

# 네트워크 가상현실 응용의 설계 및 구현과 가변적인 네트워크 상태를 고려하는 동기화 기법

최 성, 유 준호  
남서울대학교 컴퓨터학과  
rjh8306@hotmail.com

## Design and Implementation of Networked Virtual Reality Applications and An Efficient Synchronization Mechanism Dynamically Adapting to the Network State

Choi Sung, You Jun-Ho  
Namseoul University Computer Major

### 요 약

네트워크 가상현실 시스템은 먼 거리에 떨어진 사용자들 사이에 일관성 있는 가상 세계를 제공하며, 군사, 오락, 건축 등 여러 부분에 응용되고 있다. 본 논문에서는 그 동안 고성능 그래픽 워크스테이션 환경에서 중심으로 연구되어 왔던 네트워크 가상현실 시스템을 가장 보편적인 플랫폼인 네트워크 환경에서 구현할 때 생기는 3차원 그래픽 처리 성능의 문제 등과 같은 이에 대한 해결책들을 제시하고 미로 환경과 지형 환경에 근거한 가상 세계를 응용 실험 대상으로 삼아 성능을 분석하였다. 실시간으로 상호 작용하는 어플리케이션은 성능을 유지하면서 모든 참가자들에게 동일한 뷰를 보여주는 것이 중요하다. 동일한 뷰를 제공하기 위해 참가자들이 전송하는 이벤트마다 재생 지연 시간을 설정하고, 수신한 이벤트에 대해서는 예정 재생 시각까지 버퍼에 저장하여 모든 참가자들이 동시에 이벤트를 실행하도록 하는 기법이 제안되어 왔다. 그러나 네트워크 트래픽이 동적으로 변하므로 네트워크 상에서 발생하는 전송 지연 시간도 동적으로 변화한다. 그러므로 고정된 재생 지연 시간을 사용할 경우, 네트워크 트래픽 감소에 따라 상호 작용 성능을 향상시킬 수 있는 기회를 상실하게 되고, 네트워크 트래픽 증가에 따라 이벤트 손실률이 크게 증가하게 되어 참가자들 간에 일치하지 않는 뷰를 초래하게 된다. 네트워크 트래픽이 적어서 전송 지연 시간이 짧을 경우 짧은 재생 지연 시간을 적용하여 상호 작용 성능을 높여주고, 트래픽이 많은 경우에는 재생 지연 시간을 늘림으로써 상호 작용 성능을 해치지 않는 범위에서 이벤트 손실률을 줄인다. 실험을 통하여 제안하는 기법의 성능을 평가하였으며 그 결과 본 동기화 기법이 고부하(heavily loaded) 네트워크 상태가 지속되는 동안 기존 기법에 비해 참가자들에게 일치된 뷰를 제공할 수 있고, 저부하(lightly loaded) 네트워크 상태에서는 상호 작용 성능을 향상시킬 수 있다는 것을 확인하였다.

키워드: 네트워크 가상현실, 동기화, 프레임 레이트 제어

### 1. 서론

컴퓨터 그래픽 시스템의 궁극적인 형태로서 가상 현실 시스템은 사용자가 테이블이나 등불과 같은 가

상 세계의 개체들 뿐 아니라 그 시스템에 연결된 다른 사용자와도 투명하게 상호작용할 수 있는 시스템을 목적으로 한다. 지역적 제한성을 극복한 복수 사용자의 지원은 네트워크를 기반으로 하며 여러 사용

자 사이의 공유 가상세계의 “일관성(consistency)”을 보장해주는 것을 목적으로 한다. 이 때 “일관성”이란 각 사용자가 인식하게 되는 가상세계가 충분히 동기화되어서, 단일하게 통합된 하나의 시스템을 사용하는 듯한 느낌을 제공해주는 것을 말한다.

