

# 네트워크 가상환경에서의 선별적 메시지 동기화 기법

윤청하<sup>†</sup>, 백두원<sup>††</sup>

## 요 약

가상환경에 참여하는 참여자들의 하드웨어와 네트워크 성능은 매우 다를 수 있으므로 다양한 환경 하에 있는 참여자들에게 동일한 대우를 하는 공정성과 적절한 응답성의 보장과 이를 위한 동기화는 반드시 해결되어야 할 과제이다. 동기화를 위하여 메시지를 받은 즉시 처리하지 않고 일정한 대기시간이 지난 후 처리하는 방법을 사용할 수 있다. 대기시간이 길어질수록 공정성은 좋아지나 응답성이 떨어지므로 적절한 대기시간을 결정하는 것은 쉬운 문제가 아니다. 본 논문에서는 참여자들의 네트워크 지연시간 중 일정 비율을 제외한 나머지만을 고려하여 대기시간을 결정하는 방법을 제안하고 이 방법의 효용성을 검증했다. 이를 위해 가상환경을 구축하고 메시지를 생성하여 모의실험을 하였다. 실험에서는 본 논문에서 제안한 방법이 대부분의 경우에 좋은 결과를 보여 네트워크로 연결된 가상환경에서 메시지를 동기화 하는데 효과적인 방법임을 입증하였다.

## A Selective Message Synchronization Mechanism for Networked Virtual Environment

Chung-Ha Yun<sup>†</sup>, Doowon Paik<sup>††</sup>

## ABSTRACT

Participants of a NVE(networked virtual environment) may have different hardwares and network environments, and to guarantee the fairness, 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 only the participants with small network delays and perform experiments for the performance evaluation. Experiments show that the proposed method leads to significant improvement of responsiveness, while retaining fairness.

**Key words:** Networked Virtual Environment(네트워크 가상환경), Synchronization(동기화), Waiting Period(대기시간), Area of Interest(관심영역)

## 1. 서 론

인터넷이 확산되고 그 성능이 향상되면서 다수의 참여자가 네트워크 가상환경(NVE, networked virtual environment)을 공유하는 것이 가능해졌다. 일례가 온라인게임으로 흠어져있는 다수의 참여자들이

하나의 가상환경을 공유하며 함께 게임을 즐길 수 있다. 네트워크 가상환경의 변화정보는 모든 참여자 간에 즉시 공유되어야 하며 참여자의 수가 증가하여도 환경의 일관성이 유지되어야 한다. 하지만, 이 가상환경에 참여하는 참여자들의 네트워크 환경과 하드웨어 성능은 서로 매우 다를 수 있으므로 다양한 환경

※ 교신저자(Corresponding Author): 윤청하, 주소: 서울시 동작구 상도5동(156-743), 전화: 02)820-0114, FAX: 02)822-3622, E-mail: guma2@hanmail.net  
접수일: 2005년 7월 21일, 완료일: 2005년 10월 13일

<sup>†</sup> 준회원, 숭실대학교 미디어학부 석사과정

<sup>††</sup> 정회원, 숭실대학교 미디어학부 부교수

(E-mail: dpaik@computing.ssu.ac.kr)

※ 본 연구는 숭실대학교 교내연구비 지원으로 이루어졌음.

하에 있는 참여자들에게 동일한 대우를 하는 공정성(fairness)과 적절한 응답성(responsiveness)은 보장되어야 하고 이를 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 조절할 수 있는 중요한 변수로서 적절한 대기시간을 결정하는 것은 매우 중요하다.

