SWINBURNE UNIVERSITY OF TECHNOLOGY

# Doing battle with the real world: How online game developers try to defeat the laws of physics

Associate Professor Grenville Armitage

Centre for Advanced Internet Architectures

## Online Games – what are they?

- Virtual worlds: Many people playing roles beyond their day to day experience
- Entertainment, escapism, community… many reasons

World of Warcraft

http://www.worldofwarcraft.com/info/underdev/images/azuregos.jpg

Second Life

http://blogs.zdnet.com/images/secondlife1.jpg

Quake 4

http://www.techspot.com/reviews/games/quake4_preview/building_b_marines06.jpg



## (Some) Types of games

- FPS: First Person Shooter
  - 3D virtual environment from 1st person visual perspective
  - Primarily combat based (team or individual)
- RTS: Real time strategy
  - E.g. Control a large number of troops fighting enemy players
- RPG: Role playing game
  - Play a detailed character solving quests and interacting with other player and non-player characters
- MMORPG: Massively multiplayer online RPG
  - Thousands of players (or more) interact simultaneously in the game world



## First person shooter (…first person view)

Most often these are fast-paced and frantic 'twitch games'

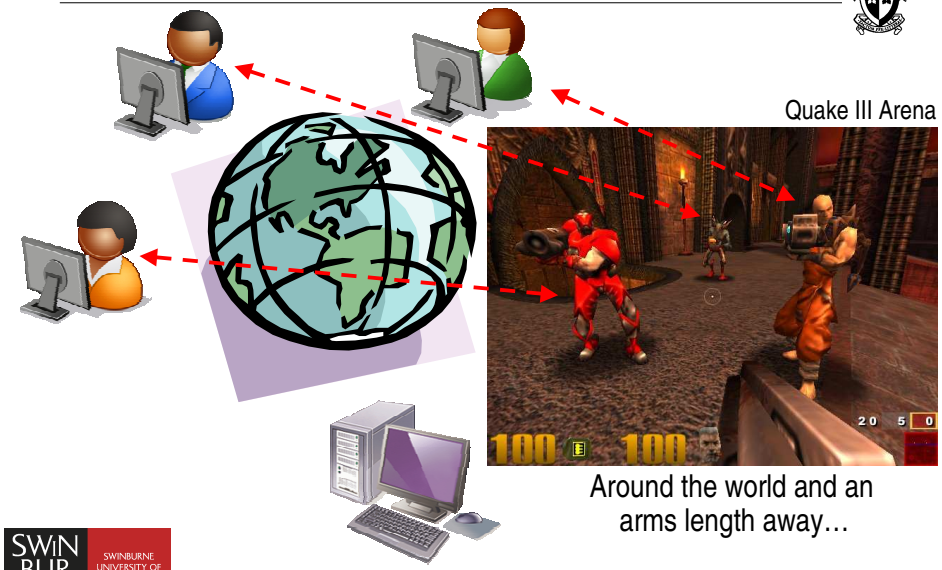
Unreal Tournament 2007

http://grispemmix.googlepages.com/u5.jpg/u5-full.jpg

Wolfenstein Enemy Territory

(Also Counterstrike Source, Quake 4, Halo/Halo2 on Xbox Live, etc….)

# Immersion – illusion of presence

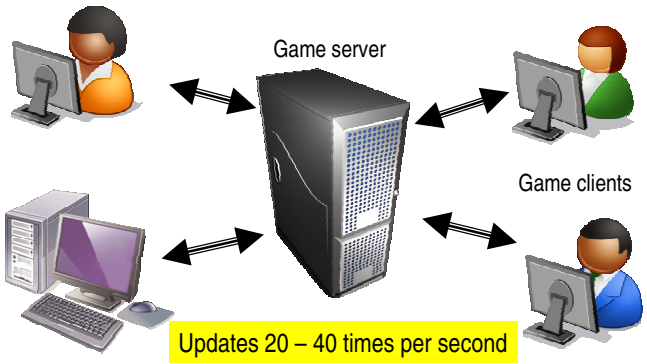





# Achieving a common, shared reality

- *Shared reality*
  - What one player perceives, all players must perceive
- A central *game server* acts as umpire to *game clients*

