

# 다중 사용자 게임 성능 향상을 위한 데이터 가상 그룹핑 방법

## (Virtual Data Grouping for Performance Enhancement of Multi-User Games)

이 철 민 <sup>†</sup>      박 흥 성 <sup>\*\*</sup>  
 (Lee Chul Min) (Park Hong Seong)

**요 약** 본 논문은 다중 사용자 네트워크 게임에서 응답 시간과 응답 데이터 손실을 줄일 수 있는 가상 그룹핑 방법을 제안한다. 제안하는 방법은 게임상의 모든 맵을 일정 크기의 영역으로 분할하고, 이들 각각의 영역을 그룹으로 묶은 후 각각의 그룹을 가상의 그룹으로 분할하고 각각의 가상 그룹에 데이터를 전송하는 방법이다. 또한 본 논문에서는 주어진 비용 함수를 최소화하는 최적 그룹수를 유도하였고, 제안된 방법이 기존의 그룹핑 방법과 비교하여 유용함을 보였다.

**키워드** : 데이터 그룹핑, 네트워크 게임, 게임 서버, 클라이언트/서버

**Abstract** This paper presents a virtual grouping method used in multi-user network games, which reduces a response time and losses of response data. The proposed method divides each group into virtual groups and transmits data in them after dividing an overall map on a game into several fixed regions and grouping them. And this paper derives the optimal number of groups minimizing a given cost function. The proposed method is shown to be useful by comparing with a general grouping method.

**Key words** : data grouping, network game, game server, client/server

### 1. 서론

인터넷이 발전함에 따라 게임 산업에도 많은 변화가 일어났다. 기존의 게임은 사용자 혼자서 즐기는 싱글 게임이나 모뎀을 이용하여 4~8명 정도의 사용자들이 같이 게임을 즐기는 정도의 게임이 주류를 이루고 있었다. 그러나 현재는 인터넷을 기반으로 하여 3000~4000명이 동시에 하나의 서버에 접속하여 게임을 즐기는 네트워크 게임이 주류로 발전하고 있다.

일반적인 네트워크 게임의 구조는 그림 1처럼 하나의 서버와 다수의 클라이언트들로 구성된다. 서버는 클라이언트들로부터 수신되는 요청을 처리하여 그 결과를 클라이언트에 전송하는 역할을 하고, 클라이언트는 서버로부

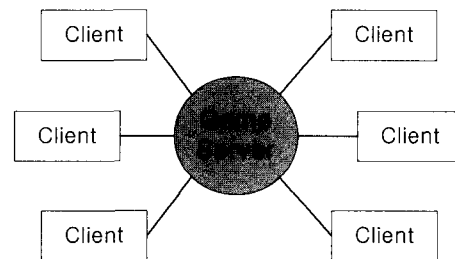


그림 1 네트워크 게임의 서버/클라이언트 구조

터 수신한 응답 결과에 따라 게임을 진행하게 된다.

네트워크 게임은 하나의 서버에 다수의 클라이언트들이 접속해서 동시에 게임을 진행하므로 같이 게임을 하고 있는 다른 클라이언트들의 현재 상태나 위치, 동작 등에 대한 정보가 매우 중요하다. 만약 이러한 정보들이 동기화 되지 않는다면, 클라이언트마다 게임 상황이 다르게 되어 정상적인 게임을 진행할 수 없게 된다. 이러한 문제를 해결하기 위해서는 클라이언트들은 주기적으로 서로

<sup>†</sup> 학생회원 : 강원대학교 전기전자정보통신공학부  
 jazonsim@control.kangwon.ac.kr

<sup>\*\*</sup> 종신회원 : 강원대학교 전기전자정보통신공학부 교수  
 hspark@kangwon.ac.kr

논문접수 : 2002년 4월 22일

심사완료 : 2002년 11월 29일

의 정보를 교환해야한다. 이 경우  $N$ 명의 클라이언트들이 게임을 하고 있다면, 서버가 전송해야하는 데이터 개수는  $N \times N$ 개가된다. 그러므로 클라이언트의 수가 증가할수록 게임 서버와 네트워크에는 더욱 많은 부하가 걸리게 된다. 이렇게 서버와 네트워크에 많은 부하가 걸리게 되는 경우, 클라이언트는 서버의 처리 지연이나 네트워크 트래픽 증가로 인한 네트워크 전송 지연 때문에 해당 요청에 대한 응답을 늦게 수신하거나, 최악의 경우 과도한 네트워크 트래픽으로 인한 패킷 손실이 발생하여 응답을 받지 못하는 경우가 존재한다.

응답 결과를 다른 사용자보다 늦게 받는다면 즉 응답이 지연되면 해당 처리가 다른 사용자보다 늦기 때문에 게임 결과가 달라진다. 또한 패킷 손실로 서버로부터 응답을 받지 못하는 경우, 즉 응답 손실의 경우 사용자는 해당하는 처리를 못하게 되어 게임을 진행하지 못하게 된다. 따라서 네트워크 게임을 원활하게 운영하기 위해서는 이러한 지연이나 손실 문제를 해결할 필요가 있다.

[1,2]에서는  $N \times N$ 연결이 이루어지는 경우의 동기화 분석을 하였는데, 대상 환경이 빠른 LAN에서 적은 수의 클라이언트들을 대상으로 하였으며, 또한 전송회수를 줄일 수 있는 방법이 없기 때문에 많은 클라이언트가 연결되는 경우 클라이언트 및 서버의 부하와 네트워크의 부하가 증가하게 되어 동기화 유지가 불가능하다.

