

MMORPG 에서 좁은 지역에 유저가 N 명이 모여 있을 때 처리 방법

(출처 : <http://www.npteam.net/855>)

1. 기획적 의도로 해결하는 방법

- 국지전이 일어나는 장소를 여러 곳으로 분할한다.
- 국지전에 참여할 수 있는 최대 인원수를 제한한다.

2. 프로그램적으로 해결하는 방법

• 영향을 미치는 요소들

- 데이터 크기
- 데이터를 보내는 빈도 수
- 전송 특성
 - 1:1 패킷 보내기
 - 1:N 패킷 보내기
 - N:M 패킷 보내기

• 해결방법

- 데이터 크기
 - 패킷 데이터의 크기를 최대한 줄인다.
 - 패킷에 포함된 기본 정보 단위를 최소화 한다.
(자료형 단위를 줄이는 방법, 불필요한 정보를 제거하는 방법)
 - 패킷 데이터를 압축한다.
 - 패킷 데이터를 압축/해제시 CPU 자원을 사용한다.
 - 브로드캐스팅할 경우 패킷이 작아지면 트래픽이 줄어드는 잇점이 있다.
- 데이터를 보내는 빈도 수
 - 최대 빈도수를 정한다.
 - 클라이언트 응답 속도 250ms 일 경우 1 초에 4 번
 - 클라이언트 응답 속도 200ms 일 경우 1 초에 5 번
(이동 패킷과 스킬 사용 패킷을 1 초에 4-5 번으로 해결하기 어렵다.)
 - 우선순위 큐를 이용한 패킷 전송
 - 전투 관련 패킷을 우선적으로 처리한다.
 - 비우선 패킷들 : 로그, 채팅, 환경 오브젝트 처리
 - 이미 전송한 명령은 Send 버퍼 Queue 에서 삭제한다.
 - 동일한 좌표로 계속 이동하는 패킷은 클라이언트에서 한번만 보내도 된다.

- 필요 없는 패킷을 제거하여, 트래픽도 줄이고 서버의 CPU 자원 사용도 줄일 수 있다.
- 전송 특성
 - 1:1 패킷 보내기
 - 문제점 없음
 - 1:N 패킷 보내기
 - Send 패킷 버퍼 재활용 필요함.
 - N:M 패킷 보내기
 - 패킷 압축 필요함.
 - Send 패킷 버퍼 재활용 필요함.
 - 브로드캐스팅을 게임 서버에서 하지 않고, Front 혹은 Agent 서버로 패스하여 처리한다.
(UDP SuperPeer 를 사용할 경우 클라이언트에서 브로드 캐스팅하는 것도 고려해 볼 수 있다.)

3. 그외에 더 생각해 볼 사항

• CPU 자원

- 메모리간 데이터 복사를 최소화 한다.
 - C++0x 의 std::move 를 사용하여 데이터 값 복사를 줄인다.
 - Reference Count 기능을 가진 패킷 컨테이너를 사용한다.
- 메모리간 데이터 복사를 빠르게 한다.
 - FastMemcpy 와 같이 CPU 의 기본 레지스터 이외의 MMX 와 같은 크기가 큰 레지스터에 데이터를 unrolling 하여,
Loop 를 이용한 데이터 복사 속도 저하를 개선한다.

• 네트워크 자원

- 보장이 필요한 패킷과 보장이 필요 없는 패킷으로 구분하여 처리한다.
 - 보장이 필요한 패킷 : 주기적인 시간 간격으로 동기화에 필요한 중요한 정보
예) X, Y, Z, 좌표 + Angle
 - 보장이 필요 없는 패킷 : 주기적인 시간 간격보다 짧은 간격이면서,
손실되어도 문제 없는 정보
예) Angle 정보
- 보장이 필요한 패킷은 TCP 로 보내고, 보장이 필요 없는 패킷은 UDP 로 나누어 처리할 수 있다.

(단, 중국의 경우 UDP 가 불가능한 상황도 발생한다.)

- **브로드 캐스팅을 주체하는 서버를 구분한다.**

- Game 서버가 모든 패킷을 브로드캐스팅 하면, Front 혹은 Agent 서버에 N 번의 요청을 전달한다.
- 브로드캐스팅이 필요한 패킷을 각 Front 혹은 Agent 서버에 1 회만 보내고(받을 클라이언트 목록을 함께), Front 혹은 Agent 서버에서 처리한다면, Game 서버에서 N 번 요청에 대한 CPU 자원 사용을 줄이고, 내부 네트워크 트래픽도 감소한다.

- **유저 상태에 대한 패킷 처리**

- 유저가 전투 상태일 경우 패킷을 먼저 처리하고, 비 전투 상태일 경우 나중에 처리하도록 한다.
예) 마을에 모인 유저인 경우, 패킷이 늦게 처리되어도 문제가 발생하지 않는다.