

## 대용량 온라인 게임을 위한 확률 기반 메시지 동기화 기법

윤청하<sup>0</sup>, 백두원

송실대학교 미디어학과

chunggha@gmail.com<sup>0</sup>, paik@computing.ssu.ac.kr

### A Probability based Message Synchronization Mechanisms for Massively Multi-Player Online Game

Chung-Ha Yun<sup>0</sup>, Du-Won Paik

School of Media, Soongsil Univ.

#### 요약

인터넷을 이용하여 수천 명이 동시에 접속할 수 있는 온라인 게임에서는 참여자들의 하드웨어와 네트워크 성능은 매우 다를 수 있으므로 메시지 동기화는 반드시 해결되어야 할 과제이다. 동기화를 위하여 메시지를 받은 즉시 처리하지 않고 일정한 대기시간이 지난 후 처리하는 방법이 많이 사용되는데, 대기시간이 길어질수록 공정성은 좋아지나 응답성이 떨어지므로 적절한 대기시간을 결정하는 것은 쉬운 문제가 아니다. 본 논문에서는 대용량 일인칭 액션 게임의 특징과 참여자간의 상호작용 확률을 고려한 대기시간 결정 방법을 제안했다. 제안된 방법을 구현하고 가상환경 하에서 메시지를 생성하는 모의 실험을 하여 제안된 방법의 효용성을 검증했다.

#### Abstract

In the online game that thousands of participants connect to a game world by using Internet simultaneously, participants of the game world may have different hardwares and network environments, and messages of participants should be synchronized. A solution for the synchronization problem is to wait for a certain amount of time until the server processes the message. It is not easy to determine a suitable waiting period because of the trade-off between the fairness and the responsiveness. In this paper we propose a method to decide a waiting period by considering features of the MMOFPS game and the probability of interaction among participants. We implement the proposed method and perform experiments for the performance evaluation.

Keyword : MMOFPS, Networked Virtual Environment, Synchronization, Waiting Period, Area of Interest

#### 1. 서론

인터넷의 발전으로 흩어져있는 다수의 참여자들이 하나의 게임공간을 공유하며 함께 게임을 할 수 있는 온라인게임이 확산되고, 하드웨어와 네트워크 성능이 향상되면서 수

천 명의 참여자가 동시 접속하는 온라인게임이 가능해졌다. 수백 명이 한 세계에서 동시에 일인칭 액션(FPS) 게임을 플레이하는 형태로 정의되는 MMOFPS (Massively Multiplayer Online First Person Shooting) 게임이 한 예인데, 이 게임 장르는 게임 세계의 지리적인 환경에서 참여자의 주

변에 있는 다른 참여자들이 자주 바뀔 수 있고, 참여자간의 상호작용이 매우 빈번히 일어나며 제한적인 시야를 통해 플레이해야 하는 특징이 있다.

게임 세계의 변화정보는 모든 참여자 간에 즉시 공유되어야 하며 참여자의 수가 증가하여도 환경의 일관성이 유지되어야 한다. 하지만, 게임 세계에 참여하는 참여자들의 네트워크 환경과 하드웨어 성능은 서로 매우 다를 수 있으므로 다양한 환경 하에 있는 참여자들에게 동일한 대우를 하는 공정성(fairness)과 적절한 응답성(responsiveness)을 위한 동기화(synchronization)는 반드시 해결되어야 할 과제이다. 참여자 환경의 차이 때문에 생기는 불공정함을 해결하기 위한 대표적인 방법은 메시지를 받은 즉시 처리하지 않고 일정한 대기시간(waiting period)이 지난 후 처리하는 방법이다. 대기시간이 길어질수록 참여자간의 공정성은 높아지지만 응답성이 저하된다. 대기시간은 공정성과 응답성의 균형을 조절 할 수 있는 중요한 변수로서 적절한 대기시간을 결정하는 것은 매우 중요하다.

