# 온라인 게임 동기화를 위한 데이터 송수신 구조※

주우석
동서대학교 디지털콘텐츠학부
savrang@dongseo.ac.kr

# Data Send-Receive Structure for Online Game Synchronization

Woo-Suk Ju
Dept. of Digital Contents, Dongseo University

## 요 약

본 논문에서는 게임 진행의 동기화를 위해 빠른 데이터 송수신이 가능한 통신 구조에 대해서 제안한다. 이를 위해서 우선, 온라인 게임에서 데이터 송수신 속도가 게임의 장르에 따라 미치는 영향에 대해서 분석하였다. 그리고 게임의 동기화가 게임 진행에 큰 영향을 주는 실시간 온라인 대전 게임과 같은 장르에 적합한 데이터 통신 구조에 대해서 제안하였다. 본 연구에서 제안한 통신 구조는 TCP/UDP 프로토콜을 함께 이용하였으며 공유기를 사용하는 네트워크 환경에서도 적용하여 데이터 송수신 속도가 향상 되었음을 확인하였다. 이러한 데이터 송수신 구조는 다양한 온라인 게임에 적용할 수 있을 것이다.

## ABSTRACT

On this paper, suggests a communication structure making possible to fast data send-receive for the gaming process synchronization. For these things, first of all, analyze the data send-receive speed effecting on the online game by the game genre. Moreover, we suggested a proper data communication structure like real-time online action game genre which gaming synchronization effects on the gaming process. Communication structure suggested on this paper is being used with TCP/UDP protocol, and it is applied to the network system and verify improved data send-receive speed. These data send-receive structure can be applied to the various online games.

**Keywords :** Online Game, TCP/UDP, Hole Punching, Synchronization

접수일자 : 2010년 09월 30일, 일차수정 : 2010년 12월 01일, 심사완료 : 2010년 12월 03일
※ 본 연구는 지식경제부의 지역혁신센터의 연구결과로 수행되었음.

## 1. 서 론

단일 호스트에서 혼자서 플레이하는 게임이 대다수였던 과거와 다르게 인터넷의 보급으로 인해 최근에는 네트워크를 통하여 플레이하는 게임이 대중적으로 크게 인기를 얻고 있다[1].

국내 온라인 게임시장은 MMORPG 장르를 중심으로 성장하였고 국내외 많은 개발사들이 막대한 비용과 인력을 투자하여 온라인 게임을 개발하고 있다[2].

이로 인해 참신한 게임소재의 부재와 과도한 경쟁을 피하고 리니지2, 월드오브워크래프트와 같은 거대 게임 개발사들이 독점한 시장의 틈새를 노려 새로운 장르의 게임들이 개발되기 시작하였다.

오랜 시간 게임을 해야 하는 MMORPG 보다는 짧은 시간에 쉬운 조작을 통하여 즐길 수 있는 게임들이 등장하기 시작하였고 최근에는 레이싱, 격투, 스포츠 등의 실시간 온라인 대전 게임들이 인기를 얻으면서 다양한 게임들이 개발되고 있다.

온라인 대전 게임은 MMORPG의 제작 방식과 다르게 갖춰야 하는 몇 가지 특징이 있는데, 그 중에서 가장 중요한 것은 사용자의 빠른 입력을 처리할 수 있어야 한다. 대전 게임은 특성상 상대방의 동작에 순간적으로 반응하여 대전하는 형식을 취하기 때문에 사용자의 순간적인 입력에 즉각적인 반응을 해야 한다. 그렇지 못할 경우 게임에 대한 몰입도가 낮아지게 되고 이는 게임의 재미를 반감시키는 주요한 원인이 된다.

온라인상에서 사용자의 입력 처리가 즉각적이어야 한다는 것은 결국, 빠른 데이터의 송수신이 이루어 져야한다는 것을 의미한다. 데이터의 송수신 속도는 사용되는 프로토콜과 밀접한 관련이 있는데, [표 1]은 온라인 게임 제작에서 가장 많이 사용하는 프로토콜을 비교한 것이다.