Each game client continuously tells the game server what their player is doing, and hears what other players have done



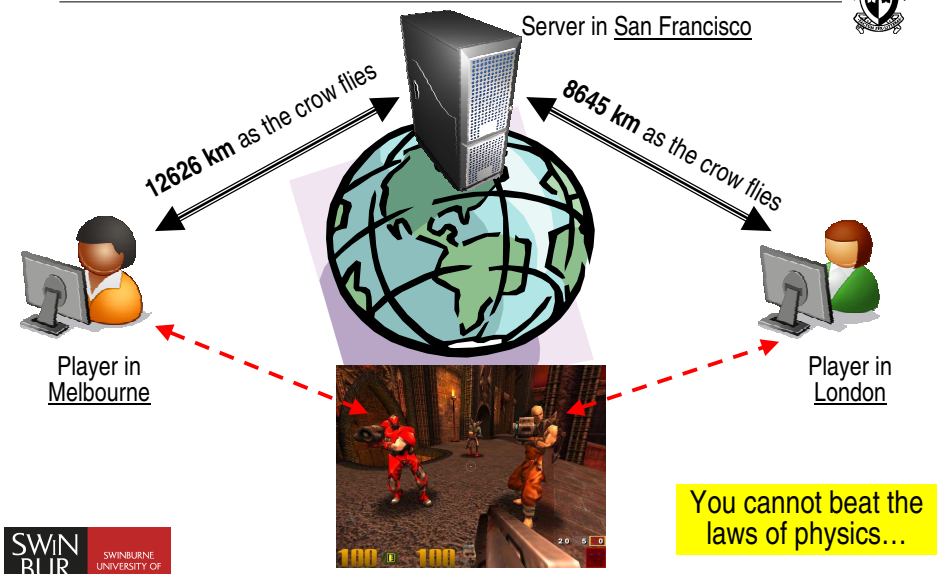



# Game server as umpire

- Every virtual world has its own 'laws'
  - How fast players can move, shoot, reload…
  - How well players can aim, see, recover their health…
- Would you trust your opponent to decide if they hit you?
- The game server calculates every move
  - A *sanity check* on what a client claims its player is doing
  - *Arbitrates* when two or more players attempt something and only one can succeed (e.g. picking up scarce ammunition)
  - Every event goes through the game server



# And then *real* reality gets in the way



# Distance… Equals delay…

- We typically measure delay in milliseconds (ms)
  - Round trip time (RTT) is combined delay in both directions, sometimes also known as *ping*, *lag* or *latency*
- Speed of light in air ~299792km/sec
  - Melbourne to San Francisco: 12626km path -> ~42ms (RTT ~84ms)
  - London to San Francisco: 8645km path -> ~ 28ms (RTT ~56ms)
- Internet rarely goes "as the crow flies" (usually a longer, convoluted physical path which equates to higher delay)
- Speed of light is ~30% slower in optical fiber (which makes the path seem longer - even more delay)



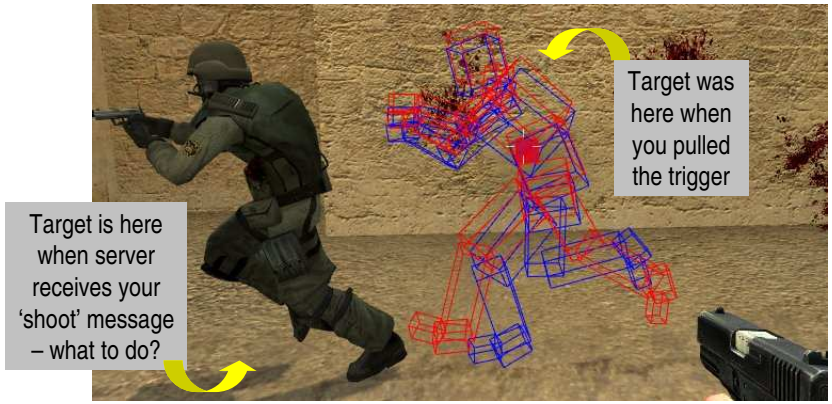
# Does delay matter?