본 논문에서는 MMOFPS게임과 같이 다수의 참여자가 동기화가 필요한 상호작용을 하는 클라이언트/서버 구조의 대용량 게임에서 향상된 응답성을 제공하면서 공정성의 저하를 최소화 할 수 있는 동기화 방법을 제안하고, 이 방법의 효용성을 실험을 통해 검증하였다. 게임에 참여하는 모든 참여자를 고려하여 대기시간을 결정하는 것은 실제로 상호작용을 하지 않는 참여자들까지 고려되어 동기화 성능을 떨어뜨릴 수 있다. 그러므로 실제로 상호작용을 하는 참여자만을 고려하여 대기시간을 결정해야 하는데, 서버에서 대기시간을 결정할 때 실제로 상호작용을 하는 참여자를 결정하는 것은 불가능하다. 따라서 본 논문에서 제안된 방법에서는 참여자간의 상호작용이 일어날 확률을 이용하여 대기시간을 결정하여 공정성을 유지하면서 응답성을 향상시켰다.

이후 본 논문은 다음과 같이 기술한다. 2장에서는 기존 동기화 방법의 장단점을 기술하며 3장에서는 공정성과 응답성을 정의하고 4장에서는 본 논문에서 제안하고자 하는 방법에 대해 기술한다. 5장에서 제안된 방법의 성능평가를 위해 가상환경을 구성하여 실험한 방법과 실험결과를 기술하고 6장에서 결론으로 본 논문을 맺는다.

## 2. 관련 연구

본 장에서는 기존에 연구되었던 동기화 방법에 대해 설명하고 장단점을 기술한다. 버킷 동기화[1, 2] 방법은 일정한 시간구간 내에서 발생한 메시지를 받은 즉시 처리하지 않고 버킷에 저장하여 일정시간(play-out time)이 지난 후에 모두 동시에 처리하는 방법이다. 미리 설정되는 일정시간보다 네트워크지연시간이 작을 때에는 불필요하게 기다리게 되어 응답성이 떨어진다. 네트워크지연시간이 커서 일정시간 내에 도착하지 못할 수 있는데 이를 해결하기 위해 늦게 도착한 메시지를 저장하여 참여자간에 불일치하는 화면을 보정하는 방법도 제시되었다[3, 4]. 하지만 보정을 하기 위해 많은 계산 량과 메모리 공간을 필요로 한다.

[5]에서는 네트워크지연시간을 예측하여 대기시간을 결정하는 동기화 방법들을 제안하였다. 첫 번째는 예측된 네트워크지연시간 중에서 최댓값을 대기시간으로 결정하는 방법이다. 이 방법은 동기화를 비교적 정확하게 수행할 수 있지만 네트워크지연시간이 큰 일부 참여자에 의해서 응답성이 크게 저하될 수 있다. 두 번째는 최근에 메시지를 보냈을 만한 참여자들의 네트워크지연시간 중에서 최댓값을 대기시간으로 결정하는 방법이다. 이 방법은 최근에 메시지를 보냈을 참여자들을 정확히 예측할수록 향상된 동기화 성능을 제공한다. 세 번째는 미리 정의된 가중치를 적용한 네트워크지연시간 중에서 최댓값을 대기시간으로 결정하는 방법이다. 이 방법은 미리 정의된 가중치를 적용하기 때문에 상황에 맞지 않는 가중치가 적용되었을 때에는 동기화 성능이 저하될 수 있다.

온라인게임에서 참여자의 관심영역과 참여자가 특정 아이템을 획득할 확률을 이용하여 대기시간을 결정하는 동기화 방법도 제안되었다[6]. 둘 이상인 참여자의 관심영역에 동일한 아이템이 존재 할 때 아이템과 참여자간의 거리에 반비례하는 확률을 이용하여 대기시간을 결정한다. 이 방법은 확률에 따라 대기시간을 감소시키기 때문에 향상된 응답성을 제공할 수 있지만 아이템이 주어진 상황에서만 사용될 수 있다.

다른 방법으로 현재 네트워크의 상태를 모델링하여 상태에 따라 동적으로 대기시간을 조절하는 방법도 제안되었다 [7]. 이 방법은 네트워크지연시간의 변화에 따라 적절한 대기시간을 결정하며 고정된 대기시간을 결정하는 방법보다

향상된 동기화 성능을 제공하지만 네트워크지연시간이 급격히 변하는 순간에는 동기화 성능이 매우 낮아지는 문제점이 있다.

