

<http://dx.doi.org/10.7236/JIIBC.2015.15.4.135>

JIIBC 2015-4-17

기본 사본을 갖는 P2P 멀티플레이어 게임 구조의 수준별 일관성 제어 기법

Multi-level Consistency Control Techniques in P2P Multiplayer Game Architectures with Primary Copy

김진환*

Jin-Hwan Kim *

요 약 멀티플레이어 온라인 게임(MOG)은 수백 또는 수천명의 플레이어들을 위한 게임이며 대부분 인터넷을 이용하여 수행된다. MOG를 위한 P2P(peer-to-peer) 구조는 잠재적으로 규모조정성 증대, 비용 절감, 성능 향상 등이 가능하다. P2P 기반 게임의 기본 개념은 게임 상태를 플레이어들에게 분산 시켜서 각 플레이어가 처리, 통신, 저장 업무를 수행하도록 하는 것이다. 객체에 대한 갱신 과정이 기본 사본에서 먼저 수행되는 기본 사본 기반 복제 기법에서도 객체들의 기본 사본은 플레이어들 간에 분산된다. 대부분의 멀티플레이어 게임은 각 객체마다 강한 일관성을 제공하기 위하여 기본 사본 모델을 사용한다. 게임은 여러 수준의 민감도를 가지며 각각의 일관성 요건에 따라 영역화될 수 있는 다양한 유형의 게임 액션들로 구성된다. 게임 내에서 각 액션의 유형마다 적절한 일관성 수준이 유지될 때 본 논문에서는 게임 개발업체가 성능과 일관성 간에 적절한 상충 관계를 선택할 수 있도록 한다. 기본 사본 모델을 갖는 P2P 게임 구조에 대한 성능 분석 결과가 기술된다.

Abstract A Multiplayer Online Game(MOG) is a game capable of supporting hundreds or thousands of players and is mostly played using the Internet. P2P(peer-to-peer) architectures for MOGs can potentially achieve high scalability, low cost, and good performance. The basic idea of many P2P-based games is to distribute the game state among peers and along with it processing, network, and storage tasks. In a primary-copy based replication scheme where any update to the object has to be first performed on the primary copy, this means distributing primary copies of objects among peers. Most multiplayer games use a primary-copy model in order to provide strong consistency control over an object. Games consist of various types of actions that have different levels of sensitivity and can be categorized according to their consistency requirements. With the appropriate consistency level of each action type within a game, this paper allows developers to choose the right trade-off between performance and consistency. The performance for P2P game architecture with the primary-copy model is evaluated through simulation experiments and analysis.

Key Words : consistency control, P2P, multiplayer, primary-copy, performance

*정회원, 한성대학교 멀티미디어공학과
접수일자 2015년 4월 29일, 수정완료 2015년 6월 7일
게재확정일자 2015년 8월 7일

Received: 29 April, 2015 / Revised: 7 June, 2015 /

Accepted: 7 August, 2015

*Corresponding Author: kimjh@hansung.ac.kr

Dept. of Multimedia Engineering, Hansung University, Korea

I. 서 론

멀티플레이어 온라인 게임(MOG; Multiplayer Online Game)은 인터넷을 이용하여 동시에 수백 또는 수천명의 플레이어들이 참여할 수 있는 게임이다^[1, 2]. 인터넷을 기반으로 하는 World of Warcraft^[3], Eve Online^[4], Final Fantasy XI^[5] 등은 현재 사업 규모가 계속 증가하는 MOG들이다. 게임에 참여하는 플레이어들이 많을수록 게임 환경은 더욱 복잡하고 상호작용과 흥미가 증가하게 된다. 대규모 가상 세계인 게임 세계는 다양한 유형의 객체와 플레이어나 인공지능에 의하여 제어되는 캐릭터들로 구성된다. 수많은 플레이어들이 다른 플레이어들과 및 게임 환경과 상호작용하기 위하여 자신의 아바타를 직접 제어한다. 플레이어들은 MOG에서 기본적 상호작용 방법인 액션을 사용하여 게임 세계에서 캐릭터나 객체의 이동, 아이템의 획득, 다른 플레이어와 아이템의 교환 기능 등을 수행하게 된다. 따라서 게임의 상태는 끊임없이 변경되며 참여한 플레이어들에 의해서 공유된다. 변경되는 게임 세계는 모든 플레이어들에게 동일하게 나타나며 특정 플레이어에 의해 수행된 액션은 다른 플레이어들에게 관찰될 수 있어야 한다.