본 논문에서는 클라이언트/서버 구조의 가상환경에서 지리적인 환경에 포함 되었을 때, 향상된 응답성을 제공하면서 공정성의 저하를 최소화 할 수 있는 동기화 방법을 제안하고 이 방법의 효용성을 실험을 통해 검증하였다. 지리적인 환경을 기반으로 하는 가상환경에서 거리상으로 멀리 떨어져 있거나 같은 화면에 보이지 않는 참여자들 사이에는 상호작용이 일어날 확률이 적고 따라서 동기화의 필요성이 적어진다. 또한 모든 참여자의 공정성을 고려하면 네트워크 지연시간이 큰 소수의 참여자에 의해 전체 참여자의 응답성이 크게 좌우되는 문제점이 발생하기 때문에 상대적으로 네트워크지연시간이 큰 소수 참여자의 공정성을 희생함으로써 전체 참여자의 응답성을 향상시켜야 할 필요성이 있다. 제안된 방법은 동기화가 필요한 참여자만을 고려하고 네트워크지연시간에 따른 선별을 통해 네트워크지연시간이 큰 참여자를 배제하여 대기시간을 결정함으로써 공정성을 최대한 유지하면서 응답성을 향상시킨다. 본 논문에서는 동기화에 실패하는 참여자에 대한 처리방법에 대해서는 다루지 않는다.

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

## 2. 관련연구

본 장에서는 기존에 연구되었던 동기화 방법에 대

해 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 일정시간은 동기화를 충분히 수행할 수 있을 만큼 크게 결정하기 때문에, 네트워크지연시간이 작을 때에는 불필요하게 기다리게 되어 응답성이 떨어진다. 또한 네트워크지연시간이 커서 일정시간 내에 도착하지 못한 메시지는 처리하지 않기 때문에 동기화 성능이 저하 될 수 있다. 버킷 동기화에서 동기화 성능이 저하되는 문제점을 해결하기 위해 일정시간 내에 도착하지 못한 메시지를 저장하여 참여자간에 불일치하는 화면을 보정하기 위해 사용하는 방법도 제시되었다[3,4]. 하지만 이 방법은 보정을 하기 위해 많은 계산 량과 메모리 공간을 필요로 한다.

[5]에서는 네트워크지연시간을 예측하여 대기시간을 결정하는 세 가지 동기화 방법을 제안하였다. 첫 번째 방법은 서버와 모든 클라이언트 사이의 예측된 네트워크지연시간 중에서 최댓값을 대기시간으로 결정하는 방법이다. 이 방법은 동기화를 비교적 정확하게 수행할 수 있지만 상대적으로 네트워크지연시간이 작은 클라이언트는 불필요하게 기다리게 되어 응답성이 저하되는 문제가 있다. 두 번째는 최근에 메시지를 보냈을 만한 클라이언트들을 선별하여 그들의 네트워크지연시간 중에서 최댓값을 대기시간으로 결정하는 방법이다. 이 방법은 최근에 메시지를 보냈을 클라이언트들을 정확히 예측할수록 향상된 동기화 성능을 제공한다. 세 번째는 미리 정의된 가중치를 적용한 네트워크지연시간 중에서 최댓값을 대기시간으로 결정하는 방법이다. 가중치에 따라 대기시간이 줄어들기 때문에 응답성을 높일 수 있으며 가중치를 조절하여 동기화 성능과 응답성의 정도를 조절할 수 있다. 단, 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. 관심영역을 이용하는 방법은 모든 참여자 중에서 동기화가 필요한 참여자만을 선별하여 대기 시간 결정에 이용하는 방법이다. 가상환경에서 참여자들의 상호작용은 참여자들이 지리적으로 근접했을 때 주로 발생하므로 참여자의 주변지역인 관심영역 안에 있는 참여자만을 선별하여 고려대상 참여자의 수를 줄일 수 있다.

다른 방법으로 현재 네트워크의 상태를 모델링하여 상태에 따라 동적으로 대기시간을 조절하는 방법도 제안되었다[7]. 이 방법은 네트워크 지연시간의 변화에 따라 적절한 대기시간을 결정하며 고정된 대기시간을 결정하는 방법보다 향상된 동기화 성능을 제공하지만 네트워크 지연시간이 급격히 변하는 순간에는 동기화 성능이 매우 낮아지는 문제점이 있다.