[표 1] TCP 프로토콜과 UDP 프로토콜의 비교[3]

| TCP 프로토콜 | UDP 프로토콜 |
|---|---|
| · 연결형(connection-oriented) 프로토콜<br>· 연결이 성공해야 통신 가능 | · 비연결형(connectionless) 프로토콜<br>· 연결 없이 통신 가능 |
| · 데이터 경계를 구분하지 않음<br>· 바이트 스트림(byte-stream) 서비스 | · 데이터 경계를 구분함<br>· 데이터그램(datagram) 서비스 |
| · 신뢰성 있는 데이터 전송<br>· 데이터를 재전송함 | · 비신뢰적인 데이터 전송<br>· 데이터를 재전송하지 않음 |

대부분의 게임들이 신뢰성 있는 데이터의 송수신을 보장하는 TCP 프로토콜을 사용하지만, 빠른 데이터 송수신을 필요로 하는 대전 게임의 경우, 송수신 되는 데이터의 특성에 따라 TCP 프로토콜과 함께 UDP 프로토콜을 병행하여 사용한다.

[표 2]는 대전 게임을 진행하기 위해 필요한 송수신 데이터들을 사용목적에 적합한 프로토콜로 구분해 놓았다.

데이터의 송수신 속도보다 중요도가 우선시 되어야 하는 사용자 인증이나 캐릭터 레벨, 아이템 정보 등은 TCP 프로토콜을 이용하고, 캐릭터들 간에 공격이나 이동정보와 같이 중요도는 낮지만 빨리 처리되어야 하는 데이터들은 UDP 프로토콜을 이용한다.

[표 2] 온라인 대전 게임에서 프로토콜의 사용 범위

| TCP 프로토콜 | UDP 프로토콜 |
|---|---|
| · 사용자 인증<br>· 게임 데이터(캐릭터, 아이템, 레벨, 스킬 등)<br>· 기타 게임 진행에 필요한 중요 데이터 | · 캐릭터의 이동/공격<br>· 기타 게임의 동기화를 위해 빠른 송수신이 필요한 데이터 |

[표 1]에 나와 있는 것처럼 UDP 프로토콜은 통신의 안정성을 보장하지 않는다는 것 외에 사설아이피와 공유기에 연결된 호스트들은 외부의 호스트

들과 UDP 프로토콜을 이용하여 통신할 수 없다는 치명적인 문제점을 가지고 있다.

본 논문에서는 UDP 프로토콜을 사용하여 P2P (Peer-to-Peer) 방식의 빠른 데이터 송수신이 필요한 온라인 대전 게임을 제작할 경우, 위에서 언급한 문제점들을 해결하기 위한 데이터 송수신 구조를 제안한다.

## 2. 관련연구

### 2.1 온라인 게임의 트래픽 분석

온라인 게임은 대부분 TCP와 UDP 프로토콜을 이용하여 서비스된다. TCP 프로토콜은 MMORPG에서 사용되며, UDP 프로토콜은 TCP 프로토콜과 함께 온라인 대전게임에서 사용된다.

네트워크 게임에서 네트워크 지연 문제를 장르별로 살펴보면 FPS와 같은 실시간 대전게임이 가장 큰 영향을 받는 반면, RTS와 MMORPG는 게임의 진행이 느리기 때문에 영향을 덜 받게 된다[4].

#### 2.1.1 MMORPG 트래픽 분석

MMORPG의 대역폭은 로그인, 맵의 이동, 게임 플레이에 의해 크게 변화한다. 대체로 대역폭이 증가하는 곳은 초기 접속시 게임 데이터 전송이 많은 로그인과 사냥터에서 마을 등으로 맵을 이동하는 과정에서 발생한다[5].

게임의 동기화를 위한 데이터의 송신 속도는 접속자 수와 게임 방식에 따라 크게 영향을 받게 되는데, MMORPG에서 트래픽이 높게 나타나는 전투 장면에서의 데이터 송신 속도는 다음과 같게 나타났다[6].

[표 3] 송신 데이터 도착 시간 간격(측정시간:20분)

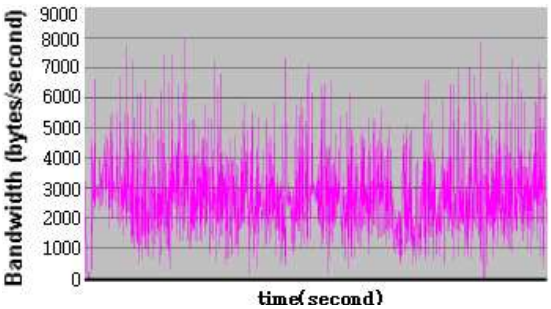
| | 전쟁-적음 | 전쟁-많음 |
|---|---|---|
| 패킷 수 | 7110 | 6892 |
| 평균 | 169 ms | 174 ms |
| 표준편차 | 340 | 340 |
| 최소값 | 0 ms | 0 ms |
| 최대값 | 1292 ms | 2002 ms |

TCP 프로토콜은 MMORPG의 경우처럼 접속자가 늘어나거나 전송 데이터가 증가할수록 데이터 전송 속도도 늘어나게 되어 실시간 온라인 대전게임에 사용하기엔 부적합하다.