공통적으로 게임 엔진은 미들웨어로써 구현되며 플레이어들의 요청 작업을 수신하고 게임 세계를 관리하며 갱신 결과를 다시 플레이어들에게 전송하는 역할을 수행한다. 본 논문에서는 게임 실행과 게임 상태의 전송이 서버의 게임엔진에 의해 전적으로 제어되는 클라이언트-서버 구조^[6] 대신 P2P(Peer-to-peer) 구조를 게임의 기본 구조로 가정한다. 컴퓨팅 작업과 통신량을 플레이어들에게 분산시킬 수 있는 P2P 구조는 잠재적으로 규모조정성 증가, 비용 절감, 성능 향상 등을 추구할 수 있다^[7].

임의의 플레이어가 게임에 참여할 때 다양한 유형의 게임 객체로 구성되는 게임 세계의 인스턴스를 수신하게 된다. 대부분의 게임 엔진들은 기본 사본 복제(primary-copy replication) 방법을 사용한다^[8]. 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. P2P 구조에서 각 플레이어는 자신이 책임지는 기본 사본과 다른 플레이어들이 책임지는 기본 사본의 복제본

을 가지게 된다. 특정 플레이어가 복제본 객체에 대하여 갱신 과정을 수행할 경우 객체의 기본 사본으로 갱신 과정이 전송되며 기본 사본을 소유한 해당 플레이어가 갱신 과정의 허용 여부를 결정하게 된다.

MOG같은 분산 구조에서 수많은 플레이어들이 게임에 참여할 경우 복제본 수와 갱신 비율이 증가함에 따라 모든 복제본에 대한 갱신 과정이 정확히 동시에 수행되긴 사실상 어렵다. 즉 모든 플레이어들에게 정확한 게임 상태를 제공해서 게임의 일관성을 유지해야 하나 충돌이 발생하는 갱신 현상이 여러 사이트에서 동시에 실행될 수 있으며 결과적으로 비일관적인 상태가 발생될 수 있다^[9, 10].

본 논문에서는 P2P 구조의 멀티플레이어 게임에서 수행되는 액션의 중요도에 따라 적절한 일관성이 유지되는 기법을 사용한다. 예를 들어 서로 다른 두 플레이어가 동일한 객체를 획득하려 할 경우 한 플레이어가 성공하면 나중에 요청한 다른 플레이어는 획득에 실패하게 된다. 객체의 기본 사본을 가진 사이트에서 요청 작업의 순서를 정확히 유지함으로써 일관성을 보장해야 한다. 게임에서 일관성을 낮은 수준으로 유지할 수 있는 액션과 높은 수준으로 유지해야 하는 액션을 수행할 경우 플레이어들 간에 통신량과 처리 시간이 달라진다. 일관성 수준을 높일수록 시스템의 성능은 저하될 수 있으므로 향후 게임 개발업체는 일관성과 성능의 상충 관계를 고려하여 적절한 일관성을 선택할 필요가 있다.

본 논문에서는 MOG를 위한 P2P 구조의 기본 사본 모델을 2장에서 분석하며 3장에서 여러 수준으로 유지되는 일관성 기법이 기술된다. 4장에서는 기본 사본을 가진 P2P 게임에서 수준별 일관성 제어 기법과 통신 대역폭이 분석된 결과가 기술되며 5장에서 결론이 기술된다.

II. 기본 사본을 갖는 P2P 게임 구조

1. 기본 사본에 대한 갱신 과정

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 관련된 플레이어들에게 변경된 내용을 통보해야 한다. 이 과정은 기본 사본 복제본 기법^[8]

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 객체를 책임지는 사본을 기본 사본(primary copy)이라고 하며 나머지 사본들을 2차 사본(secondary copy) 또는 복제본(replica)이라고 정의한다.