본 논문에서는 [5]에서 제안된 네트워크 지연시간의 최댓값을 이용한 방법의 단점을 해결하기 위해 네트워크 지연시간을 선별적으로 고려하여 대기시간을 결정함으로써 응답성을 향상시키는 방법을 제안한다. 이 방법은 특정상황에서 사용한 관심영역[6]을 일반상황에서 사용하여 동기화가 필요한 부분만을 고려함으로써 동기화 성능을 향상시킨다. 또한 일시적으로 네트워크 지연시간이 커진 참여자는 응답성을 저하시킬 뿐 아니라 공정성도 보장하기 힘들기 때문에 네트워크 지연시간을 선별적으로 고려한다면 [7]에서의 문제점도 해결 가능하다.

### 3. 메시지 동기화 기법

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 위해서는 도착한 갱신메시지를 모든 클라이언트가 동시에 처리해야 한다. 이벤트메시지 동기화

와 갱신메시지 동기화는 동기화를 수행하는 주체만 다르므로 동일한 동기화 방법을 사용할 수 있다. 본 논문에서는 이벤트메시지를 중심으로 동기화 방법을 기술한다. 또한 본 논문에서 참여자와 클라이언트는 같은 의미이며 참여자와 가상환경, 클라이언트와 서버는 동일한 관계이다.

#### 3.1 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 네트워크 지연시간(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 기다림으로써 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 처리된다. 기다리는 동안에 클라이언트 A가 보낸 메시지 a가 서버에

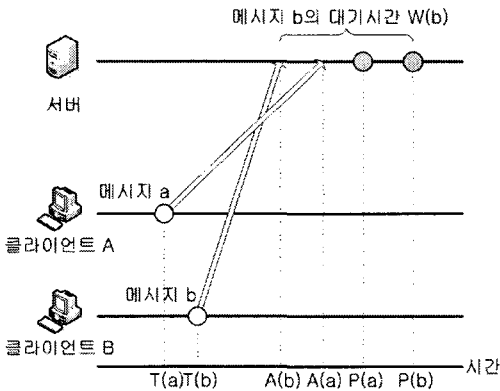


그림 1. 대기시간을 이용한 방법

도착하고, 서버는 메시지 a를 처리할 때 일어난 시간 정보를 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.

### 3.2 네트워크지연시간을 이용한 선별적인 대기시간 결정 방법

관심영역 안에 있는 참여자들의 네트워크지연시간 중에서 최댓값을 대기시간으로 결정하는 방법은 간단하면서도 높은 동기화 성능을 제공한다[3]. 하지만, 이 방법은 관심영역 안에 있는 참여자들의 정확한 동기화를 보장할 수 없으며 상대적으로 네트워크지연시간이 큰 소수의 참여자가 동기화되기 위해서는 상대적으로 네트워크지연시간이 작은 다수의 참여자의 응답성이 희생되어야 하며 네트워크지연시간이 큰 참여자에 의해 가상환경의 응답성이 좌우 될 수 있는 문제점을 가진다. 이러한 단점을 극복하기 위해 관심영역 안에 있는 참여자들의 네트워크 지연시간의 최댓값에서 일정 값을 뺀 결과를 대기시간으로 결정할 수 있지만 빼는 값의 크기가 커질수록 향상되는 응답성에 비해 공정성이 크게 저하 될 수 있다.

본 연구에서는 이와 같은 문제를 해결하기 위해 네트워크지연시간에 따라 선별하여 대기시간 결정 방법을 이용했다. 이 방법은 참여자들의 네트워크지연시간에서 상위 X%를 제외한 값들 중 최댓값을 대기시간으로 결정하는 방법이다. 예를 들어 X가 20

인 경우 네트워크 지연시간들의 집합으로 {10, 3, 5, 9, 1, 3, 5, 49, 23, 29} 이 주어졌을 때, 상위 20%에 속하는 {49, 29}를 제외한 값들 중 최댓값인 23을 대기시간으로 결정한다. 이 방법에서는 X의 값이 작을 수록 대기시간이 네트워크지연시간 집합에서의 최댓값에 가까워지기 때문에 공정성은 향상되지만 응답성이 저하되며, X의 값이 커질수록 대기시간이 작게 결정되기 때문에 응답성은 향상되지만 공정성은 저하된다.