네트워크 가상현실 시스템(또는 분산 가상현실 시스템, networked/distributed virtual reality system)은 잠재적인 응용 분야가 넓기 때문에 활발한 연구 대상이 되고 있으며, 특히 고성능 그래픽 워크스테이션에 기반하여 군사 시뮬레이션 등의 응용에서 유용성을 보여 주었고, 오락, 교육, 의학, 건축 등의 분야로 그 응용 범위를 넓혀 가고 있다. 그러나 일정 수준 이상의 3차원 그래픽 처리 능력과 네트워크 통신 처리능력이 네트워크 가상현실 시스템의 동작에 필요하기 때문에, 이 시스템들은 주로 고성능 그래픽 워크스테이션이나 전용 시뮬레이터 환경에 국한되어 구현되어 왔다. 본 논문에서는 이러한 네트워크 가상현실 시스템을 PC와 같은 상대적으로 낮은 성능을 가지는 플랫폼에서 구현할 때 부딪히는 몇 가지 문제점들 3차원 그래픽 처리상에서의 문제점과 네트워크 통신 처리상에서의 문제점과 그에 대한 해결책을 제시하며, 실험을 통해 이 해결책이 성능에 미치는 영향을 분석한다.

## 2. 네트워크 통신 처리

네트워크 가상현실 시스템을 위한 분산 아키텍처는 동료 대 동료 아키텍처와 클라이언트 서버 아키텍처로 나누어질 수 있다. 클라이언트 서버 아키텍처의 경우에 가상 세계에 대한 데이터베이스는 서버에 저장되어 있으며 모든 데이터베이스에 관한 처리는 서버에서 이루어진다. 따라서 사용자들은 자신의 정보가 갱신될 때마다 서버에 정보 갱신 패킷을 보내야 하며, 많은 사용자가 빈번한 패킷을 보내면 서버에서 처리해야 할 패킷 수가 너무 많아져 scalability에 문제를 보이게 된다. 반면, 동료 대 동료 아키텍처는 데이터베이스 자체가 클라이언트에 분산되어 있고 패킷 처리는 많은 시스템에 분산되기 때문에 보다 균형적인 부하 균형이 이루어지나, 패킷의 전송, 수신을 위한 절차가 복잡해지는 단점이 있다.

각 호스트 사이에서 교환되는 메시지들은 가상 세계 내의 객체들에 대한 상태 정보 값을 가지고 있으며, 각 사용자에게 대한 정보가 변할 때마다 패킷을

보내는 것이 사용자 수가 증가함에 따라 많은 수의 대역폭을 소모하게 됨을 의미한다. 이를 해결하기 위해 추측 항법 알고리즘을 이용하여 패킷의 수를 줄인다.

추측 항법 알고리즘은 각 사용자가 연결되어 있는 호스트는 로컬 사용자에게 대한 정보인 player list와 연결된 모든 사용자에게 대한 정보인 ghost list를 관리한다. ghost list에는 리모트 사용자 뿐 아니라 자신에 대한 ghost도 포함되어 있다. 각 사용자에게 대한 정보가 변할 때마다 패킷을 보내는 비효율성을 극복하기 위해 각 프레임마다 각 호스트는 자신의 현재 player 정보와 ghost 정보를 비교한 후, 그 차이값이 오차를 넘어설 때만 메시지를 보낸다. 리모트 사용자는 이쪽 사용자의 정보를 그에 해당되는 ghost 정보에 의해 추정한다. 일정한 오차를 넘어섰을 때 메시지를 보내기 때문에 이 추정된 값은 가상세계의 동기성을 크게 훼손시키지는 않는다.

추측 항법에는 여러 종류의 기법이 사용될 수 있는데 여기에서는 사용자의 가장 최근의 위치 이동점을 3개까지 저장해 두었다가 위치 변화의 각도에 따라, 2차 또는 3차의 곡선 맞추기 알고리즘을 적용하는 기법을 채택하였다

## 3. 가변적인 네트워크 상태를 고려하는 동기화 기법 구현

본 논문에서 제안하는 동기화 기법은 가변적인 네트워크 상태를 고려하기 위해 상호 작용 성능과 뷰 일관성의 관계 네트워크 트래픽 상태 측정 및 오버헤드 시각 동기화 측면을 고려한다. 네트워크 게임과 같은 실시간 상호 작용 시스템은 참가자들 간의 동일한 뷰를 제공할 수 있어야 하며 동시에 참가자가 만족할 만한 상호 작용 성능을 제공해야 한다. 하지만 이러한 두 가지 고려 사항은 서로 상충적인 성격을 가진다.

