Lab 14

Part A

Latency is the measure in milliseconds of the time it takes for the command to be sent and the data request to be serviced and returened to the user, or the Round Trip Time as it is more properly known.

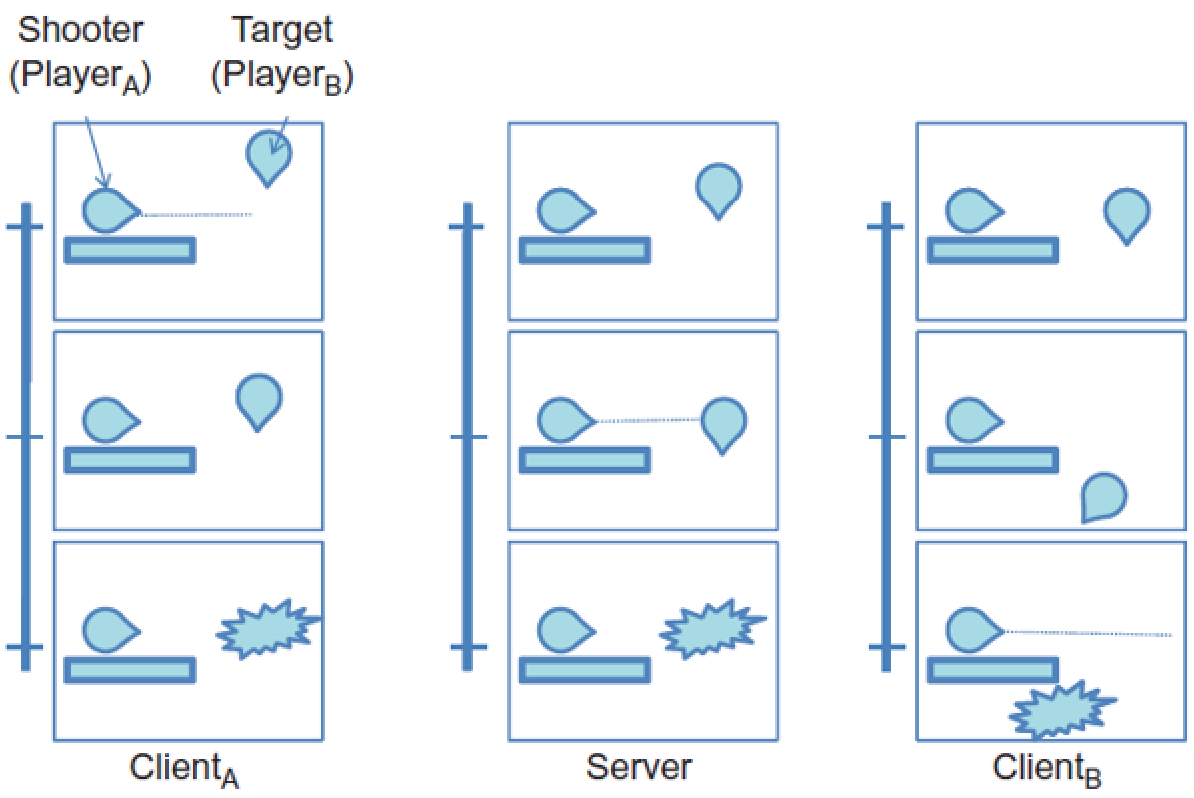
Latency should always be minimised, but although data moves around at the speed of light on fibre optic enabled networks, the distance between the user and the server, along with delays introduced by internet infrastructure equipment, mean latency can never be eliminated completely. There are also propagation delays due to speed of communications link and speed of light delays.

Queuing and processing at routers and on source and destination nodes.

Transcodingh delays due to data manipulation.

Part B

The latency of different players can affect how the result of a gunfight can appear on different players screens. If player 1 has latency and can see player 2 standing in front of them they may try to shoot at the player. However, on players 2 screen they may have already moved around a corner so that they are no longer in the line of sight of player 1 so from their perspective they should not be shot. The result of if player 2 is actually shot or not depends on when each of the players information is sent to the server. If player 2’s position is not sent to the server in time they may still be in the position that player 1 can see them in when player 1 shoots meaning that they will be shot according to the server as their movement has not been fully sent over yet. There is also a chance that player 1 shooting may take extra time to send over to the server instead of player 2’s position information depending on their internet speed as well as how close to the server they are which will instead result in player 1 thinking that they shot player 2 but in reality the server will have registered player 2 moving first and will take extra time to see that player 1 tried to shoot.



Part C

Pessimistic (Conservative)

Assume that all changes to the game world will result in inconsistency, so impose strict limitations on how updates are applied. There is a global order which means that events are processed in the same order.

Optimistic

Assume that a small number of changes to the game world will result in incostiency, so impose loose limitations on how updates are applied. If issues occur, fix them at a later time.

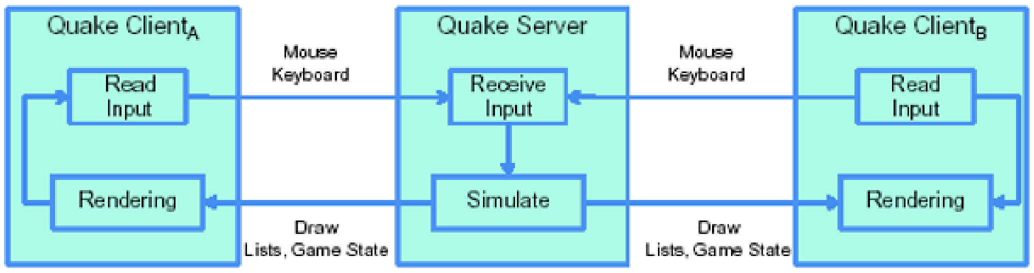
Dumb Client

Client sends inputs to the server.

Server calculates and sends to the client.

Client just renders the state sent to it from the server.

This is part of the conservative group as the server handles all of the events.



Client side prediction

First introduced in the QuakeWorld add on the Quake 1.

- Send out inputs to server, but also apply them immediately locally to update your own state.
- Server can override player input if simulation on server side has a constraint player doesn't know about yet.
- When you receive update from the server, check if it differs from local state.
- If different, correct local state.

This is an optimistic approach as all of the clients also run the game on their side until they are told by the server that they need to update smoething in case they had incorrect information about something. This helps to improve the responsiveness of the game but it can slightly increase the differences between each client until the server gives them the correct information.