상위 X%의 네트워크지연시간을 제외하는 방법에서는 관심영역에 있는 참여자들의 네트워크지연시간의 분포에 따라 상대적으로 네트워크지연시간이 큰 참여자가 제외된다. 네트워크 가상환경의 관리자는 가상환경의 질을 높이기 위해 요구사항보다 큰 평균 네트워크지연시간을 가지는 참여자의 접속을 해제하는 방법을 사용할 수 있다. 하지만 네트워크 가상환경의 요구사항을 만족하는 참여자의 네트워크지연시간은 대부분 평균에 가깝지만 평균에 비해 매우 큰 값이 될 수 있으며, 네트워크 환경에 문제가 있을 경우 지속적으로 큰 값이 될 수 있다. 제안된 방법은 위와 같은 경우에 네트워크지연시간이 큰 참여자를 적절하게 제외함으로써 전체 사용자의 응답성을 보장하고 공정성의 저하를 최소화 할 수 있다. 단, 참여자의 네트워크 환경에 문제가 오랫동안 지속되어 공정성을 보장받기 힘들게 되면 참여자에게 통지 메시지를 보내거나 접속을 해제하는 방법을 사용할 수 있다. 반면에 제외되지 않는 참여자는 대부분 동기화 될 수 있기 때문에 참여자간의 네트워크지연시간 차이가 커질수록 향상되는 응답성은 커지지만 공정성이 저하되는 정도는 작아진다. 네트워크지연시간이 상대적으로 큰 참여자들이 많을 경우 응답성은 저하될 수 있지만 공정성은 유지할 수 있으며, 네트워크지연시간이 상대적으로 작은 참여자들이 많을 경우 응답성과 공정성을 동시에 향상시킬 수 있다. 또한 가상환경의 응답성은 예측된 네트워크지연시간의 오차에 영향을 덜 받는다.

### 4. 실험방법 및 결과

본 연구에서는 관심영역에 속하는 참여자들의 네트워크지연시간에서 상위 일부를 제외한 값들 중 최댓값을 대기시간으로 결정하여 동기화를 실현했다.

제안된 방법을 검증하기 위해 가상환경에서 관심영역의 크기와 제외될 상위비율의 값을 조절하여 실험하였으며 관심영역에 속하는 참여자들의 네트워크 지연시간에서의 최댓값과 최솟값에서 일정 값을 뺀 값을 대기시간으로 결정한 방법과 비교하였다.

본 실험에서는 100 x 100 크기의 지형에 2500명의 참여자가 분포되어 있는 가상환경을 구축하고 이벤트 메시지를 생성한 후, 시뮬레이션을 통해 동기화 성능 및 응답성을 측정했다. 동기화 성능으로 모든 이벤트 메시지에 대하여 불공정하게 처리된 이벤트 메시지의 비율을 사용했다. 가상환경에서의 응답성으로 클라이언트와 서버 사이에서 주고받는 네트워크지연시간과 서버에서 처리되는 시간을 합한 왕복지연시간(Round-Trip Delay)을 이용할 수 있다. 네트워크지연시간은 네트워크의 상황에 따라 매우 다를 수 있고, 서버에서 CPU에 의해 처리되는 시간은 대기시간에 비하면 작아 무시할 수 있으므로, 본 실험에서는 응답성을 모든 메시지의 평균대기시간으로 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. 가상의 이벤트 메시지는 참여자간의 거리에 반비례 하는 확률로 생성되며 한 번의 주기에서 일어나는 모든 메시지는 임의의 순서로 분포된다. 시뮬레이션은 가상의 이벤트 메시지를 이용하여 약 80000ms동안 진행하였다.