네트워크 게임에서의 지연을 해결하기 위해 여러 방법이 제안되었다[3,4,5,6]. [3]은 패킷의 크기를 줄이는 것은 한계가 있고, 전송량을 줄이는 방안이 없으므로 클라이언트 수가 증가하는 경우 전송 개수는  $N \times N$ 으로 증가하는 문제가 있다. 방법[4]의 경우, 서버 프로그램 구조 개선은 개발하는 응용에 따라 많은 변화가 있는 부분이므로 모든 응용에 적용하기 어렵고, 분산 서버 구조를 사용하는 경우 추가적인 하드웨어 비용 및 분산 서버의 제어 등의 문제가 존재한다. [5]에서는 클라이언트들의 가상 범위(Vision Domain)에 따라 클라이언트들을 그룹화 하여 필요 없는 데이터 수신을 줄이는 방법을 제안하였다. 하지만 클라이언트가 직접 그룹 관리를 해야하고, 주기적으로 자신이 참여할 그룹을 검색해야 하는 등의 그룹관리에 많은 오버헤드가 존재하고, 클라이언트들 사이에서 데이터 전송이 발생할 때마다 그룹을 검사해야하므로 처리 부하가 심해지는 문제점이 있다. [6]에서는, 맵을 특정 크기의 영역으로 분할하고 그 영역 내에 있는 클라이언트들을 그룹으로 묶어, 그룹 단위로 데이터를 전송함으로써 서버의 데이터 전송 회수를 줄일 수 있는 방법을 제안하였다. 하지만, 서버의 전송 회수를 더욱 줄이기 위해서는 영역을 더욱 작게 나누어야하므로 영역을 관리하는 처리 프로세

스 개수가 증가하게 되어 서버의 부하가 증가하는 문제가 있다.

요약하면 [1-4]는 그룹화보다는 동기화분석[1,2], 패킷 크기 줄이는 방법[3], 운영체제 개선을 통한 서버 성능 향상 방법[4]를 제안하였다. 게임에서 맵이 동적으로 바뀐다면 동적으로 그룹화[5]하는 것이 적당하나, 일반적으로 맵이 정적으로 고정되고 그 맵에서 사용자가 움직이기 때문에 정적 맵을 기준으로 정적으로 그룹화[6]하는 것이 동적으로 그룹화 하는 것보다 서버 부하등의 성능측면에서 효율적이다.

본 논문에서는 클라이언트 그룹화를 통한 서버의 데이터 전송 개수를 줄이는 방법을 제안한다. 이를 위해 기존 그룹 단위 전송 방법[6]을 기본으로 그룹에 포함된 클라이언트들을 가상 그룹으로 분할하는 방법을 사용하여 영역을 더 작게 나누지 않고도 전송 회수를 줄여 서버의 부하와 네트워크 부하를 동시에 줄일 수 있는 방법을 제안하고, 성능 분석을 통해 이 방법의 유효성을 보이도록 한다

2장에서는 기존 그룹 단위 전송 방법[6]의 문제들을 기술하고 이 문제들을 해결하기 위한 인접 그룹 전송 방식을 제안한다. 3장에서는 2장의 인접 그룹 전송 방식을 개선한 가상 그룹화 방식을 제안하고, 4장에서는 인접 그룹 단위 전송 방법과 가상 그룹화 방법의 성능 분석 및 비교한 후 5장에서 결론을 맺는다.

## 2. 인접 그룹 단위 전송 방법

다수의 클라이언트가 동시에 게임을 진행하는 네트워크 게임에서 클라이언트들 사이의 동기화를 유지하기 위해서는 클라이언트마다 서로의 위치나 상태 동작 등에 관련된 정보를 교환해야한다. 이 경우 게임 상에  $N$ 개의 클라이언트가 있는 경우 서버가 클라이언트로부터 수신하게되는 데이터 수는  $N$ 개이지만, 서버가 전송해야되는 데이터 수는  $N \times N$ 개가된다.  $N \times N$ 데이터 전송은 클라이언트 수가 적은 경우 서버나 네트워크에 주는 부하가 작기 때문에 많이 사용하는 방법이지만, 클라이언트가 증가하는 경우 서버가 전송해야할 데이터 개수는 기하급수적으로 증가하게 되어, 서버의 부하 및 네트워크의 부하가 증가하는 문제점이 있다.

클라이언트는 게임상의 가상 지역에서 게임을 진행하게 되는데, 이러한 게임상의 가상 지역들을 맵(map)이라고 부른다. 이러한 맵을 크기가 같은  $M \times M$ 개의 영역으로 분할한 후 각 영역 내에 존재하는 클라이언트들을 별도의 그룹으로 묶어 그룹내의 클라이언트에서 발생한 데이터는 그룹내의 클라이언트에게만 전송하도록 하면, 서버가 전송해야되는 총 데이터 수는  $N \times N$ 방식에 비해

$N \times N / M^2$ 로 줄어들게 된다. 이러한 그룹 전송 방법은 서버의 데이터 전송 개수를 줄이는 장점이 있으나, 그룹들의 경계에 인접해 있는 클라이언트들의 경우 자신이 속한 그룹의 클라이언트들 정보만을 수신해서는 자신에게 필요한 모든 클라이언트들의 정보를 얻지 못하는 문제점이 있다. 따라서 이러한 문제를 해결하기 위한 인접 그룹 전송 방식을 제안한다. 인접 그룹 전송 방식의 동작은 다음과 같다.