[6]은 공정한 아이템 획득을 위해 아이템 획득 확률을 고려한 대기시간 결정 방법을 사용하였지만 본 논문에서는 관심영역 안에서 참여자 간의 상호작용 확률을 정의하고 관심영역에서 모든 참여자의 상호작용 확률을 고려하여 대기시간을 결정함으로써 참여자 간의 동기화 성능을 향상시킨다.

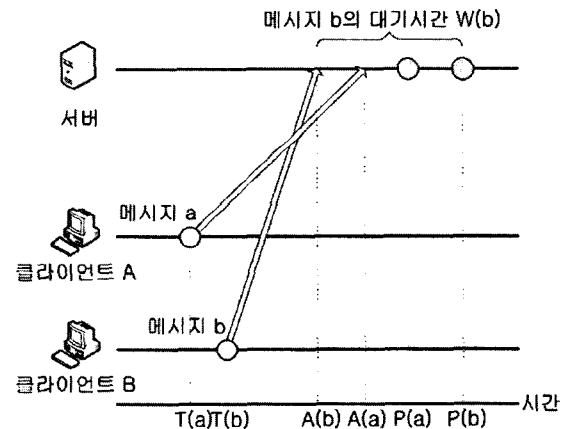
### 3. 공정성과 응답성

공정성은 동기화 성능을 평가하기 위한 측정 방법 중의 하나이다. 공정함이란 참여자들이 보낸 이벤트메시지가 실제 일어난 순서대로 처리됨을 의미한다. 본 논문에서는 클라이언트가 메시지 a를 보낸 시간을  $T(a)$ 라고 하고  $A(a)$ 를 메시지 a가 서버에 도착한 시간,  $P(a)$ 를 메시지 a가 서버에서 처리된 시간이라 하면, 메시지 a와 b에 대하여  $T(a) < T(b)$ 일 때  $P(a) \leq P(b)$ 를 만족하는 경우에 메시지 a와 b가 공정하게 처리되었다고 정의한다. 또한 불공정이란 메시지 a가 메시지 b보다 먼저 보내졌음에도 불구하고 메시지 a가 서버에 도착하기 전에 메시지 b가 서버에서 처리된 것이라 정의했다. 즉,  $T(a) < T(b)$  인 메시지 a, b에 대하여  $A(a) > P(b)$  인 경우에 메시지 b에 대하여 메시지 a가 불공정하게 처리되었다고 정의했다.

메시지가 서버에 도착하는 즉시 그 메시지를 처리하는 경우, 서로 다른 환경에 있는 참여자들의 메시지는 불공정하게 처리될 수 있다. 예를 들어 [그림 1]에서는 클라이언트 A가 클라이언트 B보다 먼저 서버로 메시지를 보냈으나 네트워크지연시간(network delay) 때문에 메시지가 일어난 순서대로 서버에 도착하지 못했다. 메시지가 도착 즉시 바로 처리가 된다면 클라이언트 B가 보낸 메시지 b의 처리가 클라이언트 A가 보낸 메시지 a의 처리보다 먼저 일어나게 되어 불공정하다.

참여자 환경의 차이 때문에 생기는 불공정함을 해결하는 대표적인 방법으로는 서버에 도착한 메시지를 즉시 처리하지 않고 일정한 대기시간(waiting period)이 지난 후에 처리하는 방법이 있다[그림 1]. 이 방법은 대기시간 동안 기다림

으로써 서로 다른 환경의 차이를 극복할 수 있는 기회를 제공한다. [그림 1]에서 클라이언트 B가 보낸 메시지 b는 서버에 도착 즉시 처리되지 않고 대기시간  $W(b)$ 를 기다린 후에  $A(b) + W(b) = P(b)$  시각에 처리 된다. 기다리는 동안에 클라이언트 A가 보낸 메시지 a가 서버에 도착하고, 서버는 메시지 a를 처리할 때 일어난 시간정보를 확인하여서  $T(a) < T(b)$ 이므로 메시지 b보다 먼저 처리한다. 대기시간  $W(b)$ 가 길어질수록 응답성은 저하되지만 공정성은 높아진다. 대기시간은 공정성과 응답성의 균형을 조절 할 수 있는 중요한 변수로서 양질의 가상환경을 구현하기 위해서 적절한 대기시간을 결정하는 것은 매우 중요한 문제이다.



