The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 16, No. 4, pp.73-80, Aug. 31, 2016. pISSN 2289-0238, eISSN 2289-0246

http://dx.doi.org/10.7236/JIIBC.2016.16.4.73
JIIBC 2016-4-12

멀티플레이어 온라인 게임을 위한 하이브리드 구조의 복제와 일관성 제어 기법

Replication and Consistency Control in Hybrid Architectures for Multiplayer Online Games

김진환*

Jin-Hwan Kim*

요 약 인터넷을 이용하는 멀티플레이어 온라인 게임(MOG)은 전형적으로 클라이언트-서버 또는 peer-to-peer 구조를 기반으로 구성된다. 본 논문에서는 두 가지 구조의 장점을 활용하기 위하여 클라이언트-서버 구조와 peer-to-peer 구조를 결합하는 방법을 제시한다. 대부분의 멀티플레이어 온라인 게임들은 객체마다 강력한 일관성 제어를 제공하고자 기본 사본을 갖는 복제 기법을 사용한다. 이 기법에서 각 객체와 캐릭터는 기본 사본이라고 하는 절대적 사본과 2차 사본이라고 하는 복제본들로 구성된다. 객체에 대한 갱신은 기본 사본에서 먼저 수행되어야만 한다. 제시된 하이브리드 구조에서는 기본 사본들이 서버 또는 클라이언트에 존재할 수 있다. 이러한 구조에서는 서버가 유지하는 객체의 수를 감소시킴으로써 서버와 클라이언트들 간의 부하 조정이 가능하다. 게임은 일관성 요건이 상이한 다양한 액션의 유형들로 구성된다. 게임 일관성에 대해서는 성능과 일관성 간에 적절한 상충 관계를 제공할 수 있는 여러 수준의 기법이 합리적이다. 본 논문에서 기본 사본 모델을 갖는 하이브리드 게임 구조에 대한 성능 분석 결과가 기술된다.

Abstract Multiplayer Online Games(MOG) using the Internet are typically organized based on a CS(client-server) or P2P(peer-to-peer) architecture. We then propose a method that combines a P2P architecture with a CS architecture in order to utilize their advantages. Most MOGs use a primary-copy replication approach that provides strong consistency control over an object. For each object and character there exists an authoritative copy, called primary copy and all other copies are secondary copies or replicas. Any update to the object has to be first performed on the primary copy. In the proposed hybrid architecture, primary copies may reside on the server or be held by clients. In this architecture, load balancing between a server and clients can be achieved by reducing the number of objects maintained by the server. Games consist of various types of actions with different consistency requirements. A multi-level approach to game consistency is sensible as it provides the best trade-off between consistency and performance. The performance for the hybrid game architecture with the primary-copy model is evaluated through simulation experiments and analysis in this paper.

Key Words : hybrid architecture, consistency control, replication, primary-copy, performance

*정회원, 한성대학교 멀티미디어공학과
접수일자 2016년 6월 20일, 수정완료 2016년 7월 20일
게재확정일자 : 2016년 8월 5일

Received: 20 June, 2016 / Revised: 20 July, 2016 /
Accepted: 5 August, 2016
*Corresponding Author: kimjh@hansung.ac.kr
Dept. of Multimedia Engineering, Hansung University, Korea

Ⅰ. 서 론

멀티플레이어 온라인 게임(MOG; Multiplayer Online Game)은 게임 상태가 서버와 많은 클라이언트들 간에 부분적으로 복제되는 대규모 분산 시스템으로 간주된다[1]. MOG는 플레이어들 간 또는 플레이어와 환경 간에 상호 작용을 위하여 플레이어가 자신의 아바타를 제어하는 대규모의 가상 세계를 중심으로 구성된다. MOG의 전형적인 구조는 그래픽 유저 인터페이스를 통하여 게임 세계의 특정 캐릭터나 아바타를 제어하는 플레이어들 즉 클라이언트들로 구성된다. 게임 세계는 플레이어 또는 인공 지능으로 제어되는 캐릭터, 환경을 형성하는 비 상호작용적 객체, 캐릭터와 상호 작용할 수 있는 변경가능한 아이템 등 다양한 유형의 객체로 구성된다. 플레이어들은 MOG의 기본적 상호작용 방법인 액션을 사용하여 게임 세계에서 이동하거나 아이템을 습득하거나 다른 플레이어와 아이템을 교환할 수 있다. 게임 세계는 모든 플레이어들에게 동일하게 적용되어야 하며 한 플레이어의 액션은 다른 모든 플레이어들이 관찰할 수 있어야만 한다.

