Reliable UDP 구현

김성익

2005.03.24

Introduction

- 게임에서의 네트워크 프로그래밍
- 네트워크 게임의 한계와 제약
  대역폭, 지연시간(latency)

- 소규모 방 개념의 게임 스타일
- Peer2Peer 게임 추세
- 운영비 절감

이슈

- TCP와 비교한 UDP의 특성
- UDP의 활용한 RUDP의 구현 이슈들
- NAT 환경과 UDP연결 이슈들

TCP 와 UDP의 차이점(1)

- 비 연결 지향적이다
  TCP는 연결 지향적이다
- 데이터의 수신여부를 검사하지 않는다
  TCP는 반드시 도착한다고 가정하며, 수신이 안된 경우 다시 보낸다.
- 보낸 패킷의 단위로 도착한다
  TCP는 연속된 데이터 형태로 전송하며, 단위는 임의로 정해진다.

TCP 와 UDP의 차이점 (2)

- 보낸 순서와 상관없이 도착한다
- 도착하지 않을 수 있다
- 내용이 손상되어 도착할 수 있다
- 여러 번 도착할 수 있다
  TCP로 전송된 데이터는 보낸 스트림 그대로라고 신뢰할 수 있다

- 패킷 크기 제한

UDP의 특성

- 하나의 포트로 여러 클라이언트와 통신 가능하다.

- 전송 확인 절차가 없으므로 빠르다.(*)
- 데이터 전송의 신뢰도가 떨어진다.

- 데이터의 손실이 있어도 무관한 정보, 빠른 전송이 필요한 정보에 적합

Reliable UDP 구현

- UDP를 이용해서 TCP처럼 Reliable한 데이터 교환을 구현
  전송 보장 + 패킷이 순서대로 도착처리

- 연결 지향적인 특성을 가진다.
- 데이터의 수신 여부와 순서를 보정한다.

RUDP 특징(1)

- 전송 패킷을 3단계 타입으로 구분
- 1. UDP고유의 전송 보장 안 되는 패킷
  UDP 고유의 빠른 반응 처리
- 2. 전송은 보장하지만, 받는 순서는 상관없는 패킷 (*)
  TCP는 모든 패킷의 순서를 보장하기 때문에 중간에 패킷 loss 가 생긴 경우 모든 패킷이 딜레이 되지만, 순서 상관없는 패킷 형태는 앞 뒤 패킷의 영향을 받지 않으므로 다른 패킷으로 인한 딜레이가 없다.

RUDP 특징(2)

- 3. 전송도 보장하고, 보낸 순서대로 받는 패킷
  TCP와 동일한 모든 패킷의 순서를

연결 지향적 특성

- 기본 ack 신호와 sequence 정보를 주고 받기 위해 상대와의 전송 상태 정보 필요

- recvfrom으로 전송 받을 때 처음 데이터가 전송되거나, 특정 패킷을 보낼 때 연결 정보를 생성

패킷구성(1)

- 패킷의 전송 타입
  Reliable 합니까 ? Sequential 합니까 ?
  Acknowledge 패킷 혹은 ping 패킷입니까 ?

- 패킷의 고유 넘버
  수신쪽에서 acknowledge 신호를 보낼 때 사용
  sequential 특성을 가지는 패킷인 경우 수신측에서 순서를 판단하기도 하므로 일정하게 증가

패킷구성(2)

- 에러검출 코드
  수신된 데이터가 올바른지 판단하기 위한 checksum 혹은 crc 코드

- 패킷 데이터
  실제 전송하고자 하는 데이터 및 크기

1. 데이터 변형 검출

- UDP 데이터 자체가 변형된 상태일 수도 있으므로 패킷의 에러검출 코드를 이용하여 패킷이 올바른 정보인지 판단

2. 데이터 수신 여부 판단

- Reliable 한 데이터는 send 할 때 버퍼에 보관
- 특정 시간동안 상대방으로부터 acknowledge 신호 없으면 다시 보낸다.
  실제로 못 받았거나 acknowledge 신호가 도착 안 한 경우므로 후자인 경우 수신측에 중복 데이타 전송됨
- 수신한 쪽에서는 Reliable 데이터를 수신하면 ack 패킷을 보낸다.
  Ack 패킷도 손실가능하므로, 송신측에서 패킷을 다시 보낸 경우 ack 못 받은 상태므로 다시 보냄

3. 순차적인 수신

- 수신쪽에서 고유번호를 이용해서 패킷을 순서대로 정렬
- 예측된 패킷 번호인 경우 recv 처리, 아닌 경우 (나중 패킷) 버퍼에 임시 보관하였다가, 중간에 빠진 패킷 처리 후 버퍼에서 예측된 패킷 번호의 패킷이 있는지 검사

데이터 송신

- 1. 패킷구성
- 2. 패킷 타입이 도착 보장인 경우 보낸 내용과 고유번호를 send 리스트에 저장
- 3. 보내기

데이타수신(1)

- 1. ack 패킷인 경우 send 리스트에서 찾아서 삭제
- 2. 타입1의 패킷인 경우 recv listener 호출
- 3. 도착 보장(타입 2, 3)인 경우 ack 패킷 보냄
- 3b. 이전에 처리된 패킷이면 ack 패킷만 보내고 리턴

데이타수신(2)

- 4. 타입 2인 경우 recv listener 호출
- 5. 순서 보장(타입 3)인 경우
- 5a. 패킷 고유값과 (최근 고유값 + 1) 이 같으면 recv listener 호출, recv 리스트에서 다음 패킷이 있으면 recv listener 호출 후 recv 리스트에서 삭제
- 5b. 다르면 recv 리스트에 저장

기타

- 주기적으로 send 리스트를 검색해서 ack 신호 못 받은 패킷은 재 전송

- 주기적으로 데이터 전송이 없는 연결은 ping/pong 패킷을 보내서 연결확인
  TCP와의 다른 점은 ?
- 패킷 암호화
- Nagle's Algolithm 적용

RUDP사용의 장점

- TCP 에 비해서 더 넓은 네트웍 환경에서 Peer 간 연결이 가능하다.

NAT 환경에서의 연결(1)

- 내부망에 포트N으로 소켓을 생성한 경우 라우터의 포트M과 매핑되어 외부에서는 포트M으로 나감
- 내부 IP는 라우터를 통해서 외부로 나갈 때는 외부 IP사용

- 내부에서는 매핑된 포트와 외부 IP알 수 없다.

NAT 환경에서의 연결(2)

- 하지만, 외부 서버에 udp로 패킷을 보낼 경우 외부 서버에서 외부에 노출되는 ip와 port을 detect 할 수 있다.

UDP 연결 제약

- 1. 모든 UDP 연결 허용
- 2. (방화벽) 임의의 네트웍 Peer에서 UDP 포트로 패킷 보내기 허용
- 3. (방화벽) UDP 포트에서 외부로 메시지를 보낸 적이 있는 네트웍 Peer에 대해서만 해당 UDP 포트로 패킷 보내기 허용

  3번의 특성을 가진 피어 끼리는 연결불가, 터널링 필요