Topological Optimisation for Online First Person Shooter Game Server Discovery

G. Armitage, C. Javier, S. Zander

{garmitage,cjavier,szander}@swin.edu.au

http://caia.swin.edu.au

ATNAC 2006

Overview

- Motivation
- Game server discovery mechanism
- Empirical analysis of popular game
- Proposed optimisation
- Conclusions & future work

Motivation

- Multiplayer **First Person Shooter** (FPS) games have become very popular
- Locating **playable** game server is key challenge for player
- Game client **probes** all available game servers and present information to players
  - **Latency/ping**, map, number of players, etc.
- Latency measured as Round Trip Time (RTT) between client and server

SWINBURNE UNIVERSITY OF TECHNOLOGY

Motivation cont'd

- Players usually select 'close' game servers → RTT typically ≤180-200ms
  - G. Armitage, "Sensitivity of Quake3 Players To Network Latency," SIGCOMM Internet Measurement Workshop 2001
- Probe traffic generated is significant → ~8GB measured at a local game server over 20 weeks
  - S. Zander, D.Kennedy, G. Armitage, "Dissecting Server-Discovery Traffic Patterns Generated By Multiplayer First Person Shooter games", ACM NetGames 2005

➢ Optimise server discovery: probe closer servers first

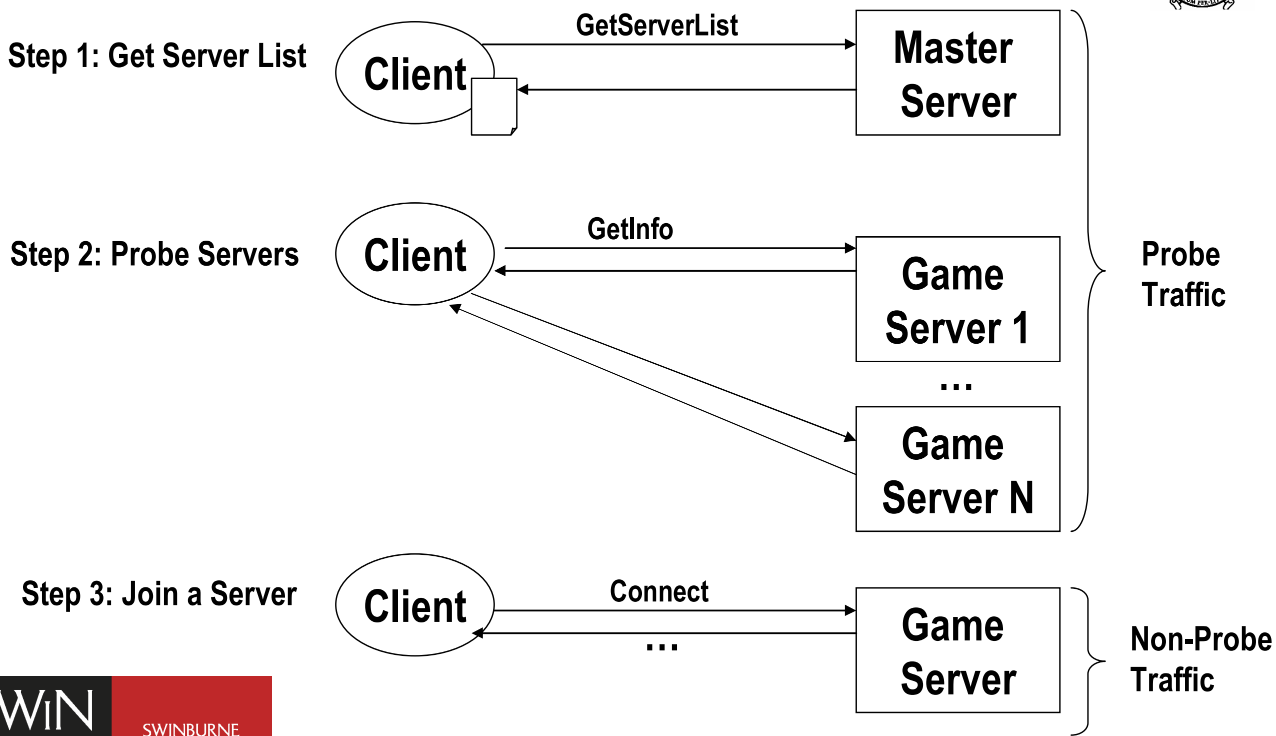
- ➢ Minimise time it takes user to find playable server
- ➢ Reduce network traffic (terminate probing ASAP)