MOG들은 엄청난 통신량과 처리 부하를 유발할 수 있어서 동시에 수천명 이상의 플레이어들을 수용할 수 있는 규모조정성, 일관성, 보안, 신속한 응답시간 제공 등이 어려운 문제로 인식되고 있다[2, 3]. 게임 실행과 게임 상태의 배포가 서버에 의해 전적으로 제어되는 클라이언트-서버(Client-Server) 시스템(이하 CS 시스템이라 함)은 가장 보편적인 게임 구조이다. 그러나 CS 구조는 규모조정성이 약하고 서버에 장애가 발생할 경우 게임이 중단되는 상태가 발생할 수 있다. P2P(peer-to-peer) 구조는 통신량과 작업 부하를 플레이어들에게 분산시킬 수 있기 때문에 잠재적으로 규모조정성의 증가, 비용 절감, 성능 향상 등이 가능하여 MOG 구조로 연구되고 있으나 P2P 구조는 아직 보안이나 게임 관리에 관한 약점이 문제시되고 있다[4]. 본 논문에서는 CS 구조와 P2P 구조를 결합하여 두 구조의 장점을 최대한 활용할 수 있는 하이브리드 구조를 제시한다.

공통적으로 게임 엔진은 미들웨어로써 구현되며 플레이어들의 요청 작업을 수신하고 게임 세계를 관리하며 갱신 결과를 다시 플레이어들에게 전송하는 역할을 수행한다. 대부분의 게임 엔진들은 기본 사본 복제(primary-copy replication) 방법을 사용한다[5]. 각 객체와 캐릭터를 책임지는 사본을 기본 사본 또는 마스터 사본이라 하며 이외의 다른 사본들은 2차 사본 또는 복제본(replica)이라 한다. 각 플레이어는 관련있는 게임 객체의 사본들을 자신의 컴퓨터에 저장하게 되며 객체에 대한 임의의 갱신 과정은 기본 사본에서 먼저 수행되어야 한다. 본 논문에서 제시된 하이브리드 구조에서 기본 사본은 서버 또는 플레이어들이 유지하게 되며 특정 플레이어가 복제본 객체에 대하여 갱신 과정을 수행할 경우 객체의 기본 사본으로 갱신 과정이 전송되며 해당 기본 사본을 소유한 서버 또는 플레이어가 갱신 과정의 허용 여부를 결정하게 된다. 모든 기본 사본을 서버가 보유하는 것이 아니라 서버와 플레이어들이 적절히 구분하여 보유함으로써 복제본 수와 갱신 비율이 증가할 경우 서버의 작업 부하를 조정할 수 있게 된다.

MOG같은 분산 구조에서 수많은 플레이어들이 게임에 참여할 경우 모든 복제본에 대한 갱신 과정이 정확히 동시에 수행되긴 사실상 어렵다[6, 7]. 본 논문에서는 하이브리드 구조의 멀티플레이어 게임에서 수행되는 액션의 중요도에 따라 적절한 일관성이 유지되는 기법을 사용한다. 객체의 기본 사본을 가진 서버나 플레이어가 있는 사이트에서 요청 작업의 순서를 정확히 유지함으로써 일관성을 보장해야 한다. 게임에서 일관성을 낮은 수준으로 유지할 수 있는 액션과 높은 수준으로 유지해야 하는 액션을 수행할 경우 통신량과 처리 시간이 달라진다. 일관성 수준을 높일수록 시스템의 성능은 저하될 수 있으므로 일관성과 성능의 상충 관계를 고려하여 적절한 일관성을 선택할 필요가 있다.

본 논문에서는 MOG를 위한 하이브리드 구조의 기본 사본 모델이 2 장에서 제시되며 3 장에서 여러 수준으로 유지되는 게임 액션과 일관성 기법이 기술된다. 4 장에서는 기본 사본을 가진 하이브리드 구조에서 MOG의 수준별 일관성 제어 기법과 통신 대역폭이 분석되었으며 5 장에서 결론이 기술된다.

Ⅱ. 기본 사본을 갖는 하이브리드 구조

1. 기본 사본 기법

각 플레이어는 게임 소프트웨어를 이용하여 게임 세계를 렌더링하고 자신의 아바타를 제어하며 필요한 액션을 다른 플레이어에게 전송할 수 있다. 다른 플레이어는 수신한 요청 작업들을 순서적으로 처리하고 게임 상태를

조정하며 필요할 경우 서버나 관련된 플레이어들에게 변경된 내용을 통보해야 한다. 이 과정은 기본 사본 복제 기법[5]으로 제공될 수 있으며 대부분의 게임 엔진들이 이 기법을 사용하고 있다. 실제 게임 세계는 다양한 객체들로 구성되며 각 객체는 여러 속성을 가질 수 있다. 각 객체의 갱신을 책임지는 사본을 기본 사본(primary copy)이라고 하며 나머지 사본들을 2차 사본(secondary copy) 또는 복제본(replica)이라고 정의한다. 기본 사본 모델에서 모든 복제본은 읽기 전용이며 플레이어가 임의로 변경할 수 없다.