(그림 2)]는 관심영역의 크기에 따른 평균 대기시간을 측정 한 결과이다. 그림에서 X(%)는 제외되는 상위비율을 의미한다. X값이 10에서 50까지의 경우에는, 관심영역의 크기가 증가함에 따라 평균 대기시간이 급격하게 증가하다가 그 증가량이 급격히 감소하여 일정한 평균 대기시간 값을 갖는다. 이와 같은 결과는 처음에는 관심영역이 커짐에 따라 관심영역 안에 있는 네트워크지연시간이 큰 참여자의 수가 증가하여 평균 대기시간이 증가하다가 관심영역의 크기가 어느 정도 커지면 관심영역 안에 있는 참여자들의 최대, 최소 네트워크지연시간 사이에 분포하는 값의 분포가 균일해지게 되어 최대 평균 시간으로 일정해지기 때문이다. X값이 0인 경우에는 관심영역의 크기가 증가함에 따라 그 증가량은 감소하지만 평균 대기시간이 계속적으로 증가하는데 이는 관심영역

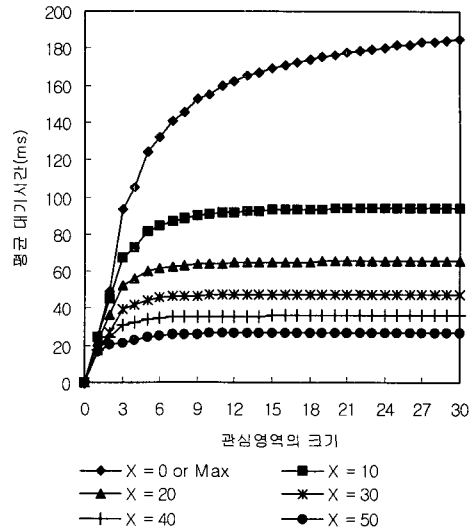


그림 2. 관심영역의 크기에 따른 평균 대기시간 측정

에 있는 네트워크지연시간이 매우 큰 소수의 참여자들 때문이다.

(그림 3)은 관심영역의 크기에 따른 불공정 비율을 측정 한 결과이다. 관심영역의 크기가 증가함에 따라 불공정 비율이 크게 감소하다가 관심영역의 크기가 어느 정도(10 정도)에 도달하면 일정한 불공정 비율 값을 갖는다. X값이 0인 경우에 관심영역의 크기가 증가함에 따라 불공정 비율이 0%에 가까워지는 것은 전체 메시지가 공정하게 처리될 만큼 평균 대기

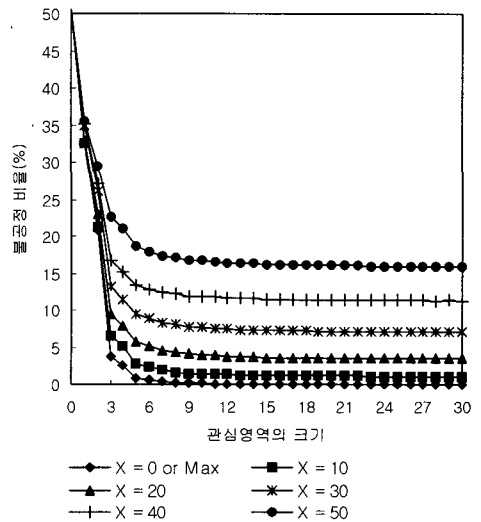


그림 3. 관심영역의 크기에 따른 전체 불공정 비율의 측정

시간의 값이 커지기 때문이다. X값이 10에서 50까지의 경우에는 관심영역의 크기가 큰 경우에는 평균 대기시간이 일정하게 되어 불공정 비율도 변하지 않는다.