SWINBURNE UNIVERSITY OF TECHNOLOGY

Server Discovery

- All popular FPS games use similar process
  1. Client requests server list from master server
  2. Master server sends server list to client
  3. Client requests information from each server and presents returned information to user
  4. User selects server and then client joins server

Server Discovery cont'd



Empirical Analysis

- Analyse server discovery process for popular FPS game: Wolfenstein Enemy Territory
- Measure game server rank
  - Position of game server (IP, port) in master server list
- Probe master server at different frequencies
  - Every 30 minutes for 20 days (Long trial)
  - Every 60/10 seconds for 4/2 days (Short60/Short10 trials)
- Also measure RTT and number of active players for all active game servers every 6 hours during Long trial

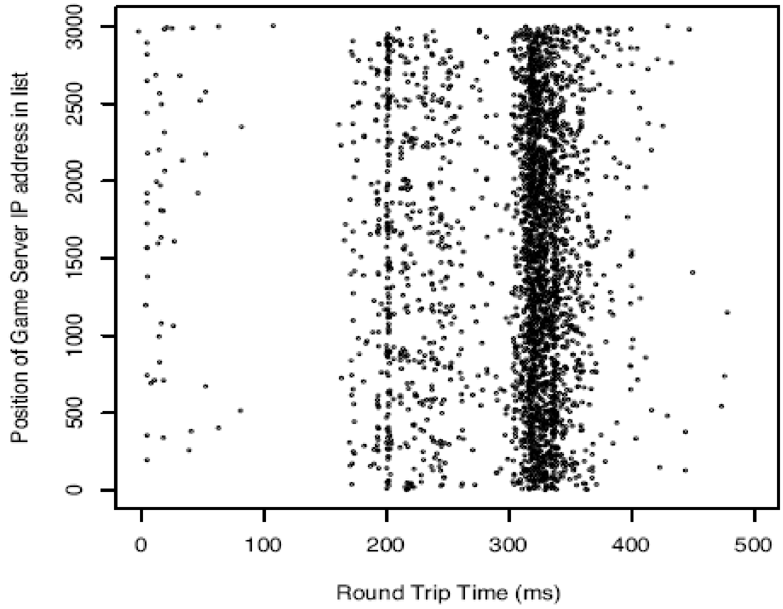
Empirical Analysis Results

- Approx. 3000 game servers at any time
- Approx. 90 servers were inactive at any time
- Reply packets from each active game server is 258-403 bytes long; median and mean both ~300 bytes
- Takes client approx. 1 minute to probe all 3000 servers
- Geographic Distribution (top 10 countries)

Germany (943)	United States (554)	Netherlands (312)	United Kingdom (209)	France (147)
Poland (83)	Finland (60)	Czech Republic (58)	Australia (51)	Japan (50)