그림 2를 보면 그룹 5의 클라이언트 A는 그룹 1, 2, 4와 인접하고 있기 때문에 실제 수신해야하는 데이터는 그룹 1, 2, 4, 5의 데이터가 되어야 한다. 그러므로, 현재의 그룹과 주변의 그룹을 포함한 데이터 전송 그룹을 만들어 전송 그룹 전체에 데이터를 전송하도록 해야 한다.

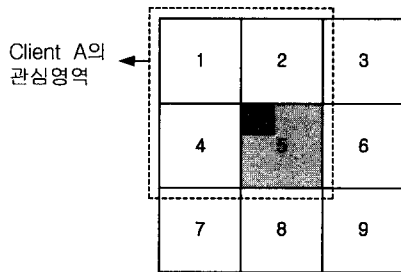


그림 2 클라이언트의 관심 영역

본 논문에서 사용되는 가정은 다음과 같다.

1. 클라이언트들은 맵상에 고르게 분포한다.
2. 맵은  $M \times M$ 개의 크기가 같은 영역으로 분할되고, 같은 영역내의 클라이언트들은 하나의 그룹으로 묶여, 별도의 프로세스에 의해 관리된다. 총 그룹수는  $M^2$  개가 된다.
3. 각 그룹마다 동일한 수의 클라이언트들이 있다.
4. 클라이언트들은 동기화 유지를 위해서 일정 주기마다 서버로 데이터를 전송해야하고, 이 데이터들은 해당 그룹 프로세스로 전달되어 처리된다. 처리 결과는 주기 내에 클라이언트로 도착해야 한다.
5. 그룹 프로세스는 그룹내의 클라이언트들로부터 수신한 데이터를 처리하여 그 결과를 데이터를 전송한 클라이언트 및 그룹내의 모든 클라이언트들에게 전송하고, 관심 대상 그룹 주변의 그룹들을 관리하는 그룹 프로세스들에게도 전달하여, 각 프로세스들로 하여금 이 데이터를 자신의 그룹 클라이언트들에게

전송하도록 한다.

본 논문에서 사용되는 기호들은 다음과 같이 정의한다.

- $P_T$  : 게임상의 전체 클라이언트 수
- $P_{SZ}$  : 특정 그룹의 클라이언트 수 ( $= P_T/M^2$ )
- $T_S$  : 클라이언트가 서버로 데이터를 전송하는 주기
- $R_G$  : 그룹단위 전송 방법에서 서버가  $T_S$  마다 클라이언트들로부터 수신하는 최대 데이터 개수
- $R_V$  : 가상 그룹화 방법에서 서버가  $T_S$  마다 클라이언트들로부터 수신하는 최대 데이터 개수
- $S_G$  : 그룹단위 전송 방법에서  $T_S$  마다 서버가 전송해야하는 최대 메시지 개수
- $S_V$  : 가상 그룹화 방법에서  $T_S$  마다 서버가 전송해야 하는 최대 메시지 개수
- $S_{GR}$  : 하나의 그룹 프로세스가  $T_S$  마다 한 그룹으로부터 수신하게 되는 최대 데이터 개수
- $P_G$  : 그룹 단위 전송 방법에서 최대 프로세스 개수
- $P_V$  : 가상 그룹화 방법에서 최대 프로세스 개수
- $L^2$  : 그룹을 가상 영역으로 분할할 때 가상 영역 개수 ( $L \times L$ )
- $C_G$  : 인접 그룹 전송 방법의 시간 비용 함수
- $C_V$  : 가상 그룹화 방법의 시간 비용 함수
- $\alpha$  : 수신된 하나의 데이터를 처리하는 시간
- $\beta$  : 한 클라이언트로 데이터를 전송하기 위해 처리하는 시간
- $\gamma$  : 클라이언트의 정보/위치 관리와 그룹 내부의 NPC (Non Playable Character) 관련 작업이나 기타 작업을 수행하는데 걸리는 시간.

그룹들은 별도의 프로세스가 관리하므로 전체 프로세스 개수는 전체 그룹수와 같다. 즉  $P_G = P_V = M^2$ 이 된다. 그리고 서버가  $T_S$  마다 수신하는 데이터 개수는 서버의 클라이언트 수  $P_T$ 와 같으므로  $R_G = R_V = P_T$ 가 된다.

관심 대상 그룹의 주변 그룹에 데이터를 전송하는 경우, 그림 3처럼 관심 대상 그룹의 위치에 따라 주변 그룹 수가 8, 5, 3개인 3가지 경우가 존재한다. 맵을  $M \times M$  개의 그룹으로 분할할 때 주위에 8개의 그룹이 연관되어있는 그룹의 수는  $(M-2)^2$ , 5개의 그룹이 연관되어 있는 그룹의 수는  $(M-2) \cdot 4$ , 3개의 그룹이 연관되어 있는 그룹의 수는 4개이다.

