ECE 435 – Network Engineering
Lecture 25

Vince Weaver

https://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

20 April 2023

Announcements

- Final is Thursday May 4th 1:30pm, here
- Will try to go over HW#10 if needed

Final Exam Preview

- Final on Thursday May 4th at 1:30pm, here
- Can have one single-side 8.5x11" piece of paper for notes
- Cumulative, but focusing on things after the first midterm
- Know the 7 OSI layers
- Physical layer: know things like the tradeoffs fiber/copper, satellite, fiber
- Link Layer: Ethernet (why it won over token ring), how collision detection works. Wireless ethernet, how

collision detection works.

- IPv4 – addresses. traceroute output
- IPv6 – addresses, why necessary
- TCP/UDP – why use one over the other, three-way handshake
- Probably no socket programming
- Might show packet dumps, not expect you to memorize all the offsets

Video Game Networking – Types

- Peer-to-peer
- Client / server
  - Easier to prevent cheating
  - Server is always right

Peer-to-Peer issues

- Trust
- Example, battleship
  - Both pick ship locations. Where is that stored?
    Locally, but then when incoming shot comes in, could like about hit or miss
    Instead could both send full map to both clients
    This would make response quicker when fire
    However cheating client would know all positions and could tell the user without other knowing

One solution might be to have a central server that knows positions and only tells client bare minimum
We'll see that has other issues

- Point-to-point hard in face of NAT. Open firewall ports?

Client server

- More trust. Who runs server? Do you trust them?
- Can avoid problem of double NAT traversal as long as server not behind NAT
- More latency

Transport Protocol

- TCP vs UDP
- if use TCP disable nagel's algorithm
- UDP usually lower latency

Protocol

- Serialization – probably don't want XML or JSON
- Text vs Binary
- Packing
- Compression
- Delta compression (Quake3)
- Encryption (prevent cheating)

Latency Issues

- Is latency important?
- Something like online battleship, probably not
- Chess via e-mail? civ4 by e-mail?
- For action game though ms of latency can matter

Latency Mitigation

- Client prediction
  - If you have copy of game world, you can make assumptions
    Walk straight ahead, can move on client w/o waiting for server to respond that it's OK (send update in parallel)
  - Synchronization, response you get from server is from Xms ago so keep that in mind when synchronizing client state with server state

- Interpolation
  - When displaying on client in multiplayer, can have issues where if a player changes suddenly estimated position would be wrong
  - Would seem to teleport suddenly
  - One option is to show your own position real time, but other players a network response time behind
- Lag Compensation
  - What happens when you aim at something, but you're aiming at the past them? And they move?
  - What the server can do is timestamp when you shot

it, and do collision detection based on where they were in the past on your client when you saw them, rather than where they were at the actual time

Video Games – Cheating

- Malicious Packets
- Reading gamestate you shouldn't
- Avoid sending data the user shouldn't know
- Run whole thing in cloud, google stadia

Sample Project Demo