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. 기본 사본을 보유한 플레이어가 갱신 요청의 수락 여부를 결정하며 객체를 갱신했을 경우 해당 객체의 복제본을 가진 모든 플레이어들에게 갱신 결과를 전송함으로써 객체에 대한 갱신 과정이 종료되는 것이다.

게임 세계를 여러 지역으로 분할하는 P2P 시스템에서는 임의의 플레이어가 조정자 역할을 수행할 수도 있다^[11]. 조정자는 자신의 노드에 모든 게임 객체들의 기본 사본을 유지하고 자신이 유지하는 객체에 대한 모든 갱신 요청을 수신하며 갱신 결과를 멀티캐스트 기법^[12]으로 다른 플레이어들에게 전송하게 된다. 이러한 시스템에서는 조정자에 많은 계산 부하가 걸릴 수 있는 단점이 존재하며 작업부하의 균형을 유지하기 위해 과부하가 걸린 노드들의 경우 해당 영역의 크기를 감소시킬 수 있다.

반면 조정자 없이 모든 플레이어들이 상호 간에 직접 통보하는 all-to-all 시스템에서는 각 플레이어가 자신의 아바타를 비롯한 객체의 기본 사본을 유지하며 갱신 작업도 자신의 노드에서 수행할 수 있게 된다. 객체를 갱신한 결과는 직접 복제본을 가지고 있는 다른 플레이어들에게 전송하게 된다. 기본 사본 모델에서 객체의 모든 복제본은 읽기 전용이며 플레이어가 임의로 변경할 수 없다. 만일 둘 이상의 플레이어들이 동시에 갱신 요청을 할 경우 기본 사본을 가진 플레이어는 갱신 요청 작업들을 순서화하여 하나의 갱신 요청 작업만 허용하게 된다. 갱신된 기본 사본은 다른 플레이어들에게 멀티캐스트 기법을 이용하여 동시에 전송된다.

2. P2P 구조

P2P 구조에서는 모든 노드가 서버와 클라이언트 역할을 동시에 수행할 수 있다. 기본 사본을 갱신할 경우 해

당 플레이어는 관련된 다른 모든 플레이어들에게 갱신된 내용을 전송하게 된다. 반면 클라이언트-서버 구조에서는 모든 객체의 기본 사본이 서버에 존재하게 되며 클라이언트들은 복제본만을 유지하게 된다.

P2P 구조에서 작업 부하는 모든 노드들에 분산되며 새로운 노드가 추가될 때 시스템에 새로운 자원도 추가되는 것이므로 규모조정성을 손쉽게 증가시킬 수 있다. 따라서 P2P 구조는 규모조정성에 대한 잠재력이 매우 크다. 게임 업체 입장에서 추가 자원에 대한 비용 부담이 작다는 것이 이 구조의 특별한 장점이기도 하다. 부하가 모든 peer 즉 플레이어가 있는 노드에 분산되기 때문에 고가의 서버를 유지할 필요가 없으며 게임 관리에 대한 책임도 각 노드가 소량의 부하를 담당하므로 전체 노드들에 분산된다. 임의의 플레이어 노드에서 결함이 발생하더라도 다른 많은 플레이어들에게 부정적인 영향을 미치지 않게 된다. 또한 플레이어들 간에 직접 통신이 적절하게 수행되며 서버에게 갱신 상태를 전송한 후 서버가 다시 플레이어들에게 전송하는 추가 과정이 필요없기 때문에 통신 지연시간을 클라이언트-서버 구조보다 감소시킬 수 있다.

그러나 P2P 구조는 여러 단점을 가지고 있으며 가장 취약한 부분은 보안이다^[10]. 부정행위 자체는 클라이언트-서버 구조보다 P2P 구조에서 더 쉽게 수행될 수 있다. 또한 게임의 전체 상태를 유지하는 중앙 서버가 없기 때문에 게임 관리와 제어가 더욱 어려워지는 문제도 있다. 즉 게임 상태가 플레이어들에게 분산되어 있기 때문에 게임 업체가 게임을 완벽하게 제어하기 어렵다.