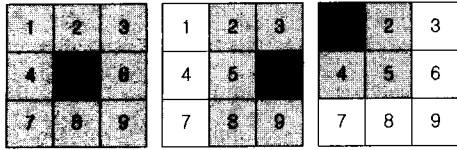


그림 3 그룹 위치에 따른 주변 그룹수

각 클라이언트는  $T_s$ 마다 최대 1개의 데이터를 보내므로 그룹 프로세스가  $T_s$ 마다 수신하게 되는 최대 데이터 개수  $S_{GR}$  는 현재 그룹에 포함된 클라이언트 수( $= P_{SZ}$ )와 같으므로 다음과 같다.

$$S_{GR} = P_{SZ} \cdot 1 \quad (1)$$

주변 그룹수가 8개인 그룹이 전송해야하는 데이터 개수는 자신 그룹 및 주변 8개 그룹에 전송해야 하므로  $S_{GR} \cdot (P_{SZ} + 8P_{SZ})$ 개가 되고 주변 그룹수가 8개인 그룹의 개수는  $(M-2)^2$ 이므로, 이러한 그룹들이 전송하는 총 데이터 개수는  $S_{GR} \cdot (P_{SZ} + 8P_{SZ}) \cdot (M-2)^2$  개가 된다. 주변 그룹수가 5개인 그룹들이 전송하는 총 데이터 개수는  $S_{GR} \cdot (P_{SZ} + 5P_{SZ}) \cdot (M-2) \cdot 4$ 개가 되고, 주변 그룹수가 3개인 그룹들이 전송하는 총 데이터 개수는  $S_{GR} \cdot (P_{SZ} + 3P_{SZ}) \cdot 4$ 개가 된다. 그러므로 모든 그룹들이 전송하는 총 데이터 개수, 즉 서버가 전송해야하는 최대 데이터 개수는 식 (2)와 같다.

$$S_G = S_{GR} \cdot 9P_{SZ} \cdot (M-2)^2 + S_{GR} \cdot 4P_{SZ} \cdot 4 + S_{GR} \cdot 6P_{SZ} \cdot (M-2) \cdot 4 \quad (2)$$

식 (1)에서  $S_{GR} = P_{SZ}$  이므로 식 (2)는 다음과 같이 정리된다.

$$S_G = P_{SZ}^2 \{ 9M^2 - 12M + 4 \} \quad (3)$$

그룹화 방법을 사용하지 않고,  $N \times N$  데이터 전송을 하는 경우를 살펴보면,  $T_s$ 마다 서버가 수신하는 최대 데이터 개수  $R$ 는 전체 클라이언트 수  $P_T (= N)$ 와 같으므로 다음과 같다.

$$R = P_T \cdot 1$$

이 경우 서버가 전송해야하는 최대 데이터 개수  $S$ 는 다음과 같다.

$$S = R \cdot P_T = P_T^2 = N^2 \quad (4)$$

식 (3),(4)를 비교해보면, (4)는 클라이언트 수가 증가함에 따라 전송 데이터수가 클라이언트 수에 제곱에 비해

하여 증가하고, (3)는 그룹수가 증가할수록,  $P_{SZ}^2$ 값이 감소하므로 데이터 전송 개수가 감소함을 알 수 있다.

### 3. 가상 그룹화

인접 그룹 단위 전송 방법을 사용하는 경우  $N \times N$  전송 방법과 비교하여 데이터 전송 횟수를 획기적으로 줄일 수 있지만, 불필요한 정보가 전달되는 경우가 발생한다. 그림 4와 같이 그룹 5를 가상 영역 5a, 5b, 5c, 5d로 구분했을 때, 5a 영역에 있는 클라이언트들의 정보는 그룹 1, 2, 4에게는 의미가 있지만, 그룹 3, 6, 7, 8, 9에게는 큰 의미가 없다.

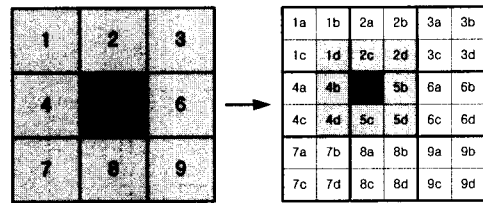


그림 4 주변에 8개의 그룹이 있는 경우 (L=2)

가상 그룹화 방법은 하나의 그룹을  $L \times L$  ( $L = 2, 3, \dots$ )개의 가상의 영역으로 나누어, 데이터를 전송한 클라이언트가 속한 가상 영역에 따라 데이터를 수신할 그룹과 가상 영역을 결정하는 방법이다.

한 그룹에서 주변 그룹의 일부 가상 영역에 데이터를 전송하는 경우는 관심 대상 그룹의 위치에 따라 그림 4, 5, 6과 같은 3가지 경우가 존재한다. L=2인 그림 4에서 그룹 5와 같이 주변에 8개의 그룹이 존재할 때, 가상 영역 5a에 존재하는 클라이언트로부터 데이터 수신이 발생하는 경우, 이 처리 결과는 그룹 5의 모든 클라이언트들과 주변 가상 영역 1d, 2c, 2d, 4b, 4d의 클라이언트들에게 전송된다. 이때 데이터를 수신하게되는 평균 가상 영역 개수는 9개가 된다.

그림 5의 그룹 4와 같이 주변에 5개의 그룹이 있는 경

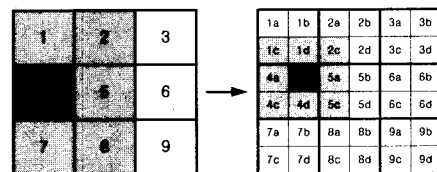


그림 5 주변에 5개의 그룹이 있는 경우

우, 가상영역 4a 및 4c의 클라이언트에서 데이터가 수신된 경우 이 데이터를 수신하게 되는 평균 가상 영역 개수는 6개이고, 가상영역 4b, 4d에서는 9개이다. 따라서 데이터를 수신하는 평균 가상 영역 개수는 7.5개가 된다.