[그림 1] 대기시간을 이용한 방법

### 4. 확률기반 대기시간 결정방법

공정한 가상환경에서는 환경의 차이에 관계없이 모든 참여자들이 같은 화면을 볼 수 있도록 해야 하며 그들의 액션 및 반응이 정확한 순서대로 처리되어야 한다. 클라이언트/서버 구조의 가상환경에서 메시지는 클라이언트에서 서버로 보내는 이벤트메시지와 서버에서 클라이언트로 보내는 갱신메시지가 있다. 가상환경에서의 동기화는 이벤트메시지 동기화와 갱신메시지 동기화를 포함한다. 이벤트메시지 동기화를 위해서는 서버에 도착한 이벤트메시지를 실제 일어난 순서대로 처리해야 하며, 갱신메시지 동기화를 위해서

는 도착한 갱신메시지를 모든 클라이언트가 동시에 처리해야 한다. 이벤트메시지 동기화와 갱신메시지 동기화는 동기화를 수행하는 주체만 다르므로 동일한 동기화 방법을 사용할 수 있다. 본 논문에서는 이벤트메시지를 중심으로 동기화 방법을 기술한다.

게임에 참여하는 모든 참여자를 고려하여 대기시간을 결정하는 것은 실제로 상호작용을 하지 않는 참여자들까지 고려되어 동기화 성능이 저하될 수 있으며 MMOFPS게임과 같이 참여자의 수가 많아진다면 동기화 성능 저하의 정도가 커질 수 있다. 그러므로 실제로 상호작용을 하는 참여자만을 고려하여 대기시간을 결정해야 한다. 하지만 서버에서 대기시간을 결정할 때 실제로 상호작용을 하는 참여자를 선별하는 것은 불가능하다. MMOFPS게임에서 참여자는 제한된 시야를 가지고 있기 때문에 타 참여자와 상호작용을 할 수 있는 범위가 제한되어 있으며 참여자의 시야에 있는 타 참여자들은 매우 빈번하게 바뀔 수 있다. 본 연구에서는 참여자의 제한된 시야를 관심영역의 범위로 정하여 관심영역 내의 참여자만을 고려하여 대기시간을 결정하였다.

본 연구에서는 관심영역 내에서 참여자간의 상호작용이 빈번하게 일어나고 참여자간의 거리가 가까울수록 상대방에게 빠르게 반응하는 일인칭 액션 게임의 특징을 이용하여 참여자간의 상호작용 확률을 고려한 대기시간 결정 방법을 제안한다. 이 방법은 참여자간의 상호작용 확률을 이용한 예측된 네트워크지연시간의 평균값을 대기시간으로 결정하는 방법이다. 제안된 방법에서 서버는 도착한 메시지의 대기시간을 결정하기 위해 메시지를 보낸 참여자의 관심영역에 있는 다른 참여자를 선별하고 메시지를 보낸 참여자와 선별된 참여자와의 상호작용 확률을 구한다. 그리고 선별된 모든 참여자의 상호작용 확률 합에 대한 선별된 각 참여자의 상호작용 확률의 비율을 구한다. 서버는 대기시간을 선별된 참여자들의 예측된 네트워크지연시간에 상호작용 확률의 비율을 곱하여 모두 합한 값으로 결정한다. 이것은 상호작용 확률을 이용한 예측된 네트워크 지연시간의 평균값을 구한 것과 같으며 [식 1]과 같이 기술 할 수 있다. [식 1]에서 상호작용 확률의 비율( $S_{ij}$ )에 따라서 대기시간에 적용되는 예측된 네트워크지연시간( $D_j$ )의 크기가 감소하기 때문에 향상된 응답성을 제공할 수 있으며 상호작용 확률을 적절히 고려하여 공정성을 유지할 수 있다.