III. 수준별 일관성 제어 기법

1. 비일관성의 발생

P2P 구조는 동시에 실행되는 갱신 과정의 충돌이 여러 노드에서 발생할 수 있기 때문에 일관성 제어 기법의 적용이 더욱 어렵고 비일관성이 쉽게 발생할 수 있다. 그러나 모든 갱신 과정이 기본 사본에서 먼저 수행되는 기본 사본 모델에서는 복제본들이 존재하는 노드에서 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 메커니즘이 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 있다. 특별히 여러 개의 객체 상태를 변경하고 원자성(atomicity)이 요구될 경우에는 신뢰성이 있는 TCP 또는 임계 영역을 위한 2 단계 commit 프로토콜^[1]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.

2. 플레이어 액션

플레이어들은 MOG의 기본적 상호작용 방법인 액션을 사용하여 게임 세계에서 이동, 아이템의 획득이나 포기, 다른 플레이어와의 아이템 교환 등을 수행하게 된다. 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. 동일한 객체에 대한 갱

신 과정이 동시에 발생하여 충돌할 경우 일관성 유지를 위해서 상호작용의 유형 파악이 중요하다.

3. 여러 수준의 일관성 제어

플레이어의 모든 상호작용을 격리성(isolation)과 원자성을 보장하는 트랜잭션으로 처리할 경우 가장 강력한 일관성 형태를 가질 수 있다^[1]. 그러나 트랜잭션으로 처리할 경우 비용이 증가하고 모든 게임 액션에 적용시키는 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 제공하며 결과적으로 궁극적 일관성을 유지하는데 주력한다.

다양한 메커니즘에 의해서 구현되는 일관성의 여러 수준은 상이한 객체와 동일한 중요도로 구성되지 않는 상호작용의 여러 형태에 따라 고려될 수 있다^[13]. 예를 들면 많은 가상 게임 객체들은 자산으로 고려되며 실제 금액으로 거래될 수 있어서 일관성 제어를 매우 중요한 요건으로 설정할 필요가 있다. 반면 플레이어의 위치 갱신은 낮은 일관성 요건을 갖게 되며 궁극적인(eventual) 일관성으로도 충분히 유지할 수 있다.

한 액션이 두 개 이상의 객체들을 갱신할 경우 상이한 기본 사본들 사이에서 조정자의 역할도 필요하며 원자성과 격리성이 보장되는 트랜잭션으로 실행될 필요가 있다. 그러나 locking이나 다수결 기법처럼 기존의 프로토콜들은 비용이 많이 발생하기 때문에 대부분의 게임들은 약한 일관성 수준을 채택하게 된다. 여러 유형으로 분석되는 게임 액션은 일관성 요건에 따라 여러 영역으로 분류된다^[1]. 상호작용마다 상이한 수준의 민감성을 갖는 특성에 따라 영역화된 것이며 각 영역을 위한 일관성 알고리즘이 제공된다. 한 개의 객체만 대상으로 하는 액션에 대해서는 두 가지의 낮은 수준이 유지된다.

가. 일관성이 전혀 없는 수준

복제본들이 best-effort 방법으로 갱신된다. 즉 객체의

색깔 변경이나 회전 등 그래픽 효과는 이 정도 수준으로 충분하며 해당 노드의 자원에 여유가 있을 때 갱신 과정이 수행되는 것이다. 기본 사본과 복제본이 반드시 일치할 필요가 없음을 의미한다.