그림 6의 그룹 1과 같이 주변에 3개의 그룹이 있는 경우, 가상영역 1a의 클라이언트로부터 데이터가 수신된 경우 이 데이터를 수신하게되는 가상 영역 개수는 4개이고, 가상영역 1b, 1c에서 수신된 경우는 6개의 가상영역이 데이터를 수신하게 되고, 가상영역 1d에서 수신된 경우는 9개의 가상영역이 데이터를 수신하게 된다. 따라서 데이터를 수신하는 평균 가상 영역 개수는 6.25개가 된다.

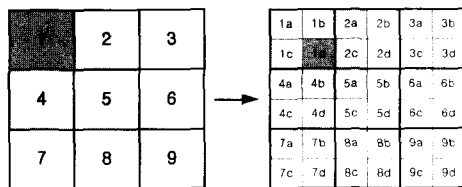


그림 6 주변에 3개의 그룹이 있는 경우

가상 그룹화 방법을 사용할 때 하나의 그룹 프로세스가  $T_s$  마다 그룹 클라이언트들로부터 수신하게 되는 최대 데이터 개수  $S_{GR}$ 은 현재 그룹에 포함된 클라이언트 수와 같으므로 다음과 같다.

$$S_{GR} = P_{SZ} \cdot 1$$

$L=2$  인 경우, 주기  $T_s$  마다 서버가 전송해야하는 최대 데이터 개수는 다음과 같다.

$$S_V = S_{GR} \cdot P_{SZ} \{ 2.25(M-2)^2 + 1.875(M-2) \cdot 4 + 1.5625 \cdot 4 \}$$

$$= P_{SZ}^2 \{ 2.25M^2 - 1.5M + 0.25 \}$$

$L=3$  인 경우, 각 그룹들은 그림 7처럼 9개의 가상영역으로 나누어진다.

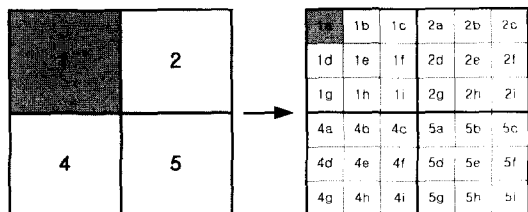


그림 7 그룹을 9개의 가상 영역으로 분할한 경우 ( $L=3$ )

$L=2$  인 경우를 참고하여,  $L=3$ 인 경우 주기  $T_s$  마다 서버가 전송해야하는 최대 데이터 개수를 구하면 다음과 같다.

$$S_V = S_{GR} \cdot P_{SZ} \{ (M-2)^2 + 0.88(M-2) \cdot 4 + 0.79 \cdot 4 \}$$

$$= P_{SZ}^2 \{ M^2 - 0.48M + 0.12 \}$$

$L=4$  인 경우, 각 그룹들은 그림 8처럼 16개의 가상영역으로 나누어진다.

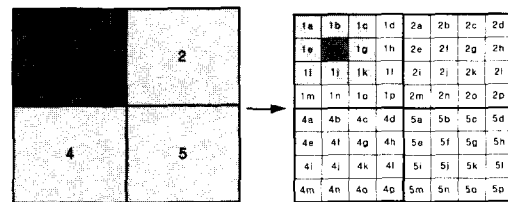


그림 8 그룹을 16개의 가상 영역으로 분할한 경우 ( $L=4$ )

이 경우 주기  $T_s$  마다 서버가 전송해야하는 최대 데이터 개수를 구하면 다음과 같다.

$$S_V = S_{GR} \cdot P_{SZ} \{ 0.56(M-2)^2 + 0.52(M-2) \cdot 4 + 0.47 \cdot 4 \}$$

$$= P_{SZ}^2 \{ 0.56M^2 - 0.16M + 0.04 \}$$

가상 그룹화 방법에서  $L$ 의 값에 따른 서버의 최대 데이터 전송 횟수  $S_V$ 를 인접 그룹 단위 전송 방법과 비교한 것이 그림 9이다. 그림에서  $S_V$ 는 서버가 전송해야할 최대 데이터 개수를 나타낸다.

그림 9를 살펴보면,  $L$  값이 증가함에 따라 서버의 최대 데이터 전송 개수가 줄어들음을 알 수 있다. 하지만 그림 7, 그림 8과 같이  $L \geq 3$  인 경우, 한 그룹 내에서 발생한 데

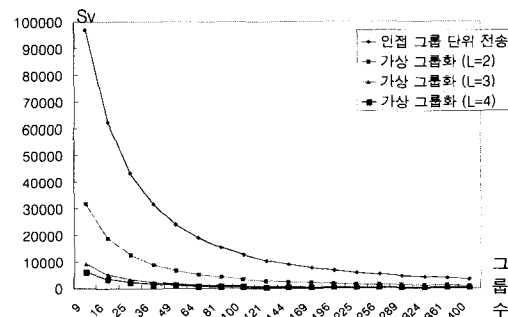


그림 9 전송 방법에 따른 서버의 최대 데이터 전송 개수

이타가 그룹 내의 모든 클라이언트로 전달되지 않는 경우가 생기게 된다. 본 논문에서는 인접 그룹 단위 전송 방법과의 비교를 위하여 L=2 인 경우를 가정하여 성능 분석을 하겠다.