#### 2.1.2 P2P 게임의 트래픽 분석

P2P 게임의 데이터 송수신은 TCP 프로토콜과 UDP 프로토콜을 이용한다. 게임 진행 중 많은 트래픽을 요구하는 게임 동기화에 필요한 데이터 송수신을 위해 사용하는 프로토콜이 UDP이다.

[그림 1]은 FPS 게임의 대역폭을 테스트한 결과로 평균 대역폭은 2800bps이다. RTS 게임인 스타크래프트의 경우, 2명이 UDP 프로토콜을 이용하여 게임을 하였을 때 대역폭이 650~1500bps이였으며 플레이어의 수에 따라 대역폭도 선형적으로 증가하였다[7].

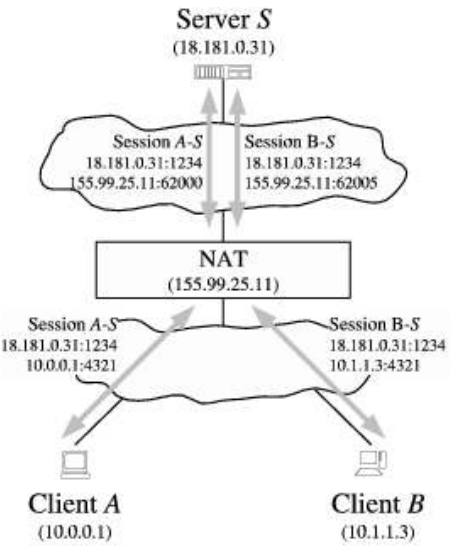



[그림 1] Counter-Strike bandwidth

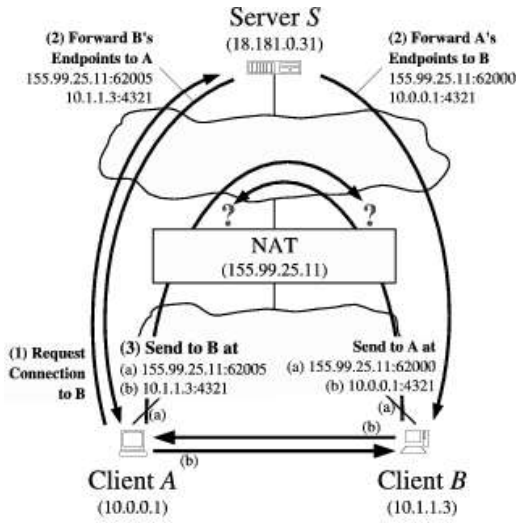
### 2.2 UDP Hole Punching

NAT(Network Address Translation)로 막혀있는 서로 다른 두 호스트 사이에 P2P 통신을 하기

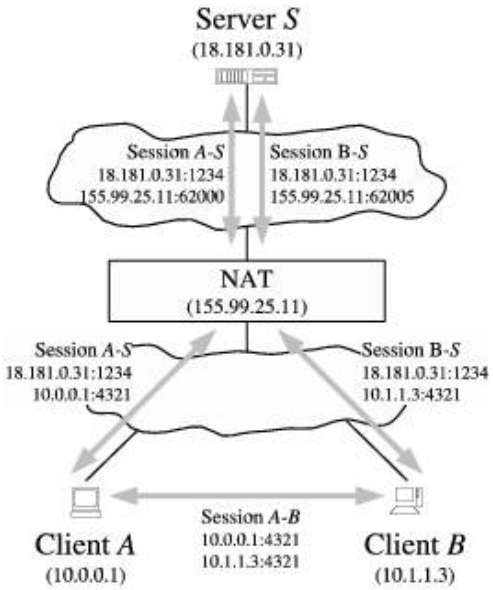
위해 제안된 것으로 UDP 홀 펀칭에 의해서 만들어진 세션을 유지하기 위해 사설망 내부의 호스트가 주기적으로 메시지를 보내어 NAT에서 유지되는 hole 유지 타이머를 리셋하여 사용하는 것이다. 이 방법은 UDP 프로토콜에서만 정상적으로 작동한다. 아래 그림은 UDP 홀 펀칭의 구조를 설명하는 것이다[8].