### 3.1 네트워크 트래픽 상태 측정

모든 참가자들은 네트워크 트래픽 상태에 따라 재생 지연 시간을 조정하기 위해 일정한 시간 간격마다 네트워크 트래픽 상태를 측정한다. 송신자는 수신을 이용하여 k번째 시간 간격 동안 자신이 수신한 이벤트에 대한 손실률  $R(k)$ 를 계산하고 이를 자신이 발생시킨 이벤트에 포함하여 타 참가자들에게 알린다. 이렇게 손실률 정보를 포함한 이벤트를 ‘피기백

이벤트'라 한다.

$$R(k) = \frac{N_{late}(k) + N_{lost}(k)}{N_{total}(k)}$$

$N_{total}(k) = N_{success}(k) + N_{late}(k) + N_{lost}(k)$ , 단,  $k=1, 2, 3, \dots$ 이고,  $R(0)=0$ 이다.

수식에서  $N_{success}(k)$ ,  $N_{lost}(k)$ ,  $N_{late}(k)$ 는  $k$ 번째 시간 간격 동안 재생 시각 전에 수신한 이벤트의 수, 네트워크 상에서 손실된 이벤트의 수, 재생 시각이 지난 후 수신된 이벤트의 수를 각각 의미한다.

### 3.2 재생 지연 시간 결정

각 참가자들은 3.1절에서 설명한 바와 같이 네트워크 트래픽 상태를 판단하고, 이에 따라 자신이 전송하는 이벤트에 적용할 재생 지연 시간을 결정한다. 여기서의 재생 지연 시간은 MiMaze에서와 동일한 개념이다. 네트워크 트래픽이 고부하 상태이면, 재생 지연 시간을 늘리고 네트워크 트래픽이 저부하 상태이면, 재생 지연 시간을 줄이도록 한다. 재생 지연 시간의 증감은 TCP에서 사용하는 AIMD (Additive Increase Multiplicative Decrease) 방식과 유사하게 재생 지연 시간 감소량과 증가량을 다르게 한다. 송신자 A가  $(k+1)$  번째 시간 간격에서 적용할 재생 지연 시간  $P(k+1)$ 의 계산은 수식과 같다.

$$P(k+1) = \begin{cases} P(k) - \alpha_1 (Lc - L(k)), & \text{저부하 상태인 경우} \\ P(k) + \alpha_2 (L(k) - Lc), & \text{고부하 상태인 경우} \end{cases}$$

### 3.3 동기화 기법 구현

본 논문에서 제시하는 동기화 기법은 ICU 협동 분산 네트워크 연구실에서 개발한 네트워크 가상 환경 프레임워크인 ATLAS를 기반으로 구현되었다. 모든 참가자가 동일한 시각을 갖는 것을 보장하기 위해 자체 타임 서버로 Tardis를 설치하고 클라이언트의 시각을 맞추기 위해 K9을 사용하였다. Tardis는 64초마다 NTP 신호를 보내고 K9은 이 신호를 받아 클라이언트 컴퓨터의 시각을 맞춘다. ATLAS는 가상환경을 위한 확장성 있는 네트워크 프레임워크로 세션 관리자(Session Manager), 영역 관리자(Region Manager), 이벤트 관리자(Event Manager), 통신 관리자(Communication Manager), 동기화 관리자(Synchronization Manager) 모듈로 이루어진다. 통신 관리자는 참가자들 간 통신 채널의 생성, 삭제 등을 수행하며 시스템 구조에 따라 클라이언트-서