- Latency introduces *timeline discrepancy* between each client's view of the virtual world
  - Game server: 'official' view of events in timeline
  - Game clients: at varying 'distances' from server
  - Everyone has slightly different perception of *when* events happen (relative to each other)
- Latency affects reaction times
  - And may be hard to adapt to variation in delays



# Did you hit or miss?

- 100ms is a long time when your target is running

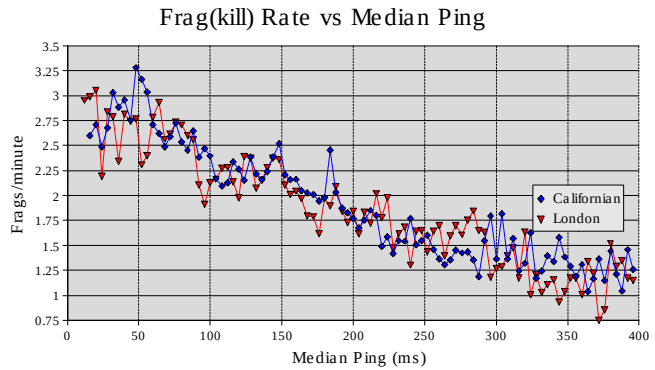



http://developer.valvesoftware.com/wiki/Source_Multiplayer_Networking



# FPS 'success' vs delay

- Success (measured in 'frags' per minute) seen to decay with increasing network delay
  - E.g. Experimental trials in 2001, Quake III Arena online FPS, two servers – one in California, one in London




G.J. Armitage, "An Experimental Estimation of Latency Sensitivity In Multiplayer Quake 3", 11th IEEE International Conference on Networks (ICON 2003), Sydney, Australia, September 2003

## Where does the delay come from?

- Aside from physical distance vs speed of light, there is:
  - *Serialisation* delay
    - Finite time to send a packet bit-by-bit (e.g. dial-up, DSL or optical)
    - E.g. A 128Kbit/sec link takes ~94ms to transmit 1500 byte packet
  - *Queuing* delay
    - Packets can be temporarily delayed as they criss-cross the Internet
    - (Like people queuing at different airports while traveling)
  - Your home broadband router, your local/domestic ISP, and International Internet links



## Broadband, ISPs and convoluted paths

- Trivia: On 23Aug07 the path between a home ADSL link in Melbourne to a San Francisco area website:
  - Home LAN to local ISP over ADSL: ~16ms to RTT
    - (mostly my ADSL modem)
  - Melbourne to Sydney inside ISP: ~15ms to RTT
  - Sydney to Los Angeles: ~155ms to RTT
    - International link
  - Los Angeles to San Francisco: ~13ms to RTT
- Total of ~198ms RTT (using 64-byte packets)



## Solutions from game developers

- Use smallest packets possible in each direction
  - Minimise serialisation delays
    - 50 – 80 bytes client to server, 80 to ~250+ bytes server to client
- Client-side prediction
  - For smoother graphical experience
- Server-side roll-back
  - Because only the server is trusted to decide crucial events



## Predicting the future (client-side)

- Every client is rendering a real-time, graphical view of the player's virtual world
  - The player must see the consequences of their own actions, and the actions of others, in real-time
- Imagine if a client's command (move, shoot, etc) went to the server for confirmation <u>before</u> the consequences are shown on screen?
  - Perhaps okay on a LAN (with < 10ms delay)
  - Shockingly jerky experience if you're on an Internet server

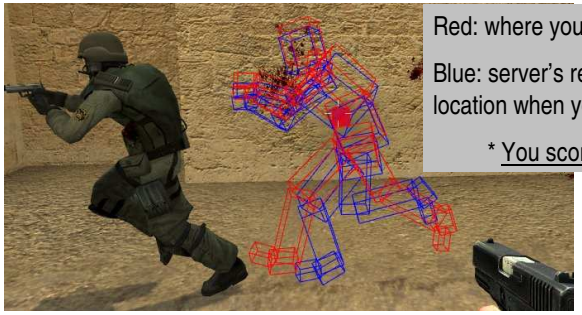
# Predicting the future #2 (client-side)