[그림 2] Before Hole Punching




[그림 3] The Hole Punching Process




[그림 4] After Hole Punching

### 2.3 NAT Traversal

NAT은 하나의 공인된 아이피를 다수의 내부 호스트들이 공유하여 사용하는 기술로 IP 주소와 포트번호의 매핑을 통하여 외부로부터의 세션을 차단하여 방화벽 역할을 하는 것이다. 따라서 NAT 환경에서는 일반적인 P2P, 인터넷게임 서비스 등에 장애요소가 된다.

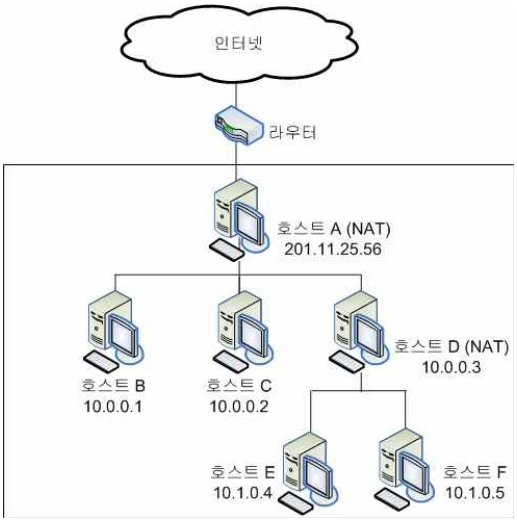
이러한 문제를 일반적으로 NAT Traversal 문제라고 하며, 이 문제를 해결하기 위해 다양한 방법들이 연구되어 왔다[9].

[표 4] NAT의 종류

| 타입 | 방식 |
|---|---|
| Full Cone | 하나의 내부 노드에 대해 NAT에 맵핑되는 외부포 트는 변하지 않지만, 외부에서 내부로 들어오는 패킷의 주소나 포트번호에 따라 제약을 주는 방식 |
| Port Restricted Cone | |
| Restricted Cone | |
| Symmetric | 하나의 내부 노드에 대해 NAT에 맵핑되는 외부 포트번호가 목적지의 주소와 포트번호에 따라 변경되는 방식 |

NAT은 작동 방식에 따라 크게 cone NAT와 symmetric NAT으로 구분되는데, [표 4]는 NAT의 종류와 작동 방식을 보여준다[10,11].

[그림 5]는 NAT을 이용해서 구축한 네트워크 모형이다. 외부에서 접속 가능한 호스트 A의 IP주소를 호스트 B, 호스트 C, 호스트 D 가 공유하여 사용하고, 다시 호스트 D의 IP주소를 호스트 E, 호스트 F가 공유하여 사용하고 있다.



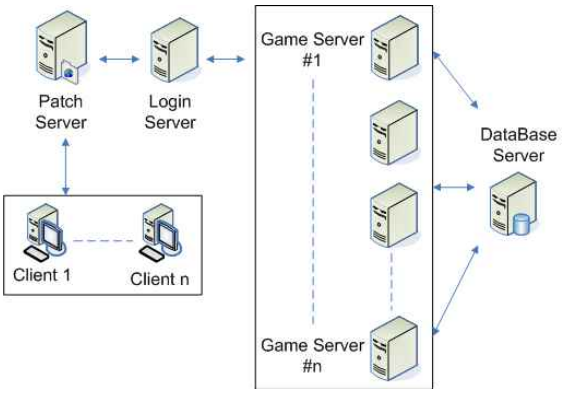

[그림 5] NAT 적용 사례

## 3. UDP 홀 펀칭(Hole Punching)을 적용한 서버/클라이언트 구조

### 3.1 일반적인 서버/클라이언트 구조

온라인 게임의 서비스 방식은 일반적으로 [그림 6]과 같다. 게임의 장르나 구현 방식에 따라서 [그림 6]에 있는 서버를 목적에 따라 분리해서 사용하기도 하고 채팅서버와 같은 별도의 역할을 담당하는 서버를 추가로 구성해서 사용하기도 한다.