각 플레이어는 다른 플레이어들과 연관된 게임 객체의 복제본을 가지게 되며 임의의 객체에 대한 갱신은 기본 사본에서 먼저 수행되어야 한다. 기본 사본과 복제본의 분산 방법은 게임 구조에 따라 달라질 수 있다. 만일 플레이어가 복제본 객체에 대하여 갱신을 수행하고자 할 경우 기본 사본에 대하여 먼저 갱신 요청을 해야 한다. 따라서 기본 사본을 보유한 서버나 플레이어가 갱신 요청의 수락 여부를 결정하며 객체를 갱신했을 경우 해당 객체의 복제본을 가진 모든 플레이어들에게 갱신 결과를 전송함으로써 객체에 대한 갱신 과정이 종료되는 것이다.

2. CS 구조

MOG를 위한 가장 일반적인 CS 구조[4]에서 모든 객체의 기본 사본이 서버에 존재하게 되고 클라이언트들은 복제본만을 유지하게 된다. 서버는 모든 게임 객체들의 기본 사본을 유지하고 이들에 대한 모든 갱신 요청을 수신하며 갱신 결과를 멀티캐스트 기법[8]으로 다른 플레이어들에게 전송하게 된다. 이러한 시스템에서는 서버에 많은 계산 부하가 걸릴 수 있고 통신 대역폭도 매우 커지게 되는 단점이 존재하게 된다.

그러나 CS 구조는 구현이 비교적 단순하며 집중화된 서버를 통하여 상태 일관성을 유지하기가 수월하고 플레이어의 속임수 행위를 방지할 수 있다. 또한 게임 제공업체는 영리를 추구할 수 있는 플레이어 가입 구조를 생성하고 제어할 수 있으며 지속적인 게임 감시도 가능하다. 이러한 이유 때문에 CS 구조가 멀티플레이어 게임에 많이 활용되고 있다.

3. P2P 구조

P2P 구조에서는 모든 노드가 서버와 클라이언트 역할을 동시에 수행할 수 있다. 기본 사본을 갱신할 경우 해당 플레이어(peer를 의미함)는 관련된 다른 모든 플레이어들에게 갱신된 내용을 전송하게 된다[9]. 즉 P2P 게임 모델에서는 각 플레이어가 자신의 게임 시뮬레이션을 직접 실행할 책임을 가지고 있다[5]. 그러나 플레이어들 간에 비일관성을 탐지하고 이를 해결하는 서버가 없기 때문에 임의의 비일관성은 분산 협정 프로토콜을 사용하는 플레이어들에 의해 직접 탐지되어야만 한다. 집중화된 서버의 부재는 P2P 구조에서 클라이언트 간 메시지 지연 시간의 감소[6], 서버의 결함 발생 시에도 시스템 계속 유지 등 여러 가지 이점을 제공한다. 가장 중요한 것은 P2P 구조의 경우 서버의 병목 현상이 없다는 것이다.

그러나 P2P 구조는 게임 산업에서 CS 구조처럼 보편적으로 사용되는 상황은 아직 아니다. 중요한 이유는 P2P 구조가 CS 구조보다 구현하기 어려우며 게임 제어가 용이하지 않기 때문이다.

4. 하이브리드 구조

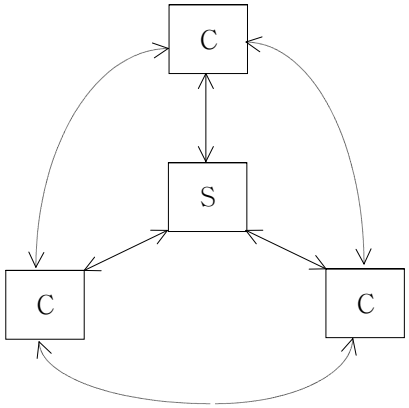

그림 1. CS와 P2P의 하이브리드 구조
Fig. 1. Hybrid of CS and P2P architecture

본 논문에서는 객체의 기본 사본이 서버(S)와 플레이어들(C)에게 적절하게 배분되는 하이브리드 구조를 제시한다(그림 1 참조). 만일 플레이어가 복제본 객체에 대하여 갱신을 수행하고자 할 경우 기본 사본을 보유한 서버 또는 해당 플레이어가 갱신 요청의 수락 여부를 결정하며 객체를 갱신했을 경우 해당 객체의 복제본을 가진 모든 플레이어들에게 갱신 결과를 전송함으로써 객체에 대한 갱신 과정이 종료되는 것이다. 이 구조에서는 객체의 기본 사본을 서버와 플레이어들이 적절히 구분하여 보유

함으로써 통신 대역폭 등 서버의 병목 현상을 완화시킬 수 있다. 또한 MOG 시스템에서 서버는 사용자 로그인, 지불 결재 정보 등 민감한 데이터 관리도 할 수 있고 매우 강한 일관성이 요구되는 액션도 수용할 수 있는 장점이 있다.