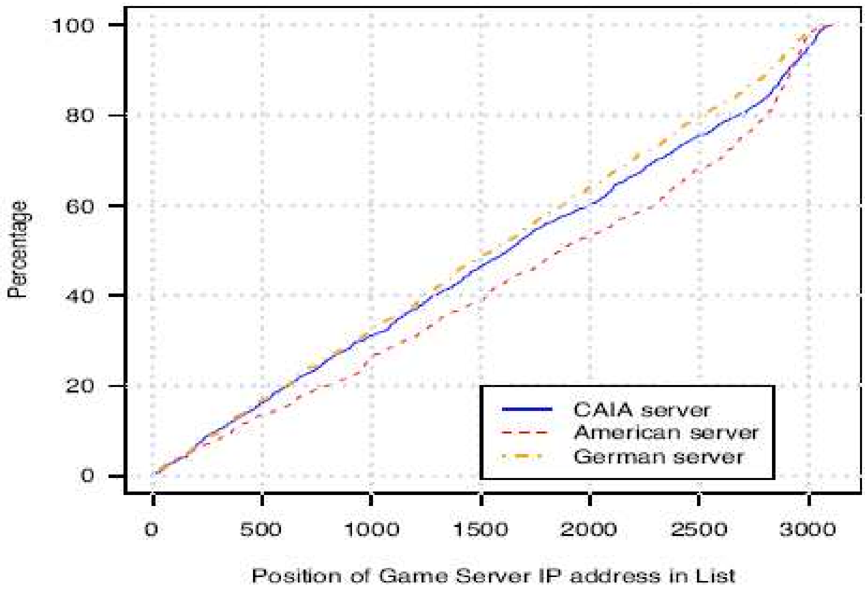
Empirical Analysis Results cont'd

- Game Server Rank versus RTT (Long trial)



Empirical Analysis Results cont'd

- Distribution of game server rank over time (Long trial)



Empirical Analysis Results cont'd

- Short trials revealed that master server cycles game server ranks
  - Newly registered game server starts at top of list (rank 1)
  - Then every time slot game server rank increases modulo size of the server list (meaning server at bottom of list will reappear at top next time slot)
  - Cycle lasts approx. 36 minutes

Proposed Optimisation

- Master servers (or proxies) can optimise server lists **but** more server-side load
- Client-side algorithm that samples master server's list to construct RTT estimates to different countries
  1. Take master server list and groups game servers by country
  2. Select one game server at random from each country
  3. Probe each of these selected game servers in random order
  4. Rank countries by RTT to each of these game servers
  5. Probe all game servers in order of their country's rank (probe servers within one country in random order)
- Local configuration of location or client IP address information is not required (good in presence of NAT)

Proposed Optimisation cont'd

- Median probe response of a game server is 300 bytes long
- Game servers over 180ms away are 'unplayable'
- Client probes 16 game servers in parallel
- Scenario A.1
  - Available game servers are uniformly distributed 20-350ms away from client (access links adding 10-20ms)
  - Game server's rank uniformly distributed
- Scenario A.2
  - Available game servers are uniformly distributed as in A.1
  - Server list ordered by RTT, client only probes servers ≤ 180ms

Proposed Optimisation cont'd

- Scenario A.3
  - Australian client: only 3% of game servers are ≤180ms
  - Server list is ordered and client probes as in A.2
- Scenario B.1
  - Client-side algorithm

Scenario	A.1	A.2	A.3	B.1
Time to probe (16 at a time)	35 sec	9 sec	0.5 sec	1.3
Inbound traffic	878KB	426KB	26KB	32KB

Conclusions

- Collected data from Wolfenstein Enemy Territory master server
- Proposed client-side optimisation of server discovery process ⟶ 878Kb of inbound traffic over 35 seconds could be reduced to 32Kb in <2 seconds
- Reduced probe traffic from clients to game servers (8GB of probe traffic would reduce to just 1.6GB)
- Server discovery is similar for most popular FPS games ⟶ proposed optimisation beneficial to other FPS-style games

Future Work

- Probing single, randomly selected game server to rank each country may be quite misleading
  - Cluster game servers inside each country using indirect indication of topological locality e.g. /8 or /16 IPv4 prefixes
  - Randomly probe one game server from each <country, prefix> group, then rank all game servers according to nominal RTT of their group
- Develop efficient algorithm for master server (or transparent proxy) to optimise servers lists to clients

The End

Questions?