전형적인 온라인 게임 서버 구조에서 게임 클라이언트와 각 게임 서버들 간의 데이터 통신은 TCP 프로토콜을 사용하는데, 온라인 대전 게임과 같이 빠른 데이터 송수신이 필요한 장르의 게임 제작에는 적합하지 않다.




[그림 6] 온라인 게임의 전형적인 서버구조

본 논문에 제안하는 게임 서버의 구조는 전형적인 온라인 게임 서버의 구조를 바탕으로 [표 2]에서 언급한 송수신 데이터의 종류와 적절한 프로토콜 분류를 기준으로, 서버와 클라이언트가 주고받는 데이터의 종류와 목적에 맞게 TCP 프로토콜과 UDP 프로토콜을 혼용하여 사용하기 위한 것이다.

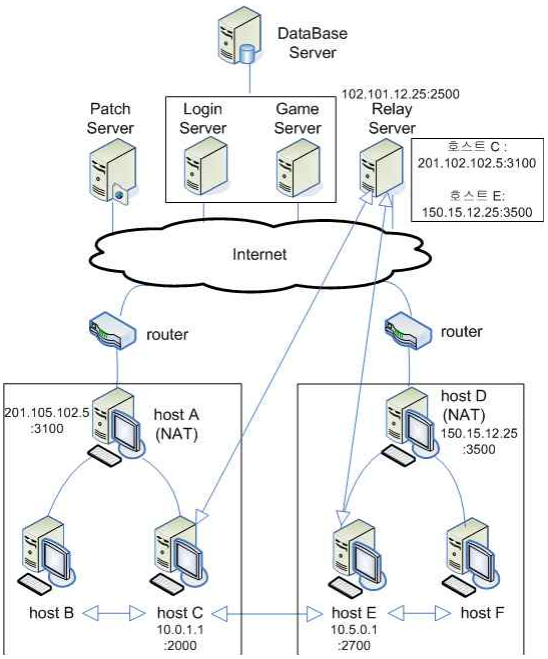
### 3.2 홀 펀칭 기반의 서버/클라이언트 구조

[그림 7]은 본 논문에서 제안하는 서버/클라이언트 구조이다. TCP와 UDP 프로토콜을 병행하여 사용하였으며 P2P 기반의 온라인 대전게임에 사용하기 위해 홀 펀칭 방법을 적용하였다.

중계서버(Relay Server)는 NAT 환경 내부에 있는 클라이언트들로 하여금 자신에게 UDP 데이터를 전송하게 하여 클라이언트가 맵핑되어 있는 NAT의 아이피 주소와 포트번호를 알아내는 역할을 담당하고, 특정 NAT 환경에서 P2P 방식으로 데이터를 보낼 수 없을 경우, 데이터 전송을 중계해주는 역할을 하게 된다.

온라인 대전게임의 일반적인 사용절차는 인증을 거쳐서 게임에 접속하고, 가상의 방(Game Room)을 만들어서 같은 방에 접속한 유저들 끼리 게임을 플레이하는 방식이다. 여기서 UDP 프로토콜 기반의 P2P 통신이 필요한 시점은 게임 플레이가

시작되는 시점부터이다.