Ⅲ. 게임 액션과 수준별 일관성 제어

1. 플레이어 액션

게임 세계에서 객체의 유형은 플레이어에 의하여 제어되는 아바타, 음식이나 무기 등 변경가능한 객체, 플레이어가 아닌 인공지능에 의하여 제어되는 NPC(Non Player Character) 들로 구성된다. 플레이어들은 MOG의 기본적 상호작용 방법인 액션을 사용하여 객체의 이동, 객체의 획득이나 포기, 다른 플레이어와의 객체 교환 등을 수행하게 된다. 본 논문에서 게임 세계의 환경을 구성하는 지형, 배경 등 변경불가능한 객체에 대해서는 플레이어의 액션이 수행되지 않는 것으로 가정한다.

게임에서 플레이어의 상호작용은 플레이어의 갱신, 플레이어와 객체 간의 상호작용, 플레이어들 간의 상호작용 등 세가지 영역으로 구분될 수 있다[8]. 플레이어의 갱신은 플레이어 자신에게만 영향을 주는 게임 세계와의 상호 작용이며 위치 변경이나 플레이어의 아바타에 대한 그래픽 변경 등이 이에 해당된다. 최적화되지 않은 단순한 게임에서 대부분의 플레이어 상호작용은 위치 변경 과정에 해당된다. 플레이어와 객체 간의 상호작용은 플레이어와 변경가능한 객체 간의 상호작용을 의미하며 도구를 집어서 창고에 넣거나 음식을 섭취하는 행위들이 이에 해당될 수 있다. 플레이어들 간의 상호작용은 한 플레이어와 다른 플레이어들 간의 상호작용을 의미한다. 예를 들면 다른 플레이어를 공격함으로써 공격받은 플레이어는 부상당하고 공격한 플레이어는 일정 점수가 증가하는 등의 행위가 이에 해당된다. NPC와 플레이어의 상호 작용은 게임 설계에 따라 플레이어와 객체 간 또는 플레이어들 간의 상호작용으로 간주될 수 있다.

2. 비일관성의 발생

모든 갱신 과정이 기본 사본에서 먼저 수행되는 기본 사본 모델일 경우 복제본들이 존재하는 노드에서는 갱신 과정 간의 충돌 현상이 발생하지 않기 때문에 일관성 관리가 단순화된다. 대신 모든 갱신 과정은 기본 사본에 대하여 순서적으로 처리되어야 한다. 예를 들어 두 플레이어가 제 3의 플레이어를 거의 같은 시점에서 공격할 경우 모든 플레이어들은 동일한 순서로 실행된 갱신과정을 파악할 수 있어야 하며 먼저 공격한 결과가 다른 모든 복제본에 성공적으로 적용되어야 한다.

일반적으로 모든 복제본들에 대하여 동일한 순서로 갱신과정이 실행될 경우 모든 노드들은 동일한 상태를 가지게 된다. 일관성을 유지하기 위한 메카니즘이 반드시 필요하나 메시지들의 전송 과정에서 순서가 변경되거나 메시지가 손실될 경우 비일관성이 발생할 수 있다. 대부분의 게임들은 속도가 빠르지만 메시지 손실이 발생할 수 있는 비신뢰적인 UDP 메시지 프로토콜을 사용하고 있기 때문이다. 특별히 여러 개의 객체 상태를 변경하고 원자성(atomicity)이 요구될 경우에는 신뢰성이 있는 TCP 또는 임계 영역을 위한 2 단계 commit 프로토콜[9]을 사용하여 갱신 과정을 전송할 수도 있다.

반면 기본 사본에서 갱신 과정이 종료된 이후에 복제본들의 갱신 과정이 수행되기 때문에 이 시간동안 복제본들은 오래된 데이터를 보유해야하는 결과가 발생한다. 플레이어들은 기본 사본이 갱신된 결과를 인지하지 못하고 의미없는 갱신 요청을 시도할 수도 있다. 즉 플레이어들은 자신이 보유한 오래된 데이터 복제본에 기반한 액션으로 예상하지 못한 결과를 접할 수도 있게 된다.

3. 여러 수준의 일관성 제어

플레이어의 모든 상호작용을 격리성(isolation)과 원자성을 보장하는 트랜잭션으로 처리할 경우 가장 강력한 일관성 형태를 가질 수 있다[9]. 그러나 트랜잭션으로 처리할 경우 비용이 증가하고 모든 게임 액션에 적용시키기는 현실적으로 어렵다. 또한 원자성을 보장하기 위하여 2 단계 commit 프로토콜을 수행할 경우 지연시간이 증가하여 게임의 상호작용성을 현저하게 감소시킬 수도 있다. 따라서 현실적인 대안으로 궁극적(eventual) 일관성이 필요하다. 궁극적 일관성은 일관성이 다소 약화된 형태이며 임의의 객체 사본이 일시적으로는 비일관적일 수 있으나 갱신 행위가 오랜 시간 동안 중지될 경우 모든 객체 사본들이 궁극적으로 동일한 상태를 유지할 수 있음을 의미한다. 대부분의 게임에서 일관성보다는 상호작용성의 의미를 더욱 강조하기 때문에 비일관성을 예방하는 대신 비일관성을 해결하는 기법을 제공한다.