버, 피어-피어, 혹은 피어-서버 구조를 지원한다. 이벤트 관리자는 통신 관리자로부터 네트워크 메시지를 받아 ATLAS 이벤트 메시지로 변환하여 상위 레벨로 전달하거나 상위 레벨로부터 ATLAS 이벤트 메시지를 받아 네트워크 메시지로 변환하여 통신 관리자에게 전달한다. 동기화 관리자는 이벤트 관리자로부터 메시지를 받아 이벤트 버퍼에 저장한 후, 예정 재생 시각이 되면 상위 레벨, 즉 ATLAS 응용 계층에 전달한다. 이벤트 관리자는 네트워크 메시지를 ATLAS 이벤트 메시지로 변환하여 동기화 관리자에게 전달한다. 그리고, 이벤트 헤더에 포함되어 있는 재생 지연 시간과 이벤트 발생 시각을 이용하여 해당 이벤트의 예정 재생 시각을 계산한다. 현재 시각이 재생 시각 이후이면 해당 이벤트를 버리고, 그렇지 않으면, 해당 이벤트는 이벤트 버퍼에 저장한다. ATLAS 이벤트 메시지는 81 바이트의 고정된 헤더 부분과 가변적인 데이터 부분으로 이루어진다. Bytes of Event 필드는 ATLAS 이벤트 메시지의 바이트 수를 나타내며 메시지에 포함된 ATLAS 이벤트를 식별하기 위해 Event Type과 Event ID 필드가 사용된다. 그리고, Handler 필드는 서로 다른 타입의 이벤트를 처리하기 위한 이벤트 핸들러를 나타낸다. 본 논문에서 제안하는 이벤트 동기화 기법을 위해 RF, Playout Delay, Drop Rate, 그리고 Generation Time 필드를 추가하였다. RF 필드는 해당 이벤트가 피기백 이벤트인지 아닌지를 표시하는 플래그로 사용된다. 피기백 이벤트의 경우에는, RF 필드를 1로 설정하고, Drop Rate 필드에 이벤트 송신자가 측정한 이벤트 손실률을 기록한다.

## 4. 3차원 그래픽 처리

가상 세계는 여러 개의 셀로 나뉘어진 가상 세계의 평면도로서의 맵 파일과 맵에서의 각 셀에 위치한 객체들의 기하학적 정의를 가지는 객체 파일들로 나뉘어진다. 본 논문에서는 미로 환경에 근거한 가상 세계와 3차원 지형에 근거한 가상세계 두 모델을 각각 구현하였지만, 그들의 구조는 여러 개로 나뉘어진 셀을 근거로하고 있다는 점에서 동일하다. 다만 지형에 근거한 경우 가시적 시야가 넓기 때문에 셀들의 2차원 배열을 quadtree로 구조화하여 계산의 편의성을 도모하였다. 프레임 레이트는 한 프레임을 렌더링하는데 걸리는 평균적인 시간이다. 이 중 렌더링 시간에 가장 중요하게 영향을 미치는 단계는

데이터베이스 쿼리 단계와 렌더링 단계이다. 다음 두 소절에서는 이 두 단계에서의 렌더링 시간을 줄이기 위해, 즉 안정적인 프레임 레이트를 얻기 위해 어떤 방법들이 이용되었는지를 설명한다.

#### 4.1 수행시간 프레임 레이트 모니터

실시간 그래픽 응용의 가장 중요한 척도인 프레임 레이트에 영향을 미치는 변수들은 화면의 복잡도, 메모리 대역폭, 프로세서의 부동소수점 연산 처리 능력 등으로 다양하며 실행 시간에 시시각각으로 변할 수 있다.