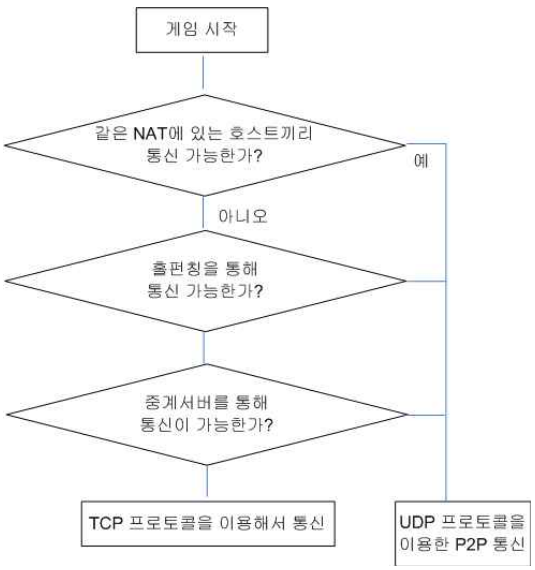

[그림 7] 홀 펀칭을 이용한 서버/클라이언트 구조

홀 펀칭 방법을 이용해서 P2P 통신을 하기위한 방법은 다음과 같다. 호스트 C에서 게임 클라이언트를 실행할 경우, 호스트 C는 중계서버 102.101.12.25:2500에게 UDP 프로토콜을 이용해서 데이터를 전송하게 된다. 중계서버는 데이터를 전송한 호스트 C의 외부 아이피 주소와 포트번호인 201.105.102.5:3100을 이용하여 호스트 C에게 자신의 외부 아이피 주소와 포트번호를 알려준다. 이러한 과정을 거친 후, 호스트 C는 로그인 서버를 통하여 인증과정을 거쳐 게임서버에 접속을 하게 되고, 게임 서버와 접속한 후, 자신의 외부 아이피 주소와 포트번호 201.105.102.5:3100을 게임 서버에 전송하여 게임 서버가 호스트 C로부터 받은 외부 아이피 주소와 포트번호를 가지고 있게 한다. 호스트 E도 호스트 C와 같은 방법으로 게임 서버에 접속하게 된다.

결국, 호스트 C와 호스트 E가 게임 서버에 접속을 하게 되면 게임 서버는 호스트 C와 호스트 E의 외부 아이피 주소 및 포트번호를 가지고 있게 된다. 이러한 과정을 거쳐 게임 서버가 게임 서버에 접속 하는 모든 클라이언트의 외부 아이피 주소와 포트번호를 특정한 자료구조 형태로 관리하게 한다.

호스트 C와 호스트 E가 가상의 방에서 같이 게임을 하게 되면, 게임 서버는 호스트 C의 외부 아이피 주소와 포트번호를 호스트 E에게 알려주고, 호스트 E의 외부 아이피 주소와 포트번호를 호스트 C에게 알려주게 된다. 게임 서버로부터 상대방의 외부 아이피 주소와 포트번호를 받은 호스트 C와 호스트 E는 상대방 아이피 주소와 포트번호를 이용하여 직접 P2P 통신을 하게 되는 것이다.

같은 방에서 게임을 하는 사용자들 끼리 서로 P2P 통신이 가능하려면 게임 시작 전에 P2P 통신을 하기 위한 상태를 설정해야 한다. 상태설정은 몇 가지 단계로 이루어지는데, 게임을 하는 사용자의 컴퓨터가 속한 네트워크가 어떻게 구성되어 있는지의 여부를 체크해서 적절한 P2P 통신 방법을 찾는 것이다.

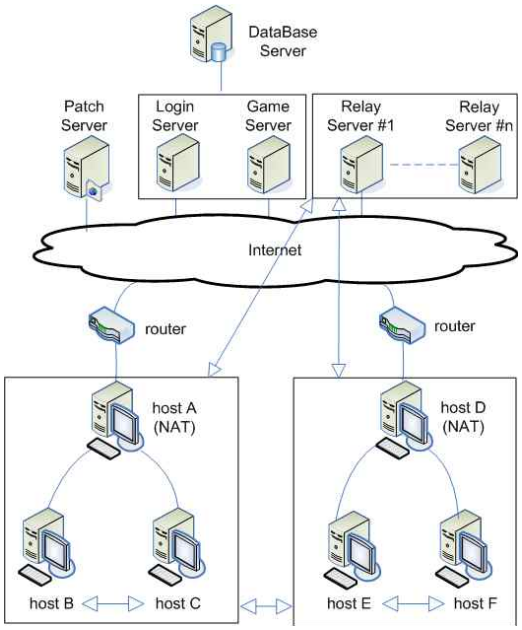



[그림 8] 홀 펀칭을 이용한 P2P 통신 순서

이러한 과정은 게임 플레이를 위해 필요한 게임

리소스가 메모리로 로드되는 과정 중에 이루어지므로 사용자들은 그 과정을 확인 할 수 없게 된다. [그림 8]은 게임 클라이언트들 간에 최적의 P2P 통신 방법을 찾는 단계를 보여준다.

이처럼 P2P 통신을 위하여 여러 가지 단계를 두는 이유는 통신 속도나 처리 알고리즘이 간단한 방법에 우선순위를 두기 위해서이다.