다양한 메카니즘에 의해서 구현되는 일관성의 여러 수준은 상이한 객체와 동일한 중요도로 구성되지 않는 상호작용의 여러 형태에 따라 고려될 수 있다[10]. 객체의 색깔 변경이나 회전 등 그래픽 효과는 일관성이 거의 없는 수준이며 best-effort 방법으로 복제본들의 갱신 과정이 수행된다. 즉 해당 노드의 자원에 여유가 있을 때 갱신 과정이 수행되는 것이며 기본 사본과 복제본이 반드시 일치할 필요가 없음을 의미한다. 반면 플레이어가 제어하는 아바타의 점진적인 움직임은 낮은 일관성 수준으로 유지되며 데이터의 진부성에 대한 허용치를 제공한다. 즉 임의의 시점에서 아바타의 정확한 위치를 파악하기보다는 오차가 있는 근사치로 파악할 수 있어서 dead reckoning 기법[11]이 적용될 수 있다.

여러 객체를 대상으로 하는 액션을 위해서는 세 가지 일관성 수준이 있다. 예를 들어 아이템을 구입할 경우 가격은 중요한 속성이며 플레이어는 복제본의 가격보다 기본 사본의 가격이 낮아진 경우는 구입하겠지만 그렇지 않을 경우에는 구입을 포기할 수도 있다. 이와 같이 매우 중요한 속성을 가진 객체에 대해서는 높은 일관성 수준이 적용될 필요가 있다. 중간 정도의 일관성 수준일 경우 기본 사본의 가격과 복제본의 가격의 차이가 허용 범위 내에 있으면 요청된 액션이 성공될 수 있고 그렇지 않은 경우는 실패할 수 있는 것이다.

정확한 일관성 수준에서는 기본 사본과 차이가 있는 오래된 복제본의 판독을 허용하지 않는다. 예를 들면 두 아바타가 돈과 아이템을 동시에 교환할 경우 해당 플레이어들은 정확한 갱신 결과를 요구하게 된다. 따라서 요청 작업과 확인 작업이 locking 기법을 이용하여 수행될 필요가 있다. 모든 복제본에 대한 lock이 걸린 상태를 확인한 이후 아이템을 팔고 돈을 받는 액션이 수행되어야 하므로. 정확한 일관성 수준이 보장되려면 플레이어들 간에 메시지 전송량과 처리 시간이 급격히 증가하게 된다.

IV. 성능 분석

1. 실험 환경과 변수

본 논문에서는 클라이언트-서버 구조에 적용되는 Quake[12] 게임을 하이브리드 구조에 기반하여 수행되는 것으로 가정한다. 실제 대규모 멀티플레이어 게임 Quake에서 발생하는 플레이어의 액션은 다섯 가지의 이벤트로 처리된다. Move 이벤트(아바타가 새로운 위치로 이동함), Fire 이벤트(아바타가 로켓을 발사함), Impact 이벤트(로켓이 충돌하며 폭발됨), Damage 이벤트(아바타가 손상되거나 제거됨), Spawn 이벤트(아바타가 임의 지역에서 다시 생성됨) 들이다. 본 논문에서 Move 이벤트를 낮은 일관성 수준으로 설정하며 Fire, Impact 이벤트들은 중간 일관성 수준으로 설정한다. 일관성이 낮은 수준에서는 매초 여러 번 갱신되는 객체들의 위치 속성을 관련 플레이어들에 모두 전송하는 대신 각 플레이어의 노드에서 dead reckoning 기법을 적용하여 전송 횟수를 감소시킬 수 있다.

중간 수준의 일관성 이벤트인 Fire 이벤트로 인한 로켓의 아바타 공격 여부나 아바타가 공격을 피한 여부 등은 모든 플레이어들에 대하여 일치하는 결과가 나타날 필요가 있다. 따라서 각 객체의 속성이 갱신될 때마다 기본 사본 노드에서 복제본 노드로 모두 전송된다.