$$S_{ij} = \frac{P_{ij}}{\sum_{k \in AOI(i)} P_{ik}}$$

$$W_i = \sum_{j \in AOI(i)} (D_j \times S_{ij}) \quad [식 1]$$

$D_j$ : 참여자 j의 예측된 네트워크지연시간  
 $P_{ij}$ : 참여자 i와 j사이의 상호작용 확률  
 $AOI(i)$ : 참여자 i의 관심영역에 있는 다른 참여자의 집합

참여자간의 상호작용 확률은 두 참여자가 상호작용을 할 확률을 정의한 것으로 실제 플레이어들의 행동 패턴을 분석하여 모델링해야 한다. 본 논문에서는 일인칭 액션 게임을 고려하여 참여자간의 상호작용 확률( $P_{ij}$ )은 두 참여자(i, j) 사이의 가상환경에서의 거리가 작을수록 크다고 가정하였으며, 참여자간의 상호작용이 일어날 확률을 표현하는 확률 함수를 [식 2]와 같이 정의하였다. [그림 2]는 [식 2]를 그림으로 표현한 것이다.

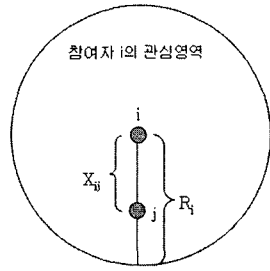
$$P_{ij} = 1 - \left( \frac{X_{ij}}{R_i} \right)^n$$

[식 2]

$R_i$ : 원으로 표현된 참여자 i의 관심영역의 반지름 길이  
 $X_{ij}$ : 참여자 i와 j사이의 거리

## 5. 실험 방법 및 결과

본 연구에서는 관심영역에 속하는 참여자들의 상호작용 확률을 이용한 예측된 네트워크지연시간의 평균값을 대기시간으로 결정하여 동기화를 실현했다. 제안된 방법을 검증하기 위해 제안된 상호작용 확률 모델을 이용하여 가상환경을 구축하고 동일한 모델로 실험하여 예측된 네트워크지연시간의 최댓값을 대기시간으로 결정하는 방법[5]과 비교하였다.



[그림 2] 상호작용 확률의 설명

본 실험에서는 100 x 100 크기의 지형에 2500명의 참여자가 분포되어 있는 가상환경을 구축하고 이벤트메시지를 생성한 후, 시뮬레이션을 통해 동기화 성능 및 응답성을 측정했다. 동기화 성능으로 모든 이벤트메시지에 대하여 불공정하게 처리된 이벤트메시지의 비율을 사용하였으며 응답성은 서버에서 결정할 수 있는 변수인 대기시간의 평균값으로 표시했다.

구축된 가상환경에서 각 참여자는 서로 다른 속도와 방향(8방향)을 가지고 이동한다. 각 참여자의 네트워크지연시간은 지수분포(exponential distribution)를 갖는 것으로 가정하였으며[5] 각 참여자의 평균(mean)은 난수 생성을 통해 결정하였다. 시뮬레이션은 가상의 이벤트메시지를 이용하여 약 80000ms동안 진행하였다.



[그림 3] 제안된 방법과 최대값 방법의 비교

[그림 3]은 제안된 방법과 네트워크지연시간의 최대값을 대기시간으로 결정하는 방법(최대값 방법)에 더해지는 상수 값을 조절하여 비교한 그림이다. 본 실험에서는 반지름의 크기가 5인 원을 각 참여자의 관심영역으로 사용하였으며 단위시간당 약 28.47명의 다른 참여자가 관심영역 내에 존재하였다. [그림 3]에서 평균 대기시간이 약 20ms 일 때 제안된 방법의 불공정비율은 약 8.38%이고 최대값 방법의 불공정비율이 약 21.72%로 제안된 방법의 불공정비율이 최대값 방법의 불공정비율에 비해 약 56.86% 작으며, 평균 대기시간이 약 50ms 일 때 제안된 방법의 불공정비율이 최대값 방법의 불공정비율에 비해 약 60.69% 작다. 전체적으로 제안된 방법과 최대값 방법의 평균대기시간이 같을 때 제안된 방법의 전체불공정비율이 낮기 때문에 최대값 방법보다 제안된 방법이 향상된 동기화 방법이라고 할 수 있다. 또한 두 가지 방법을 실제로 사용할 때에 적절한 상수 값을 찾는 것은 어려우므로 제안된 방법은 최대값 방법보다 참여자들의 네트워크지연시간의 현재 상황에 맞게 응답성을 향상시키되 불공정비율의 증가를 작게 할 수 있는 방법이다.