나. 낮은 일관성 수준

데이터의 진부성에 대한 허용치를 제공한다. 플레이어가 제어하는 아바타의 점진적인 움직임이 이 영역으로 포함되기에 적절하다. 다른 플레이어들은 임의의 시점에서 아바타의 정확한 위치를 파악하기 보다는 오차가 있는 근사치로 파악할 수 있다. 즉 dead reckoning 기법^[14]이 낮은 일관성 메카니즘의 예로 고려될 수 있다. 게임에서 플레이어의 위치를 이전 프레임에서의 위치, 속도, 움직임 방향 등에 기반하여 계산해서 다음 프레임에 적용하는 dead reckoning 기법은 향후 발생할 수 있는 충돌을 탐지하기 위해서도 사용되며 메시지들이 지연시간이나 분실로 인하여 정시에 도착하지 못할 때도 사용된다. 모든 복제본들이 dead reckoning 기법을 수행할 수 있으며 이 경우 dead reckoning으로 계산된 상태와 실제 상태와의 차이가 허용치보다 클 경우 기본 사본은 상태 갱신 결과를 모든 복제본에게 전송하게 된다. dead reckoning은 일반적으로 수백 millisecond 정도의 지연시간에 허용될 수 있으며 플레이어의 움직임에 국한되어 사용되고 있다. 실제 게임에서 갱신 과정을 자주 전송하지 않는 플레이어들에 대하여 dead reckoning 기법이 사용될 수 있다.

여러 객체를 대상으로 하는 액션을 위해서는 세 가지 일관성 수준이 있다.

다. 중간 일관성 수준

현재 게임 미들웨어가 대부분의 액션을 처리할 때 중간 수준의 일관성을 고려하며 정확한 갱신 결과가 전송될 것을 요구하고 있다. 복제본의 진부성은 기본 사본이 갱신된 직후 복제본으로 전송될 때까지의 시간동안만 허용된다. 예를 들어 동시에 두 플레이어가 동일한 아이템을 획득하려 할 경우 아이템을 보유한 노드는 요청 작업을 순서적으로 처리하여야 한다. 첫 번째 액션은 성공하지만 두 번째 액션은 실패하게 되며 이 결과가 다른 플레이어들에게 정확히 전송되어야 한다.

라. 높은 일관성 수준

예기치 못한 결과는 가끔 바람직하지 않은 결과를 유발할 수 있다. 예를 들면 아이템을 구입할 경우 가격은

중요한 속성이며 플레이어는 복제본의 가격보다 기본 사본의 가격이 낮아진 경우는 구입하겠지만 그렇지 않은 경우에는 구입을 포기할 수도 있다. 이와 같이 매우 중요한 속성을 가진 객체에 대해서는 높은 일관성 수준이 적용될 필요가 있다.

중간 일관성 수준일 경우 기본 사본의 가격과 복제본의 가격의 차이가 허용 범위 이내 일때는 요청된 액션이 성공될 수 있고 그렇지 않은 경우는 실패할 수 있는 것이다. 즉 복제본의 가격보다 기본 사본의 가격이 높더라도 허용 범위 이내일 경우에는 플레이어가 구입할 수도 있음을 의미한다. 성능 관점에서는 중간 일관성 수준과 높은 일관성 수준의 차이가 크지 않을 수 있다.

마. 정확한 일관성 수준

정확한 일관성 수준에서는 기본 사본과 차이가 있는 오래된 복제본의 판독을 허용하지 않는다. 예를 들면 두 아바타가 돈과 아이টে임을 동시에 교환할 경우 해당 플레이어들은 정확한 갱신 결과를 요구하게 된다. 따라서 요청 작업과 확인 작업이 locking 기법을 이용하여 수행될 필요가 있다. 기본 사본을 갖고 있는 플레이어는 아이템 구입에 대한 요청을 받을 경우 다른 플레이어들이 동일 아이템 구입을 요청할 수 없도록 모든 복제본에 대한 lock을 걸게 된다. 모든 복제본에 대한 lock이 걸린 상태를 확인한 이후 아이টে임을 팔고 돈을 받는 액션이 수행되며 이 결과는 해당 복제본을 가진 다른 노드들에 즉시 전송된다. 정확한 일관성 수준이 보장되려면 플레이어들 간에 메시지 전송량과 처리 시간이 증가하게 된다.