- Instead, client does two things at once:
  - Renders the <u>most likely</u> consequence of a local movement command (prediction)
  - Sends command to server
- Approximately one RTT later, server updates everyone
  - Client now knows the *official* consequences of its previous action (and any additional actions of others)
  - If previous prediction was accurate, continue smoothly
  - If prediction was wrong, re-render the scene ('snap-back')



# Remembering the past (server-side)

- Game server 'remembers' recent past events
- Rolls back time (internally) to decide where all players were when you 'pulled the trigger'



Red: where you saw your target

Blue: server's reconstruction of target location when you pulled the trigger

* <u>You scored a hit after all!</u> *

http://developer.valvesoftware.com/wiki/Source_Multiplayer_Networking



# Unavoidable consequences

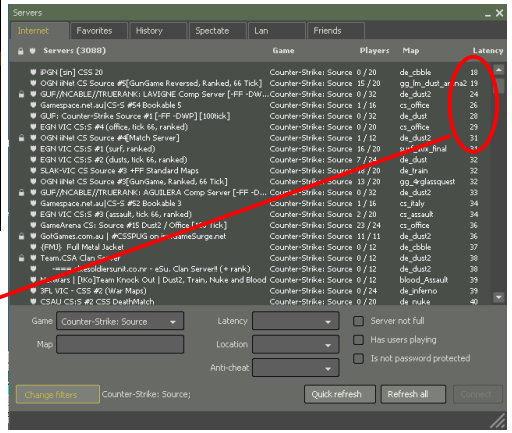
- Reality is unavoidable in multiplayer games
- Communication delays introduce tiny timeline errors, most evident in fast-paced 'twitch' games
- Clients and servers can smooth out, but never entirely eliminate, the consequences of communication delay



# Game players – pick closer servers!


Quake III Arena server browser

Counter-Strike:Source server browser




Estimated RTT to available game servers

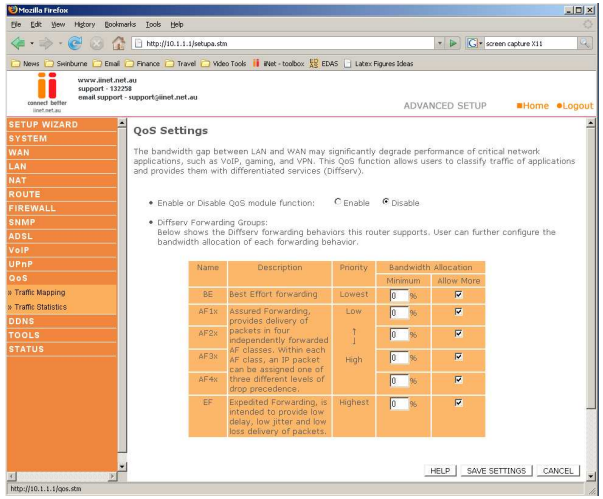
# Solutions from network engineers

- Engineers utilise various approaches to reduce delays
- Increase bandwidths
  - Reduce serialisation delays
- Optimise routes (paths)
  - Shorter physical paths, avoid places where queuing occurs
- Establish 'priority' traffic handling
  - Where queues occur, try to allow game or VoIP (voice over IP) traffic to 'jump the queue'



# Priority in your broadband router

- Consumer broadband routers often now support priority classes of traffic
- Typically manually configured through QoS (Quality of Service) menus






# And your career relates how?

- Innovation starts with problem solvers … is that you?
- Engineers build <u>systems</u> to produce desirable outcomes
- Whether for entertainment or serious work, virtual worlds are emerging whose participants span the globe
- Telecommunications and electronic engineers are needed to re-engineer the Internet's infrastructure
- Computer software engineers are needed to create more sophisticated & realistic virtual worlds



# So I have a suggestion….

- Come to Swinburne after high school ☺
  - Bachelor of Engineering (Telecommunication and Network Engineering)
  - Bachelor of Engineering (Electronics and Computer Systems)
  - Bachelor of Multimedia (Games and Interactivity)/Bachelor of Science (Computer Science and Software Engineering)
  - Bachelor of Engineering (Telecommunications and Network Engineering)/ Bachelor of Science (Computer Science and Software Engineering)

  - (see the Faculty of ICT website for other combinations…)