## 6. 결론

본 논문에서는 MMOFPS게임의 특징을 고려하여 관심영역 안에 있는 참여자간의 상호작용 확률을 이용한 예측된 네트워크지연시간의 평균값을 대기시간으로 결정하여 메시지를 동기화하는 방법을 제안하였다. 대기시간을 결정하기 위하여 참여자간의 상호작용 확률을 고려함으로써 응답성을 향상시켰으며, 동기화 성능을 적절히 조절하였다. 제안된 방법을 검증하기 위해 가상환경에서 결정된 대기시간에 더해지는 상수 값을 조절하여 실험하였다. 실험에서는 대부분의 경우에 관심영역 안에 있는 참여자들의 네트워크지연시간의 최대값에서 상수 값을 뺀 값을 대기시간으로 결정하는 방법보다 본 논문에서 제안한 방법이 더 좋은 결과를 보여 네트워크로 연결된 가상환경의 일반적인 상호작용 상황에서 메시지를 동기화 하는데 효과적인 방법임을 입증하였다.

## 참고문헌

- [1] L.Gautier and C.Diot, "A Distributed Architecture for Multiplayer Interactive Applications on the Internet," IEEE Network, Vol. 13, No.4, pp. 6-15, 1999.
- [2] L.Gautier and C.Diot, "End-to-End Transmission Control Mechanisms for Multiparty Interactive Applications on the Internet," In Proceedings of the Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies, Vol.3, pp.1470-1479, 1999.
- [3] T.K.Capin, J.Esmerado and D.Thalmann, "A Dead-Reckoning Technique for Streaming Virtual Human Animation," IEEE Transactions on Circuits and Systems for Video Technology, Vol 9, pp.411-414, 1999.
- [3] M.Mauve, "Consistency in Replicated Continuous Interactive Media," In Proceedings of ACM Conference on Computer Supported Cooperative Work, pp.181-190, 2000.
- [4] J.Vogel and M.Mauve, "Consistency Control for Distributed Interactive Media," In Proceedings of the Ninth ACM International Conference on Multimedia, pp.221-230, 2001.
- [5] Yow-Jian Lin, Katherine Guo and Sanjoy Paul, "SyncMS: Synchronized Messaging Service for Real-Time Multi-Player Distributed Games," IEEE ICNP, pp.155-164, 2002.
- [6] Kazutoshi FUJIKAWA, Atsushi ITOH, Shinji SHIMOJO, and Hideo MIYAHARA, "Synchronization Issues on Networked Virtual Environments of Large-Scale Multiparty Applications," 電子情報通信学会 技術研究報告 IA2002-32E, pp.73-79, 2002.
- [7] E. Hong, D. Lee, E. Park, and K. Kang, "An Efficient Synchronization Mechanism Adapting to Heterogeneous Transmission Delay in Networked Virtual Environments," In Proceedings of SPIE/ACM MMCN2003, pp. 24-33, 2003.



백 두 원

서울대학교 수학과 졸업 (BS)  
 University of Minnesota 전산학과 졸업 (MS, Ph. D.)  
 AT&T Bell Lab., Member of Technical Staff  
 Cadence Design System, Member of Consulting Staff  
 현 숭실대학교 미디어학부 부교수  
 관심분야 : 디지털방송, 컴퓨터 그래픽스, 알고리즘



윤 청 하

숭실대학교 미디어학부 학사 졸업  
 숭실대학교 미디어학부 석사 졸업  
 현 제너시스시스템즈 기술연구소 연구원  
 관심분야 : 네트워크, 디지털방송

논문투고일 - 2005년 11월 2일  
 심사완료일 - 2005년 11월 28일