4. 이벤트의 일관성 요건

게임 세계에서 발생하는 플레이어의 액션은 이벤트로 처리되며 이벤트의 본질에 따라 상이한 일관성 요건을 가지게 된다^[15]. 실시간 이벤트는 엄격한 시간 요건을 갖는 반면 약한 일관성 요건을 갖게 되며 반대로 일관성 이벤트는 약한 시간 요건과 강한 일관성 요건을 갖게 된다. 강한 시간 요건과 일관성 요건을 모두 가지는 일관성 실시간 이벤트도 존재할 수 있다. 사람이 이벤트에 대한 결과를 감지할 수 있는 시간이 100ms보다 큰 이벤트들은 일관성 이벤트로 분류되며 이보다 작은 시간을 갖으면 실시간 이벤트 또는 일관성 실시간 이벤트로 분류된다^[15]. 실제 대규모 멀티플레이어 게임 Quake^[16]의 이벤트들은 다섯 가지 유형으로 분류된다.

- 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

실제 Fire, Impact 이벤트들은 일관성 실시간 요건을 가지며, Move 이벤트는 실시간 요건만 가지고 Damage/die, Spawn 이벤트들은 일관성 요건만 가지는 것으로 분류된다.

IV. 성능 분석

1. 실험 환경과 변수

P2P 기반 게임의 기본 개념은 게임 상태를 플레이어들에 분산시켜서 처리, 통신, 저장 등의 과정이 수행되도록 하는 것이다^[17]. 객체의 기본 사본을 사용하는 기법도 모든 객체의 기본 사본을 플레이어들에게 분산시키고 있으며 P2P 시스템을 사용하여 갱신 작업을 전송하게 된다. 본 논문에서는 클라이언트-서버 구조에 기반한 Quake 게임을 P2P 구조에 기반하여 수행되는 것을 가정한다. 각 플레이어가 다른 플레이어들에게 전송해야 하는 통신량을 upload 대역폭 B_{up} 라 하며 이는 갱신할 객체의 평균 크기(바이트) M , 갱신할 복제본 객체의 평균 수 $N-1$, 초당 갱신되는 횟수 U , 갱신된 객체를 전송해야 하는 플레이어의 수 $(N-1)$ 의 곱으로 결정된다.

$$B_{up} = MX (N-1)XUX (N-1) \quad (1)$$

각 플레이어의 노드에서 플레이어의 기본 사본 아바타 캐릭터 1 개와 $N-1$ 개의 복제본들을 포함하여 N 개의 객체들이 초당 U 회 갱신되는 것으로 가정하였다. 본 논문에서 M 은 200 바이트, N 은 16 개에서 64 개, U 는 15 회, 전송할 플레이어의 수 $(N-1)$ 은 15 명에서 63 명의 범위에서 B_{up} 에 대한 실험을 수행하였다.

본 실험에서는 Quake의 실시간 이벤트인 Move 이벤트를 낮은 일관성 수준으로 설정한다. 플레이어의 움직임 행위는 해당 캐릭터의 위치와 목적지 속성을 사용하여 구현된다. 일정한 시간 간격을 두고 게임은 현재의 위치 속성을 판독하며 약간씩 위치를 갱신하여 목적지에 접근하게 된다. 일관성이 낮은 수준에서는 매초 여러 번

갱신되는 객체들의 위치 속성을 관련 플레이어들에 모두 전송하는 대신 매초 1-2회 이하로 전송하고 나머지는 각 플레이어의 노드에서 dead reckoning 기법이 적용되어 위치가 갱신되는 것으로 가정한다.

일관성 실시간 이벤트인 Fire, Impact 이벤트들은 중간 일관성 수준으로 설정한다. 예를 들어 Fire 이벤트로 인한 로켓의 아바타 공격 여부나 아바타가 공격을 피한 여부 등은 모든 플레이어들에 대하여 일치하는 결과가 나타날 필요가 있다. 중간 수준의 일관성에서는 각 객체의 속성이 갱신될 때마다 기본 사본 노드에서 복제본 노드로 모두 전송된다. 일관성 이벤트인 Damage/die와 Spawn 이벤트들은 모든 플레이어들에 대하여 동일하게 처리되어야 하고 게임 상태에 지속적인 영향을 미치게 되므로 높은 일관성 또는 정확한 일관성 수준으로 설정한다. 정확한 일관성을 보장하기 위하여 해당 객체에 대한 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.