(그림 4)는 관심영역의 크기에 따른 평균 대기시간과 불공정 비율을 나타낸 그림이다. 평균 대기시간이 같을 때 X값이 10에서 50까지의 경우가 0인 경우에 비해 전체 불공정 비율이 상대적으로 작다. 예를 들어 평균 대기시간이 50ms에서 X값이 0인 경우는 불공정 비율이 20.7%인 반면에 X값이 40인 경우는 불공정 비율이 10.0%로서, X값이 40인 경우의 불공정 비율이 X값이 0인 경우의 불공정 비율보다 약 51.7% 작으며, 평균 대기시간이 93ms 에서는 X값이 0인 경우에 비해 X값이 10인 경우에 불공정 비율이 약 75% 작다. 따라서 참여자들의 네트워크지연시간 중 일부 큰 값을 제외하고 대기시간을 결정한 방법은 관심영역 안에 있는 참여자들의 네트워크지연시간 중에서 최댓값을 대기시간으로 결정한 방법(Max)보다 향상된 동기화 방법이라고 할 수 있다.

(그림 4)를 보면 평균 대기시간에 따라 불공정 비율이 최소가 되는 X값이 다른 것을 알 수 있다. 예를 들어 평균 대기시간이 30ms 되는 곳에서는 X값이 50인 경우에 불공정 비율이 최소가 되며, 95ms 되는 곳에서는 X값이 10인 경우에 불공정 비율이 최소가 된다. 불공정 비율이 최소가 되는 지점에 많은 점들이 모여 있는 것은 관심영역의 크기가 증가하더라도 평균 대기시간과 불공정 비율은 거의 변하지 않기 때문이다(그림 2, 3). 따라서 허용 가능한 최대 평균 대기시간이 주어졌을 때 불공정 비율이 최소가 되는 적절한 X값을 선택할 수 있다.

(그림 5)는 X값에 따른 평균 대기시간과 불공정 비율을 나타낸 그림이다. X값이 비교적 작을 때(0~15) 평균 대기시간이 감소하는 양이 큰 것에 비해 불공정 비율이 증가하는 양이 작은 것으로 보아 제안된 방법은 X값이 비교적 작을 때 동기화 성능이 좋은 것을 알 수 있다. 반면에 X값이 증가할 수록 기울기의 절댓값이 급격히 증가하는 것으로 보아 동기화 성능이 저하되는 것을 알 수 있다. 또한 (그림 5)는 네트워크지연시간 확률분포의 평균값 범위를 다르게 하여 비교한 그림이다. 확률분포의 평균값의 최댓값이 크고 그 범위가 클수록 동기화 성능의 향상이 두드러진다.