4. 성능 분석

인접 그룹 단위 전송 방법과 가상 그룹화 방법(L=2)에서 클라이언트 수와 그룹 개수에 따라 서버가 전송해야 하는 최대 데이터 수를 비교한 것이 그림 10 - 그림12 에 있다.

그림 10 - 그림 12를 살펴보면, 두 방법 사이의 전송 개수 비율은 클라이언트 수에 상관없이 평균 3.68이 된다. 즉, 인접 그룹 단위 전송 방법이 가상 그룹화 방법보다 평균 3.68 배의 데이터를 전송해야 함을 알 수 있다. 이러한 3.68배는 그룹수가 400일 때까지의 경우이며, 그룹수가 무한히 커질 때는 (5)와 같이 계산 될 수 있다.

$$\lim_{M \rightarrow \infty} \frac{S_G}{S_V} = \lim_{M \rightarrow \infty} \frac{P_{SZ}^2(9M^2 - 12M + 4)}{P_{SZ}^2(2.25M^2 - 1.5M + 0.25)} = \lim_{M \rightarrow \infty} \frac{9M^2 - 12M + 4}{2.25M^2 - 1.5M + 0.25} = 4 \quad (5)$$

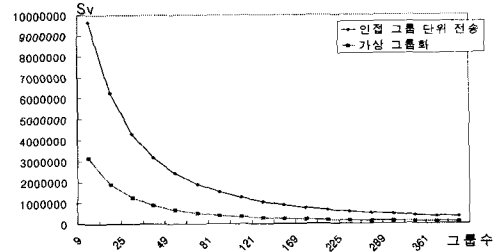


그림 12 클라이언트가 4000 일 때 그룹수에 따른 데이터 전송 개수

데이터 수신 및 전송 시간 및 프로세스 처리 시간을 고려하였을 때 인접 그룹 단위 전송 방법과 가상 그룹화 방법의 시간 비용 함수를 비교해 보겠다.

그림 13은 하나의 그룹 프로세스의 작업 처리를 3개의 단계로 나누어 보여 주고 있다.

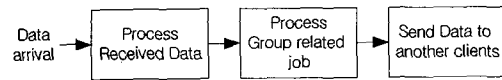


그림 13 그룹 프로세스의 처리 순서

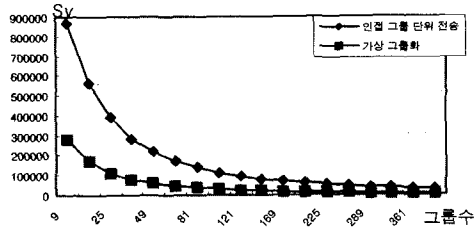


그림 10 클라이언트 수 1200 일 때 그룹수에 따른 데이터 전송 개수

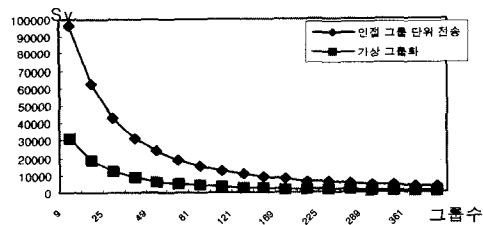


그림 11 클라이언트 수 400 일 때 그룹수에 따른 데이터 전송 개수

첫 단계인 “Process received data” 단계는 도착한 클라이언트의 요청을 분석하여 처리 및 관련 작업을 수행한다. 이 수행 시간이  $\alpha$  이므로, 하나의 그룹 프로세스가 수신되는 모든 메시지(=  $S_{GR}$ )를 처리하는 시간은  $\alpha \cdot S_{GR}$  이 된다. 서버의 프로세스 개수는  $M^2$  (=  $P_G, P_V$ )이므로 모든 프로세스들이 수신되는 모든 데이터를 처리하는 시간은  $\alpha \cdot S_{GR} \cdot M^2$  이 된다. 이때  $S_{GR} = P_{SZ}$  이고,  $P_{SZ} = PT/M^2$  이므로 모든 데이터를 처리하는 시간은  $\alpha \cdot P_T$  (=  $\alpha \cdot R_G, \alpha \cdot R_V$ )이 된다.

“Process Group related job” 단계에서는 그룹 내부 NPC들을 수행시키고, 그 결과와 클라이언트들로부터 수신한 요청을 처리한 결과를 이용해서 클라이언트들의 정보와 위치 등을 조절하고, 기타 그룹에 관련된 작업을 수행한다. 이 수행시간이  $\gamma$  이므로, 모든 프로세스들이 이 단계에서 소요되는 시간은  $\gamma \cdot M^2$  (=  $\gamma \cdot P_G = \gamma \cdot P_V$ )이 된다.

“Send data to another clients” 단계에서는 처리 결과를 수신할 클라이언트들의 목록을 결정한 후 해당 클라이언트들에게 데이터를 전송한다. 한 그룹 프로세스에서 해당 클라이언트들에게 모든 데이터를 전송하는 시간은

$\beta \times S_{GR} \times$  전송 그룹수 이다. 전송 그룹수는 관심 그룹의 위치와 전송 방법에 따라 달라지게 되어, 인접 그룹 단위 전송 방법의 경우  $S_G$ 가 되고, 가상 그룹화 방법의 경우에는  $S_V$ 가 된다. 그러므로 모든 데이터를 전송하는 걸리는 시간은 인접 그룹 단위 전송 방법일 경우  $\beta \cdot S_G$  이고, 가상 그룹화 방법일 경우에는  $\beta \cdot S_V$ 가 된다.