2. 대역폭 분석

본 실험에서 N 개의 객체들이 매초 U 회 갱신되는 것을 가정하였다. U 를 15로 설정할 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 가정하였다. 높은 일관성 수준은 오차 범위를 결정하는 논리 과정이 포함됨에 따라 중간 일관성 수준과의 대역폭의 차이가 크지 않아서 Damage, Spwan 이벤트들은 정확한 일관성 수준으로 가정하여 실험이 수행되었다.

이벤트들의 일관성 수준을 동일하게 적용한 경우와 다르게 적용한 경우의 대역폭 B_{up} 비교 결과는 그림 1과 같다. 모든 이벤트들을 낮은 일관성 수준으로 적용할 경우 Move 이벤트는 10회 갱신될 때 2회만 복제본 노드로 전송되며 중간 일관성 수준인 Fire 이벤트와 정확한 일관성 수준의 Damage 이벤트도 같은 비율로 전송된다. 즉 기본 사본 객체의 속성이 갱신될 때마다 복제본 객체로 전송되는 것이 아니라 5회당 1회만 전송되는 것이다. 일관성이 낮은 수준에서는 매초 10 회씩 갱신되는 객체들

의 속성을 관련 플레이어들에 모두 전송하는 대신 매초 2회 정도만 전송하고 나머지는 각 플레이어의 노드에서 dead reckoning 기법이 적용되어 위치가 갱신되는 것으로 가정한다. 본 실험에서는 낮은 수준의 일관성이 유지될 경우 매초 2회만 기본 사본 객체를 가진 노드가 나머지를 복제본을 가진 노드들에게 전송하는 것으로 가정하여 실험을 수행하였다.

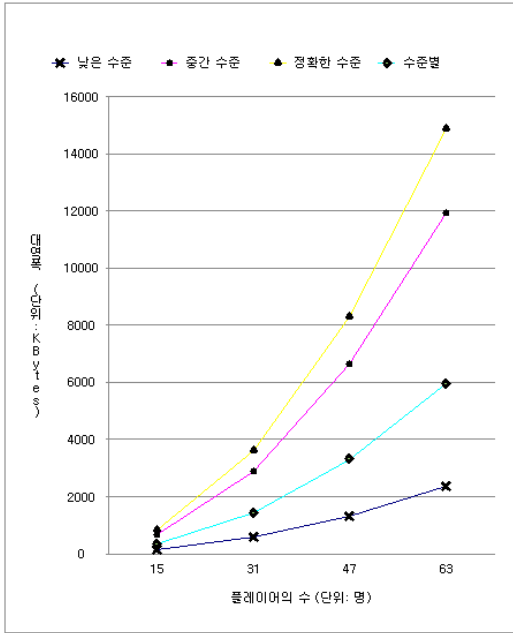


그림 1. 플레이어의 수에 따른 대역폭 B_{up}
 Fig. 1. B_{up} for the number of players

모든 이벤트들을 중간 일관성 수준으로 설정할 경우 기본 사본 객체가 갱신될 때마다 복제본 객체로 전송되므로 낮은 일관성 수준의 대역폭에 비하여 5배 정도 큰 것으로 나타났다. 그리고 모든 이벤트들을 정확한 일관성 수준으로 설정할 경우 모든 복제본에 lock을 획득하는 과정이 필요하므로 평균 50바이트의 대역폭이 추가되는 것으로 가정하였다. 정확한 일관성 수준의 대역폭은 중간 일관성 수준에 비하여 25%가 증가되는 것으로 나타났다며 실제 통신 시간은 2배 이상인 것으로 분석되었다.

실험에서 대역폭 B_{up}을 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 수

준보다는 50% 이상 감소한 대역폭 결과가 나타났다.

수식 1과 동일한 기본 사본 모델에서 각 플레이어의 download 대역폭 B_{down}은 매초 (N-1) 명의 플레이어로부터 수식 1의 MX (N-1)XU를 수신하게 되며 B_{up}과 크기가 같다. 따라서 각 플레이어마다 필요한 통신 대역폭은 B_{up}과 B_{down}을 합한 대역폭이며 결과적으로 B_{up}의 두 배가 된다.