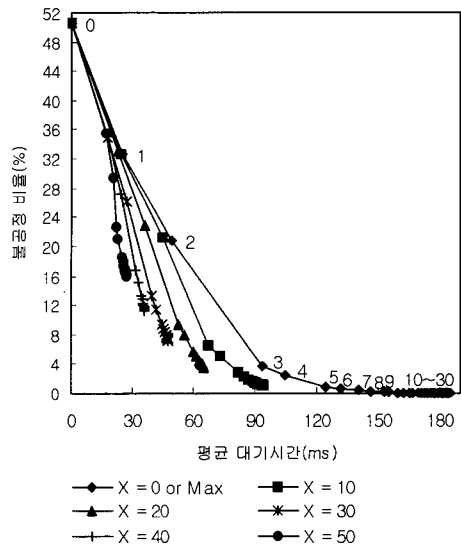


그림 4. 관심영역의 크기에 따른 평균 대기시간과 전체 불공정 비율

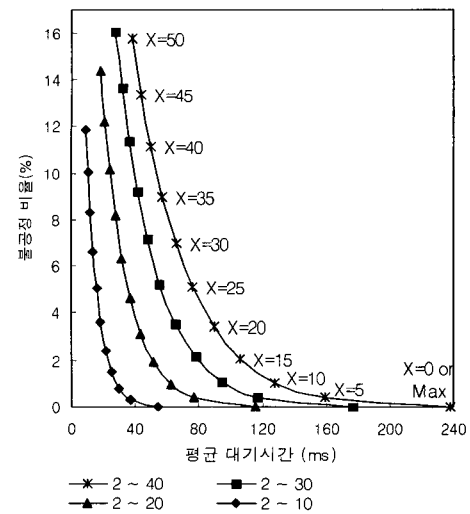


그림 5. X값에 따른 평균 대기시간과 전체 불공정 비율

(그림 6)은 제안된 방법과 관심영역에 속하는 참여자들의 네트워크지연시간에서의 최댓값에서 일정값을 뺀 값(Max - a)을 대기시간으로 결정한 방법을 비교한 그림이다. 대부분의 경우에서 두 방법의 불공정비율이 같을 때 제안된 방법의 평균 대기시간이 더 작으므로 제안된 방법이 더 향상된 동기화 성능을 제공한다. 또한 제안된 방법은 네트워크지연시간의 확률분포를 정규분포(normal distribution)로 선택하였을 때에도 향상된 동기화 성능을 제공한다.

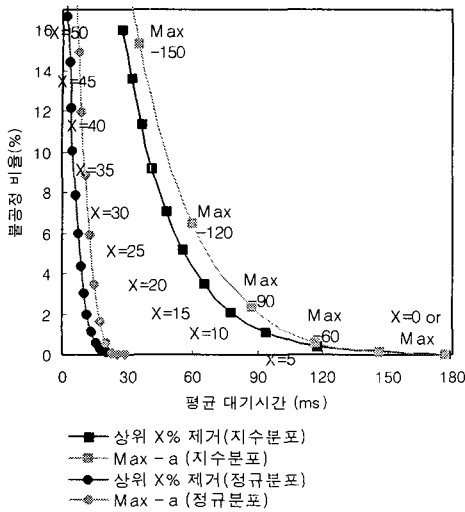


그림 6. 상위 X% 제거와 최댓값에서 일정 값(Max - a)을 뺀 방법의 비교

### 5. 결 론

본 논문에서는 관심영역에 속하는 참여자들의 네트워크 지연시간에서 상위 일부를 제외한 값들 중 최댓값을 대기시간으로 결정하여 메시지를 동기화하는 방법을 제안하였다. 대기시간을 결정하기 위하여 고려할 클라이언트와 네트워크 지연시간을 선별적으로 선택함으로써 응답성을 향상시켰으며, 동기화 성능을 적절히 조절하였다. 제안된 방법을 검증하기 위해 가상환경에서 관심영역의 크기와 제외될 상위비율(X%)의 값을 조절하여 실험하였다. 실험에서는 대부분의 경우에 관심영역 안에 있는 참여자들의 네트워크 지연시간의 최댓값에서 일정 값을 뺀 값을 대기시간으로 결정한 방법보다 본 논문에서 제안한 방법이 더 좋은 결과를 보여 네트워크로 연결된 가상환경의 일반적인 상호작용 상황에서 메시지를 동기화하는데 효과적인 방법임을 입증하였다.

### 참 고 문 헌

[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] 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, "Sync-MS: 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," *電子情報通信學會 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.



윤 청 하

2004년 숭실대학교 미디어학부  
 학사 졸업  
 2005년 숭실대학교 미디어학부  
 석사 과정  
 관심분야 : 네트워크, 디지털방송



백 두 원

1983년 서울대학교 수학과 졸업  
 (BS)  
 1991년 University of Minnesota 전  
 산학과 졸업 (MS, Ph. D.)  
 1992~1994년 AT&T Bell Lab.,  
 Member of Technical Staff  
 2001~2002년 Cadence Design  
 System, Member of Consulting Staff

현 숭실대학교 미디어학부 부교수  
 관심분야 : 디지털방송, 컴퓨터 그래픽스, 알고리즘