모든 프로세스의 각 단계에서 소요되는 시간들을 모두 합한 시간이 서버의 전체 수행시간이 된다. 그러므로 인접 그룹 단위 전송 방법에서의 시간 비용 함수,  $C_G$ 는 다음과 같다.

$$C_G = \alpha \cdot R_G + \beta \cdot S_G + \gamma \cdot P_G$$

$$= \alpha \cdot P_T + \beta \cdot P_{sz}^2(9M^2 - 12M + 4) + \gamma \cdot M^2$$

가상 그룹화 방법의 시간 비용 함수,  $C_V$ 는 다음과 같다.

$$C_V = \alpha \cdot R_V + \beta \cdot S_V + \gamma \cdot P_V$$

$$= \alpha \cdot P_T + \beta \cdot P_{sz}^2(2.25M^2 - 1.5M + 0.25) + \gamma \cdot M^2$$

$\alpha, \beta, \gamma$  값은 운영 환경과 개발하는 응용에 따라 달라지겠지만, 본 논문에서는  $\beta$ 를 1 비용 시간이라 할 때,  $\alpha$

는  $\beta$ 에 비해서 많은 작업을 수행하므로 3 또는 6의 시간 비용을 갖는 경우를 가정하고,  $\gamma$ 는 20 또는 50의 비용 시간을 갖는 경우를 가정하여 클라이언트 수가 400일 때, 각 경우에서 인접 그룹 단위 전송 방법과 가상 그룹화 방법 사이의 시간 비용을 비교해보겠다. 그림에서  $C_V$ 는 가상 그룹화 방법에서의 시간 비용함수를 나타낸다.

수신된 데이터를 처리하는 시간  $\alpha$ 의 영향은 인접 그룹 단위 전송 방법 및 가상 그룹화 방법 모두 같은 수의 데이터를 수신하므로 시간 비용에 일정한 상수를 더하는 효과만을 나타낸다.  $\beta$ 와  $\gamma$ 가 일정하고,  $\alpha$ 의 값이 다른 값을 갖는 그림 14와 그림 16을 비교해보면 그림 16의 그래프가 그림 14의 그래프보다 일정 값만큼 크음을 보여준다. 데이터 전송 시간  $\beta$ 의 영향은 서버가 전송하는 최대 데이터 개수에 따라 그 영향이 결정된다. 즉, 그룹수 (= 프로세스 수)가 최소일 때 서버의 데이터 전송 개수는 최대가 되고, 그룹수가 최대일 때 서버의 데이터 전송 개수가 최소가 되므로, 그림 14 - 그림 17을 살펴보면, 그룹수가 최소일 때 최대의 시간 비용을 나타내고, 그룹수가 최대가 되면 최소의 시간 비용을 나타낸다. 처리시간  $\gamma$ 의

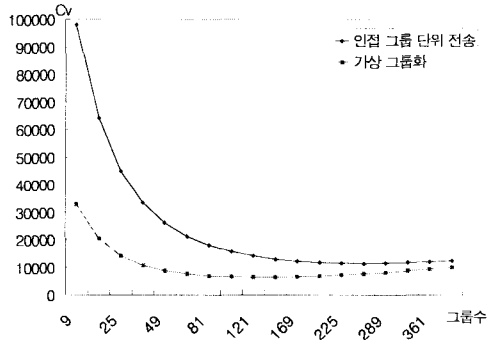


그림 14  $\alpha=3, \beta=1, \gamma=20$  일 때 시간 비용

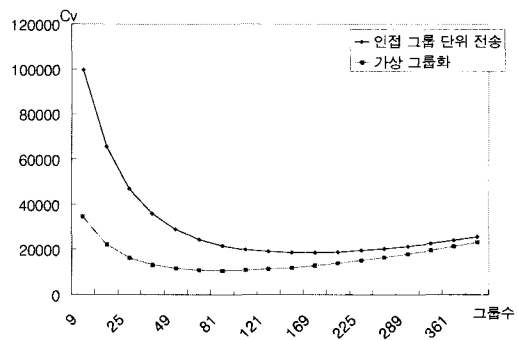


그림 15  $\alpha=6, \beta=1, \gamma=50$  일 때 시간 비용

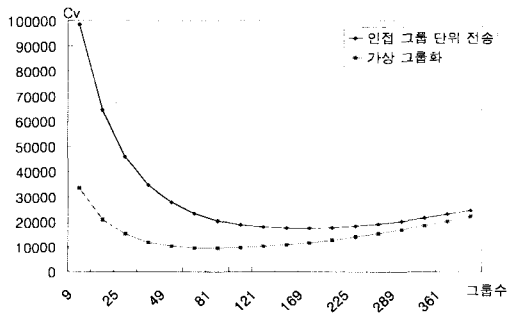


그림 16  $\alpha=3, \beta=1, \gamma=50$  일 때 시간 비용

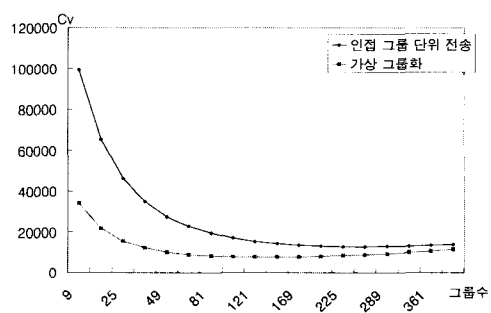


그림 17  $\alpha=6, \beta=1, \gamma=20$  일 때 시간 비용

영향은,  $\alpha$ 와  $\beta$ 가 일정하고,  $\gamma$ 의 값이 다른 값을 갖는 그림 14 와 그림 15, 그림 16과 그림 17을 비교해보면,  $\gamma$  값이 커짐에 따라 그룹수가 증가할수록 시간 비용이 급격히 증가함을 알 수 있다.