Damage/die와 Spawn 이벤트들은 모든 플레이어들에 대하여 동일하게 처리되어야 하고 게임 상태에 지속적인 영향을 미치게 되므로 중간보다 높은 수준의 일관성으로 설정할 필요가 있다. 본 논문에서는 정확한 일관성을 보장하기 위하여 해당 객체의 기본 사본을 가진 노드는 복제본에 대하여 lock을 먼저 걸고 확인 후 결과를 다시 복제본들에게 전송하기 때문에 통신 시간과 대역폭이 증가하게 된다. 이때 lock이 걸린 복제본에 대한 다른 플레이어의 액션은 지연되는 대신 실패하는 것으로 처리된다.

본 논문에서 플레이어가 아닌 인공지능으로 제어되는 NPC는 서버가 관리하는 것으로 가정한다. 서버가 N 개의 다른 플레이어들에게 전송하는 통신량 upload 대역폭을 $Server_B_{up}$라 하며 이는 갱신할 NPC 객체의 수 N_1, NPC 객체의 평균 크기(바이트) M_1, 초당 갱신되는 횟수 U, 갱신된 객체를 전송해야 하는 플레이어의 수 N의 곱으로 결정된다.

$$Server\_B_{up} = N_1 \times M_1 \times U \times N \quad (1)$$

각 플레이어가 다른 N-1 플레이어들과 서버로 전송해야 하는 통신량 upload 대역폭을 $Player_B_{up}$라 하며 이는 플레이어가 갱신할 객체의 평균 수 N_2, 객체의 평균 크기(바이트) M_2, 초당 갱신되는 횟수 U, 갱신된 객체를 N-1 플레이어들과 1 개의 서버에 전송해야 하므로 결국 N을 곱한 결과가 $Player_B_{up}$가 된다.

$$Player\_B_{up}=N_2 \times M_2 \times U \times N \quad (2)$$

본 논문에서 M_1과 M_2는 동일한 200 바이트, N은 64, U는 15 회, N_1은 128에서 512까지 128씩 증가, N_2는 32로 설정한 상태에서 Server_B_{up}와 Player_B_{up}에 대한 실험을 수행하였다.

2. 통신 대역폭 분석

서버가 없는 순수한 P2P 시스템일 경우 플레이어들이 NPC를 직접 제어하고 전송하게 되면 평균 Player_B_{up}는 증가하게 된다. 표 1에서 N_1이 128부터 512까지 128씩 증가할 경우에 대한 Server_B_{up}와 서버가 없을 경우 증가하는 Player_B_{up} 결과가 나타나 있다. 서버가 없을 경우 N_1이 128이면 64 개의 플레이어들이 각각 2 개의 NPC를 더 처리해야 하므로 N_2가 34가 되며 N_1이 512가 되면 각 플레이어는 8 개의 NPC를 더 처리하게 되므로 N_2가 40이 된다. 하이브리드 구조에서는 서버가 전송할 통신 대역폭 일부를 플레이어들이 처리함으로써 서버와 클라이언트들 간에 부하 조정이 가능하다. 서버가 512 개의 NPC를 처리하고 64 개의 플레이어들에게 전송할 경우 93.75 MBytes가 필요하나 512개의 NPC를 각 플레이어들에 분산시키면 각 플레이어는 8 개씩 더 처리하게 되어 5.86 MBytes에서 1.46 MBytes가 증가된 7.32 MBytes가 필요하게 된다.

표 1. Server_B_{up}와 Player_B_{up}
Table 1. Server_B_{up} and Player_B_{up}

(단위: MBytes)

NPC 수	0	128	256	384	512
Server_B_{up}	0	23.44	46.88	70.31	93.75
Player_B_{up}	5.86	6.23	6.59	6.96	7.32

표 1에서는 이벤트 유형을 구분하지 않고 모든 기본 사본 객체가 갱신될 때마다 복제본 객체로 전송한 결과이므로 사실상 중간 수준의 일관성이 설정된 것이다. 본 논문에서는 이벤트 유형의 일관성 수준을 고려하여 초당 갱신 횟수 U가 15일 때 낮은 일관성 수준인 Move 이벤트는 초당 10회, 중간 일관성 수준인 Fire, Impact 이벤트는 초당 3회, 정확한 일관성 수준인 Damage, Spawn 이벤트들은 초당 2회의 갱신 과정이 수행되는 것으로 가정하였으며 이에 대한 Player_B_{up} 결과가 그림 1에서 '수준별'로 표시된다. 이 결과에서 Move 이벤트는 10 회 갱신될 때 2 회만 복제본 노드로 전송되며 나머지는 각 플레이어의 노드에서 dead reckoning 기법이 적용되어 위치가 갱신되는 것으로 가정하였다. Fire, Impact 이벤트는 중간 일관성 수준이므로 기본 사본 객체가 갱신될 때마다 복제본 객체로 전송되며 Damage, Spawn 이벤트는 정확한 일관성 수준으로 모든 복제본에 lock을 획득하는 과정이 필요하므로 갱신시 평균 50바이트의 대역폭이 추가되는 것으로 가정하였다.