[그림 9] 다중 중계서버를 이용한 홀 펀칭 구조

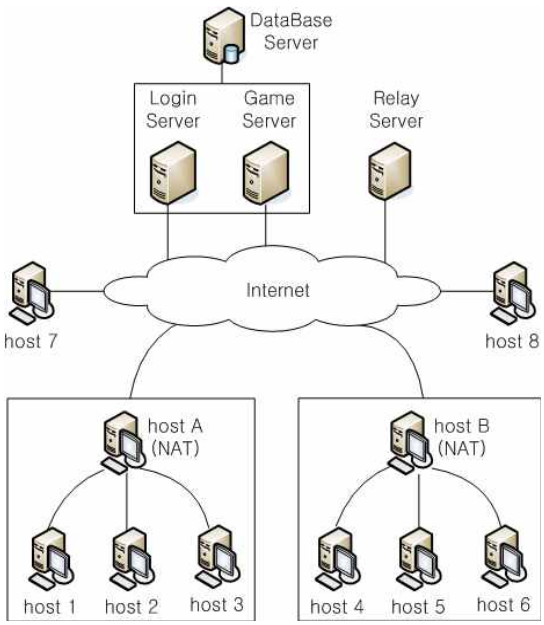
이러한 방법이 적용된 구조는 오랜 시간 데이터의 송수신이 없을 경우에 NAT에 맵핑되어 있는 클라이언트의 포트번호가 닫히는 경우가 발생 하는데, 일정시간마다 반복적으로 데이터를 전송하여 포트번호가 닫히는 것을 방지해야 한다.

게임 사용자가 늘어 중계서버에 부하가 걸릴 경우를 대비해서 중계서버를 여러 개로 구성하고, 사용자들이 접속하는 시점에 하나의 중계서버를 선택하도록 하면 중계서버 부하로 인한 통신 속도 지연을 막을 수 있다. [그림 9]는 이러한 구조를 표현한 것이다.

## 4. 성능 평가

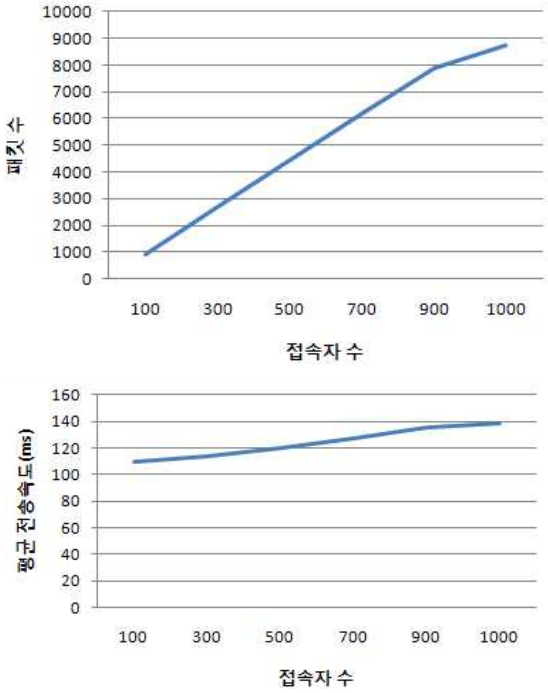
본 논문에서 제시한 구조의 성능평가를 위해 2가지 방법으로 테스트를 진행하였다. 하나는 접속자수 증가에 따른 패킷 전송 속도를 측정하였으며, 다른 하나는 패킷 손실률을 측정하였다.

테스트 환경은 [그림 10]과 같이 일반적으로 게임을 서비스하기위한 네트워크 환경을 이용하였다. 2개의 서로 다른 네트워크 그룹(host1~6)은 NAT 환경을 이용하고, 2개의 호스트(host7~8)는 NAT 환경을 이용하지 않았다.




[그림 10] NAT을 이용한 테스트 환경

먼저, 패킷 전송속도 측정하기 위해 최대 1000개의 가상의 클라이언트를 사용하였다. 가상의 클라이언트는 게임서버에 접속하여 일반적으로 게임 내에서 발생할 수 있는 데이터(방이동, 상점정보, 채팅)들을 서버로 전송하게 하였다. 가상 클라이언트의 80%는 클라이언트들이 게임 진행에 필요한 UDP 데이터들을 송/수신하게 하였다.




[그림 11]접속자 수에 따른 패킷 수와 전송속도

[그림 11]은 접속자 수에 따른 패킷 전송량과 패킷의 평균속도를 측정한 결과이다. 접속자 수에 따라서 증가한 패킷 수에 비해 패킷 전송 속도는 크게 증가하지 않은 결과를 볼 수 있다. 대전 중에 발생하는 게임 데이터의 경우 P2P 방식으로 전송되기 때문에 접속자 수의 증가에 크게 영향을 받지 않게 된다.

