 분산 시스템에서는 새로운 호스트가 참여할 때 기존의 다른 호스트들과 초기화 과정을 거쳐야 한다. 이를 위하여 본 시스템에서는 세션 서버 모듈을 두어 새로운 호스트가 참여를 요청하면 세션 서버가 다른 모든 호스트들에게 이 새로운 호스트의 참여를 알려준다. 그러면 그 호스트들이 새로운 참여 호스트에게 초기화를 위한 이벤트를 보내게 된다.

모바일 클라이언트와 프록시간의 통신은 다음과 같이 이루어진다. 모바일 클라이언트가 데몬에 접속하면 데몬은 프록시를 실행시키고, 세션 서버의 리스트를 모바일 클라이언트에게 전송한다. 모바일 클라이언트에서 접속할 세션 서버를 선택하면 프록시는 일반 호스트처럼 세션 서버에 접속하고, 게임 시작을 위한 준비를 마친다. 게임이

시작되면 모바일 클라이언트는 프록시에게 사용자 입력에 따른 이벤트를 전송하고, 프록시는 모바일 클라이언트에 게 1/10초 간격으로 동기화된 데이터를 전송하게 된다.

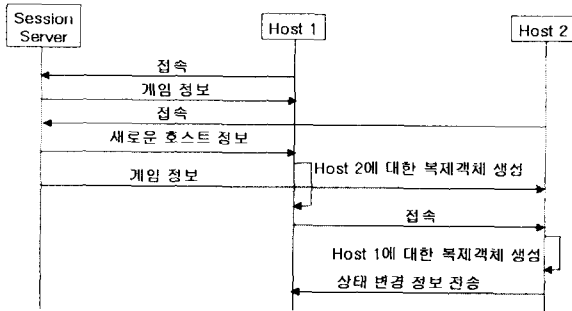


그림 3 호스트의 접속 흐름도

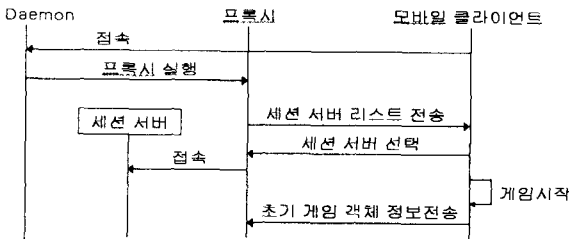


그림 4 프록시를 이용한 모바일 클라이언트의 접속 흐름도

본 연구에서는 통신 이벤트의 양을 줄이기 위하여 추측 방법(Dead Reckoning)을 이용하고 있다[6]. 서버와의 통신 지연이 발생할 경우 프록시는 각각 호스트의 현재 상태 값에 따라 예측되는 호스트의 상태를 계산하여 유지한다. 본 연구에서는 [7]에서 소개된 추측방법을 구현하였다.

또한 본 시스템에서는 게임 환경에서의 대상 영역이 실제 모바일 클라이언트에서 디스플레이 되는 영역에 비해 훨씬 클 경우 관심 영역(Area Of Interest)[5,8]를 도입하였다. 관심영역을 사용함으로써 프록시와 호스트들 사이의 통신량을 줄일 수 있었다.

본 연구에서는 KGDVE가 제공하는 분산 가상 환경의 플랫폼을 기반으로 멀티플레이 네트워크 게임 “과일먹기”를 구현하였다. 아래의 화면은 이 게임의 실행 화면을 보여준다.

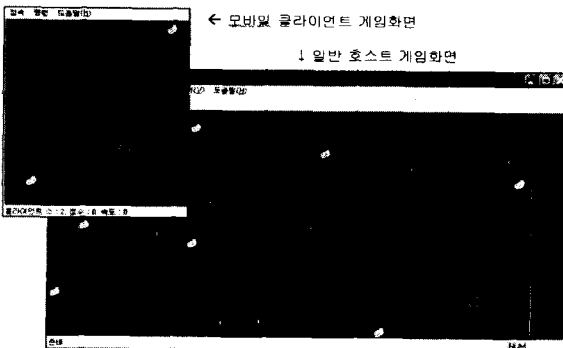


그림 5 게임 화면

본 연구에서 개발한 프록시 시스템의 테스트 플랫폼으로써 서버와 프록시, 클라이언트 모두 Microsoft Window XP상에서 Visual C++ 6을 이용하여 개발하였다.

4. 결론 및 향후 계획

모바일 단말에서의 네트워크 온라인 게임은 Stand-Alone게임에 비해 복잡한 게임 로직의 처리 및 렌더링 처리뿐 아니라, 네트워크 송·수신에 대한 처리도 고려해야 하기 때문에 모바일 네트워크 온라인 게임의 개발은 어려움이 있다. 본 논문에서는 이러한 어려움을 극복하기 위해 모바일 네트워크 온라인 게임에서의 프록시 구조를 제안하였다. 제안된 프록시 시스템은 모바일 클라이언트가 처리해야 하는 대부분의 계산 및 통신을 처리함으로써 모바일 클라이언트의 계산량을 효과적으로 줄일 수 있었고, 이를 통하여 모바일 단말에서의 네트워크 온라인 게임이 가능하게 하였다.

향후 연구 방향은 제안한 시스템을 실제 무선 통신을 사용하는 클라이언트에서 구현하고, 일반 클라이언트와의 동시 플레이를 테스트할 예정이다. 그리고, 시스템의 성능 평가를 통해 프록시 구조를 이용한 모바일 클라이언트의 성능을 검증할 예정이다.

참고문헌

- [1] <http://www.mjoynet.com> 엠조이넷 홈페이지
- [2] <http://www.dt.co.kr> 디지털 타임즈 홈페이지
- [3] 컴퓨터 용어 대사전 편찬위원회 저, 컴퓨터 용어 대사전, 정보문화사
- [4] 최미진, 조은영, 이동만, 강경란, “유무선 통신 환경에서 그룹 에디팅을 위한 사용자 관심도 기반 필터링 기법”, 정보과학회 논문지 : 인간과 컴퓨터 상호작용 제 29권 pp. 169-171, 2002
- [5] 김상욱, 이태종, 김성조, “대규모 분산 가상환경 상에서 관심영역의 부하 적응적 관리”, 정보과학회 논문지 : 시스템 및 이론 제 28권 7호, pp.317-328, 2001.
- [6] J.Aronson, “Dead rockening:latency hiding for networked games,” in Gamasutra magazine, September 19, 1997.
- [7] J.Smed, T.Kaukoranta, H.Hakonen, “Aspects of Networking in Multiplayer Computer Games”, *The Electronic Library*, 20(2):87-97, 2002.
- [8] K.L.Morse, An adaptive, distributed algorithm for interest management, Ph.D. thesis, Univ. of California, Irvine, 2000.
- [9] N.Baughman, B.Levine, “Cheat-proof payout for centralized and distributed online games,” IEEE InfoCom 2001.
- [10] M. Mauve, S.Fisher, J.Widmer, “A genetic proxy system for networked computer games,” First Workshop on Network and System Support for Games, Braunschweig, Germany, April 16-17, 2002