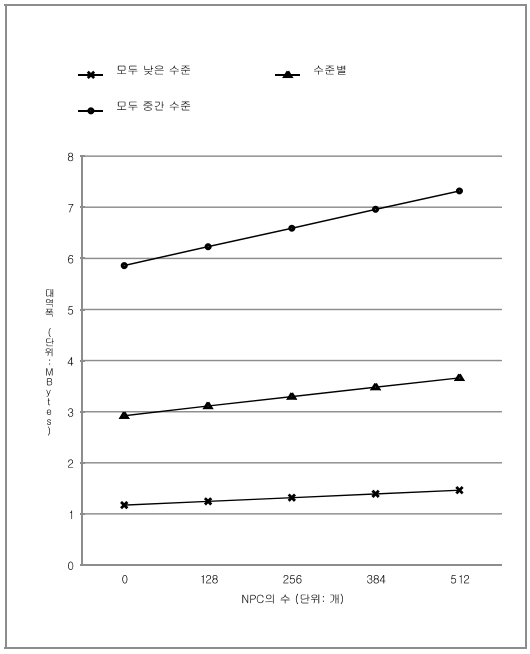

그림 1. NPC의 수에 따른 대역폭 Player_B_{up}
Fig. 1. Player_B_{up} for the number of NPC

그림 1에서 모든 이벤트들을 전부 낮은 일관성 수준으로 적용할 경우 갱신 횟수는 5회당 1회의 비율로 전송되며 '모두 낮은 수준'으로 표시되었다. 모든 이벤트들을 중간 일관성 수준으로 설정할 경우 기본 사본 객체가 갱신될 때마다 복제본 객체로 전송되므로 그림 1에서 '모두 중간 수준'의 대역폭은 '모두 낮은 수준'에 비하여 5배 정도 큰 것으로 나타났다. 이벤트의 일관성 수준을 고려한 '수준별' 대역폭은 '모두 중간 수준'보다는 작고 '모두 낮은 수준'보다는 큰 결과가 나타났다.

그림 2에서는 NPC의 수에 따라 이벤트의 일관성 수준을 고려한 Server_B_{up}의 결과가 나타났으며 '수준별' 결과가 '모두 중간 수준'보다는 작고 '모두 낮은 수준'보다는 큰 것으로 역시 나타났다.

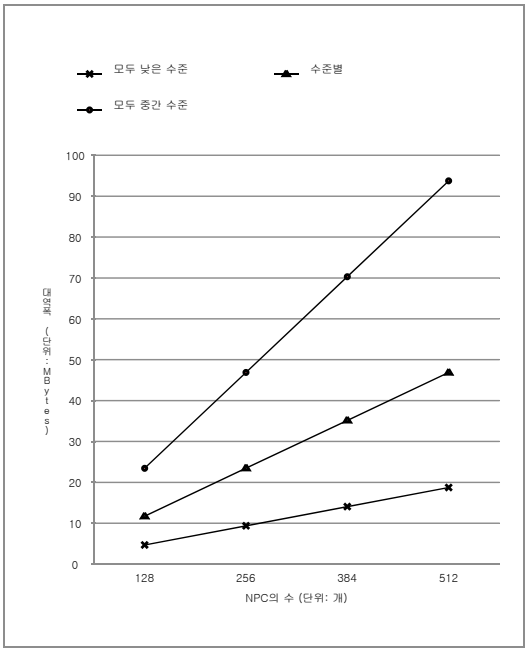

그림 2. NPC의 수에 따른 대역폭 Server_B_{up}
Fig. 2. Server_B_{up} for the number of NPC

실험에서 대역폭 Server_B_{up}와 Player_B_{up}를 분석한 결과 일관성 수준이 높을수록 필요한 통신 대역폭은 증가하는 것으로 나타났다. 본 실험에서 이벤트 수준별 일관성을 적용한 대역폭은 일관성이 낮은 수준의 대역폭보다 크고 중간 일관성 수준보다는 50% 이상 감소한 대역폭 결과가 나타났다.

서버의 download 대역폭 Server_B_{down}은 매초 N 플레이어들로부터 N_2X M_2 XU의 데이타를 수신하게 되므로 수식 (3)과 같다.

$$Server\_B_{down} = N_2 \times M_2 \times U \times N \quad (3)$$

플레이어의 download 대역폭 Player_B_{down}은 수식 (4)와 같다.

$$Player\_B_{down} = N_2 \times M_2 \times U \times (N-1) + N_1 \times M_1 \times U \quad (4)$$

각 플레이어는 다른 N-1 플레이어들로부터 N_2XM_2X U 만큼의 데이터를 수신하며 서버로부터 N_1X M_1 XU 만큼의 데이터를 수신하게 된다.