그림 14에서  $C_G$  및  $C_V$ 를 최소화하는 최적 그룹수는 각각 256, 121개이며, 이때 시간 비용은 각각 11486, 6417 이다. 그림 17에서  $C_G$  및  $C_V$ 를 최소화하는 최적 그룹수는 각각 169, 81개이며, 이때 시간 비용은 각각 18520, 10571 이다. 따라서  $C_V$ 가 훨씬 성능이 좋다는 것을 알 수 있다.

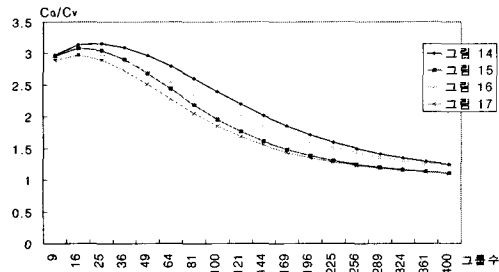


그림 18 전송 방법에 따른 시간 비용 비율 ( $C_G/C_V$ )

그림 18은 그림 14 - 그림 17의 인접 그룹 단위 전송 방법과 가상 그룹화 방법에서의 시간 비용 값의 비율을 나타낸 것이다. 그룹수가 적은 경우 가상 그룹화 방법의 효과가 최대가 되고, 그룹수가 증가함에 따라 효과가 최소가 됨을 알 수 있다. 그림에서  $C_V$ 는 가상 그룹화 방법에서의 시간 비용함수이고,  $C_G$ 는 인접 그룹 전송 방법에서의 시간 비용함수를 나타낸다.

## 5. 결론

본 논문에서는, 네트워크 게임의 지연을 최소화하기 위한 방법으로, 게임상의 맵을 일정 영역으로 분할하여 그룹화한 후, 각 그룹을 가상 영역으로 분할하여 데이터를 전송함으로써 서버의 데이터 전송 횟수를 최소화하여 서버의 부하 및 네트워크 부하를 줄여 응답 지연 및 응답 손실을 줄이는 방법을 제안하였다.

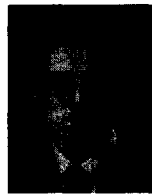
제안한 가상 그룹화 방법을 사용하는 경우, 가상 영역을 4개로 나누어 사용할 때, 인접 그룹 단위 전송 방법과 비교하여 전송 횟수가 평균 1/3.68 정도로 감소하게 되어 서버의 처리 부하 및 네트워크 부하를 해결하는데 효과적임을 알 수 있다. 서버의 데이터 전송 횟수가 줄어들수록 클라이언트가 수신하는 데이터 수가 줄어들게 되어

클라이언트의 처리 부하 또한 감소하는 효과가 있다. 또한 비용함수를 도입하여 주어진 비용 함수를 최소화하게 하는 최적 그룹수를 구하였다. 이러한 최적 그룹수로 비교하였을 때, 가상 그룹화 방식이 인접 그룹 단위 전송 방법보다 1.75 배정도 비용이 최소화됨을 알 수 있었다. 따라서 본 논문에서 제안한 가상 그룹화 방법이 유효함을 보였다.

앞으로의 연구과제는 전송최수에 따른 네트워크 트래픽을 고려한 성능분석과 특정 영역에 많은 클라이언트가 몰리거나, 특정 영역에 적은 수의 클라이언트가 있는 경우, 클라이언트가 많은 영역은 좀더 작은 영역으로 세분화하고, 클라이언트가 적은 영역들을 하나의 그룹으로 묶어 처리의 효율성을 높이는 방법을 연구해야 한다.

## 참고 문헌

- [1] Laurent Gautier, Christophe Diot, "Design and Evaluation of MiMaze, a Multi-Player Game on the Internet", IEEE Multimedia Systems Conference, 1998, 6
- [2] M.S. Borella, "Source models of network game traffic", IEEE Computer Communications, vol 23, 403-410, 2000
- [3] Yu-Shen Ng, *Designing Fast-Action Game For The Internet*, <http://www.gamasutra.com/>, 1997,9
- [4] 신동원, "첨단 기술 네트워크 게임을 위한 운영체제 연구", <http://www.synczone.net/inmainframe.html>
- [5] Li Zou, Mostafa H. Ammar, *An Evaluation of Grouping Techniques for State Dissemination in Networked Multi-User Games*, Tech. Rep. Georgia Institute of Technology, 1999,5
- [6] Jesse Aronson, "Using Grouping for networked gaming", Game Developer, 1997,8



이 철 민

1993년 강원대학교 제어계측공학과 졸업.  
동 대학원 석사(1996) 1997년~현재 동 대학원 박사과정



박 홍 성

1983년 서울대 제어계측공학과 졸업. 동 대학원 석사(1986). 동 대학원 박사(1992) 1992년~현재 강원대 전기전자정보통신공학부 부교수 관심분야는 실시간 네트워크, 무선 네트워크 등