임의의 액션이 복제본과 기본 사본에서 상이한 갱신 결과를 유발할 경우 비일관성 해결 프로토콜이 필요하게 된다^[9]. 충돌 해결의 대안책은 일관적인 이전의 상태로 복구(roll back)하는 것이다^[18, 19]. 그러나 복구 방법은 플레이어들을 완전히 만족시키기 어렵기 때문에 본 논문에서는 이러한 방법을 고려하지 않는다.

V. 결론

대부분 MOG 게임들은 클라이언트-서버 구조를 기반으로 하고 있으나 본 논문에서는 기본 사본을 갖는 P2P 구조를 기반으로 액션에 해당하는 이벤트의 일관성 수준에 따라 객체의 갱신 과정이 수행된다. P2P 구조는 고가의 서버를 유지하기 위한 비용을 감소시킬 수 있고 게임의 규모조정성도 향상시킬 수 있다. 반면 P2P 구조는 제한된 대역폭, 부정행위에 대한 취약점, 데이터의 일관성 보장 등 해결되어야 하는 여러 문제들을 가지고 있다. 본 논문에서는 P2P 구조에서 게임 시스템의 이벤트별 일관성 수준을 고려하여 기본 사본과 복제본을 갱신하는 과정을 고려하였고 이에 대한 통신 대역폭이 분석되었다. 일관성 수준이 높을수록 필요한 통신 대역폭이 증가하고 성능이 저하되는 결과가 확인되었다.

향후 대규모 멀티플레이어 온라인 게임에서 게임의 행위와 객체의 특성에 따라 여러 수준의 일관성이 유지될 수 있으므로 이에 대한 구체적인 연구와 실험적 결과가 더욱 필요할 것으로 전망된다.

References

[1] K. Zhang and B. Kemme, "Transaction models for massively multiplayer online games," International Symposium on Reliable Distributed Systems,

- IEEE, pp. 31 - 40, 2011.
- [2] Suznjivic, 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] Blizzard Entertainment, <http://www.worldofwarcraft.com/pvp/battlegrounds>, 2011.
- [4] EVE Online, <http://www.eveonline.com>, 2011.
- [5] Final Fantasy XI, <http://www.playonline.com/ffllus>, 2011.
- [6] 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.
- [7] 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.
- [8] N. Knutsson, H. Lu, W. Xu and B. Hopkins, "Peer-to-peer support for massively multiplayer games," IEEE International Conference on Computer Communications. 2004.
- [9] A. Bhambe, 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.
- [10] 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.
- [11] S. Y. Hu, S. C. Chang and J. R. Jiang, "Voronoi state management for peer-to-peer massively multiplayer online games," International IEEE Conference on Consumer Communications & Networking, pp. 1134 - 1138), 2008.
- [12] Y. Chu, S. G. Rao, S. Seshan and H. Zhang, "A case for end system multicast," IEEE Journal on Selected Areas in Communications, v. 20, no. 8, pp. 1456-1471, 2002.
- [13] 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.
- [14] 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.
- [15] A. Hsu, J. Ling, Q. Li and C. C. Jay Kuo, "On the design of Multiplayer On-line Video Game Systems," SPIE ITCOM, pp. 180-191, 2003.
- [16] Doom, Quake, ID Software, Inc. <http://www.idsoftware.com03>, 2003.
- [17] L. Chan, J. Yong, J. Bai, B. Leong and R. Tan, "Hydra: a massively-multiplayer peer-to-peer architecture for the game developer," International ACM SIGCOMM Workshop on Network & System Support for Games, ACM, pp. 37-42, 2007.
- [18] D. R. Jefferson, "Virtual time," ACM Trans. on Programming Language Systems, v. 7, no. 3, pp. 404 - 425, 1985.
- [19] S. Ferretti, "A synchronization protocol for supporting peer-to-peer multiplayer online games in overlay networks," International Workshop on Distributed Event-Based Systems, ACM, pp. 83 - 94, 2008.

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 멀티미디어공학과 교수
<주관심분야 : 멀티미디어 시스템, 실시간 게임 시스템>