임의의 액션이 복제본과 기본 사본에서 상이한 갱신 결과를 유발할 경우 복구(roll back) 기법 등 비일관성 해결 프로토콜이[13] 필요하나 본 논문에서는 이에 관한 구체적 내용을 기술하지 않는다.

V. 결 론

대부분 MOG 게임들은 클라이언트-서버 구조를 기반으로 하고 있으나 본 논문에서는 P2P 구조의[14] 장점도 활용하기 위해 하이브리드 구조를 기반으로 하는 기본 사본 복제 기법을 제시한다. 제시된 하이브리드 구조에서 서버와 클라이언트 간에 게임 시스템의 통신 대역폭과 처리량을 분산함으로써 부하 조정이 가능하다. MOG의 이벤트별 일관성 수준에 따른 기본 사본과 복제본의 갱신 과정이 기술되었고 이에 대한 서버와 클라이언트의 통신 대역폭이 분석되었다. 일관성 수준이 높을수록 필요한 통신 대역폭이 증가하고 성능이 저하되는 결과가 확인되었다.

향후 대규모 멀티플레이어 온라인 게임에서 서버와 클라이언트의 역할을 구분하여 유지하는 기본 사본 객체의 수에 따라 부하를 조정하고 게임 액션의 특성에 따라 여러 수준의 일관성이 유지될 수 있으므로 이에 대한 구체적 연구와 실험적 결과가 더욱 필요할 것으로 전망된다.

References

[1] A. Yahyavi, B. Kemme, “Peer-to-peer architectures for massively multiplayer online games: A Survey,” Journal ACM Computing Surveys(CSUR), v. 46, no. 1, Oct. 2013.

[2] Suznjevic, M., Stupar, I. and Matijasevic, M., "Traffic Modeling of Player Action Categories in a MMORPG," International ICST Conf. on Simulation Tools and Techniques, pp. 280 - 287, 2012.

[3] Chen, J., Wu, B., Delap, M., Knutson, B., Lu, H., and Amza, C., “Locality aware dynamic load management for massively multiplayer games,” ACM SIGPLAN Symp. on Principles and Practice of Parallel Programming (PPoPP). ACM, pp. 289 - 300, 2005.

[4] J. Kim, "Reduction method of network bandwidth requirement for the scalability of multiplayer game

server systems," The Journal of the Institute of Internet, Broadcasting and Communication, v. 13, no. 4, pp. 29-38, 2013.

[5] N. Knutsson, H. Lu, W. Xu and B. Hopkins, " Peer-to-peer support for massively multiplayer games," IEEE International Conference on Computer Communications. 2004.

[6] A. Bharambe, A, J. Pang and S. Seshan. "Colyseus: A Distributed Architecture for Online Multiplayer games," International Conference on Networked Systems Design & Implementation, pp. 3-6, 2006.

[7] J. Goodman and C. Verbrugge, "A Peer Auditing Scheme for Cheat Elimination in MMOGs," International ACM SIGCOMM Workshop on Network & System Support for Games, pp. 9-14, 2008

[8] Y. Chu, S. G. Rao, S. Seshan and H. Zhang, "A ca se for end system multicast," IEEE Journal on Selected Areas in Communications, v. 20, no. 8, pp. 1456-1471, 2002..

[9] K. Zhang and B. Kemme, "Transaction models for massively multiplayer online games," International Symposium on Reliable Distributed Systems, IEEE, pp. 31 - 40, 2011.

[10] A. Chandler and J. Finney, "On the effects of loose causal consistency in mobile multiplayer games," International ACM SIGCOMM Workshop on Network & System Support for Games, ACM, pp. 1 - 11. 2005.

[11] L. Pantel and L. Wolf, "On the suitability of dead reckoning schemes for games," International ACM SIGCOMM Workshop on Network & System Support for Games, ACM, pp. 79-84, 2002.

[12] Doom, Quake, ID Software, Inc. http://www.idsoftware.com03., 2003.

[13] D. R. Jefferson, "Virtual time," ACM Trans. on Programming Language Systems, v. 7, no. 3, pp. 404 - 425, 1985.

[14] Y. Jung, S. Cho, J. Lee, K. Jeong, "A Design of P2P Cloud System Using The Super P2P," International Journal of Internet, Broadcasting and Communication, v. 7, no. 1, pp. 42-48, 2015.

저자 소개

김 진 환(정회원)

- 1986년 : 서울대학교 컴퓨터공학과 졸업(학사)
- 1988년 : 서울대학교 컴퓨터공학과 졸업(석사)
- 1994년 : 서울대학교 컴퓨터공학과 졸업(박사)
- 1994년 ~ 1995년 : 서울대학교 컴퓨터신기술공동연구소 특별연구원
- 1995년 ~ 현재 : 한성대학교 컴퓨터공학부 교수

<주관심분야 : 멀티미디어 시스템, 실시간 게임 시스템>

※ This research was financially supported by Hansung University