실행시간 모니터는 다음 식들에 의해 렌더링 속도에 영향을 미치는 인자들의 값을 조절한다. 한 프레임  $i$ 를 생성해내는데 걸리는 시간을  $T_R$ 이라 하고, 그 프레임에 있는 폴리곤의 수를  $P_i$ 라고 할 때, 다음과 같이 가정한다.

$$T_R \propto P_i$$

렌더링 레벨  $l$ 과 프레임  $i$ 에 대해 각 렌더링 인자  $Param(l, i)$ 의 값을 조정하기 위하여 다음 식을 이용하였다.

$$Param(l, i) = Param(l, i-1) + coeff(l) * floor((T_R(i-1) - T_T) / T_{step})$$

### 5. 결과

본 연구에서는 가상현실 시스템을 구현하기 위한 개념들을 제시하고 그의 한 응용으로 미로 환경과 3차원 지형 환경의 가상 세계 내비게이션에 기반하여 효율적인 실시간 3차원 그래픽 처리와 통신을 위한 시스템을 보였다.

지금까지 구현된 프레임워크는 초보적인 형태에 머물러 있으며, 기능적인 면의 확장을 통해 가상현실 응용을 위한 프레임워크의 제작으로 발전시키고자 한다. 이와 결부된 향후 연구 과제는 대규모 시뮬레이션에서의 멀티캐스트 통신에 대한 시뮬레이션, 보다 풍부한 가상 세계를 위해 각 객체들에 행동의 양식의 지정과, 이 객체의 행동이 통신을 통해 분산 처리되는 과정을 분석하는 것이다. 이를 위해 메시지 포맷의 보다 정교한 설계와 효과적인 애니메이션 기법들이 필요하다

고정적 재생 지연 시간을 사용하는 동기화 기법은 참가자들에게 동일한 뷰를 제공하기 위해 일정한 재생 시각이 될 때까지 이벤트를 수신자의 시스템에 저장한 후 참가자들끼리 동시에 이벤트를 실행시킨

다. 그러나 이러한 방식은 가변적 네트워크 상태에서 전송 지연 시간이 짧은 경우 대부분의 참가자가 이벤트를 빨리 수신함에도 불구하고 일정 시간 동안 수신자 시스템에 저장해야 한다. 또한 네트워크 상에 트래픽이 증가하여 전송 지연 시간이 길어지면 재생 지연 시간이 지난 후에 수신되는 이벤트의 수가 증가하여 손실된 이벤트를 보상하기 위한 추가 처리 부담을 초래하기 때문에 가변적으로 네트워크 상태가 변하는 환경에서는 비효율적이다.

본 논문에서는 네트워크 상태를 판단하는 기준으로 평균 이벤트 손실률만을 사용하였으나 이보다 더 다양한 네트워크 상태를 기술할 수 있는 기준들을 고려하여 효율성을 높이는 작업을 진행할 것이다. 그리고, 실제 개발된 시스템의 시험 환경도 광역 네트워크로 확장하여 제안하는 동기화 기법의 확장성 등을 검증할 것이다.

#### 참고문헌

- [1] M. Ahamad and R. Kordale, "Scalable Consistency Protocols for Distributed Services," IEEE Transactions on Parallel and Distributed Systems, Vol.10, No.9, pp.888-903, 1999.
- [2] S.Bhola, G Banavar and M. Ahamad, "Responsiveness and Consistency Tradeoffs in Interactive Groupware," Proc. of the ACM Conference on Computer Supported Cooperative Work, pp.79-88, Seattle, 1998.
- [3] K.I. Calvert, M.B. Doar, and E.W. Qequra, "Modeling Internet Topology," IEEE Communications Magazine, Vol.35, No.6, pp.160-163, 1997.
- [4] 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, No.3, pp.411-414, 1999.
- [5] 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.
- [6] L.Gautier and C.Diot, "End-to-end Transmission Control Mechanisms for Multiparty Interactive Applications on the Internet," Proc. of the Eighteenth Annual Joint Conference of the IEEE

Computer and Communications Societies, Vol.3,  
pp.1470-1479, 1999.

- [7] E. Hong et al., "An Efficient Synchronization Mechanism Adapting to Dynamic Network State for Internet-based 3D Cyber Marathon," Proc. of HCI 2003 Workshop, pp.168-173, Seoul, 2003.