다음으로 패킷의 손실률을 측정하기 위해 서로 다른 크기의 UDP 패킷을 이용하였다. 테스트로 사용한 UDP 패킷의 크기는 22byte, 42byte, 60byte의 3종류로 하였다. 초당 패킷을 50개 전송하였으며, 모든 패킷에는 송신측에서 부여한 일련번호를 같이 전송하게 하고 수신측에서 일련번호를 체크하여 손실률을 계산하였다.

[표 5]는 손실률을 측정한 결과로 초당 패킷의 손실률이 14% 미만으로 나타났다.

[표 5] 평균 패킷 손실률

| 호스트 | 초당 전송 패킷수 | 평균 손실 패킷수 | 손실률 |
|---|---|---|---|
| host1 | 50 | 7 | 14% |
| host2 | 50 | 5 | 10% |
| host3 | 50 | 6 | 12% |
| host4 | 50 | 7 | 14% |
| host5 | 50 | 5 | 10% |
| host6 | 50 | 5 | 10% |
| host7 | 50 | 1 | 2% |
| host8 | 50 | 2 | 4% |

## 5. 결론 및 향후 과제

본 논문에서는 UDP 프로토콜을 이용한 홀 펀칭 방법을 기반으로 실시간 온라인 대전게임 제작을 위해 사용가능한 게임 서버의 구조를 제안하였다.

UDP 프로토콜 기반의 P2P 게임을 제작할 경우에는 기획단계에서부터 송수신 되어야 하는 데이터의 양을 고려한 설계가 이루어져야 할 것이다.

UDP 프로토콜을 기반으로 하는 게임 진행 방식은 통신의 신뢰성을 보장하지 않는 프로토콜의 특성상 데이터의 손실이 발생하게 된다. 게임에서 UDP 패킷을 사용하는 부분은 [표 2]에서 언급한 것처럼 게임 진행에 크게 중요한 데이터 보다는 전송 속도에 민감한 데이터들이다. 보통 이러한 이유 때문에 실시간 온라인 대전 게임들은 TCP 프로토콜과 UDP 프로토콜을 같이 사용하게 된다. 짧은 시간에 가끔 나타나는 이러한 UDP 패킷 손실에 대한 부분은 보간 방법들을 이용하여 보완하면 게임에서 동기화를 맞출 수 있을 것이다.

앞으로 이러한 데이터의 손실을 보간 할 수 있는 알고리즘에 대한 연구가 더 진행 되어야 할 것이다.

## 참고문헌


[1] Kuan-Ta Chen, Polly Huang, Chin-Laung Lei. “How sensitive are network quality?” Communications of the ACM, 2006.

[2] (재)한국소프트웨어 진흥원, 2006년 해외 디지털콘텐츠 시장조사, 2006.

[3] 김선우, 윈도우 네트워크 프로그래밍, 한빛미디어, 2005.

[4] Tristan Henderson, Saleem Bahatti, “Networked games-a Qos-sensitive application for QoS-isensitive user?”, Proceedings of the ACM SIGCOMM workshop on Revisition IP QoS, 2003.

[5] 임수정, 홍동철, 김수성, 김성주, 유행석, 한준탁, 장태무, “3D MMORPG 온라인 게임에서의 네트워크 분석”, 한국게임학회 총회 및 학술발표대회, pp289-293, 2004.

[6] 박효주, 김태용, “MMORPG 트래픽 분석과 모델링”, 멀티미디어학회, 9권, 5호, pp635-648, 2006.

[7] Mark Claypool, David LaPoint, and Josh Winslow, “Network analysis of Counter-strike and Starcraft”, Conference Proceedings of the 2003 IEEE International, 2003.

[8] B. Ford and P. Srisuresh. “Peer-to-peer communication across network address translators,” USENIX Annual Technical Conference, 2005.

[9] Newport networks, http://www.newport-networks.com/cust-docs/33-NAT-Traversal.pdf

[10] P. Francis, R. Gummadi, IPNL: A NAT-extended Internet architecture. In ACM SIGCOMM, August 2002.

[11] 한동훈, 온라인 게임서버 프로그래밍, 정보문화사, pp.287-294, 2007.

주 우 석 (Ju, Woo Suk)

동서대학교 디지털콘텐츠학부 게임전공 전임강사

관심분야 : 컴퓨터 그래픽스, 네트워크, 온라인 게임