Multiplayer Game Programming

Lecture 4: Routing ,Topology, Wireshark

ITP 484

Today

- Lab 1 Common Issues
- Routing
- Topology
- Wire Shark

Lab 1 Common Issues

- Accessing data in shared_ptr
- make_shared
- Passing By Const Reference
- Returning Member Variables By Reference
- Returning Local Variables By Reference

Public vs Private IP address

- Public
  - Globally Routable
- Private
  - Blocks 192.168.0.0/16, 10.0.0.0/8, 172.16.0.0/12
  - Useful for subnets and internal networks
  - Traffic can leave because the router is connected to other routers that route the Internet, but how does it get back?

Network Address Translation (NAT)

- Rewrites packets as they leave the subnet
  - Changes source IP to the Router's IP so packets can be routed back
  - Changes source Port to a unique port that can be mapped to the host's internal IP address and port
- Rewrites packets as they return
  - Looks up destination port in the mapping table to find original internal IP and Port
- Weird abstraction violation, messing with internal data from two different layers
- Several types of NATs with various restrictions on who can send data back through the NAT to the private network
- Example

NAT

- Advantages
  - Conserves public address space
  - Provides de facto firewall
- Disadvantages
  - Introduces latency
    - Mapping, checksum recalc
  - Hacky: Only works for Transport Layer protocols with ports
  - Provides de facto firewall

Server behind a NAT?

- Manual Port Forwarding
  - User enters data on which ports should forward to internal IP Addresses
- Hole Punching
  - Third party negotiates NAT traversal by fooling the NAT
  - Used by many peer to peer applications that magically work behind NATs (Xbox Live)
  - Example

Dynamic Host Configuration Protocol (DHCP)

- An Application layer protocol that sends a host configuration info, before it has an IP address
- Configuration comes with a lease time
  - To prevent naughty computers from using up all the resources
- Uses UDP
- Uses broadcast to communicate
  - Broadcast address is bitwise OR of IP address and complement of netmask
  - or 255.255.255.255 for current subnet

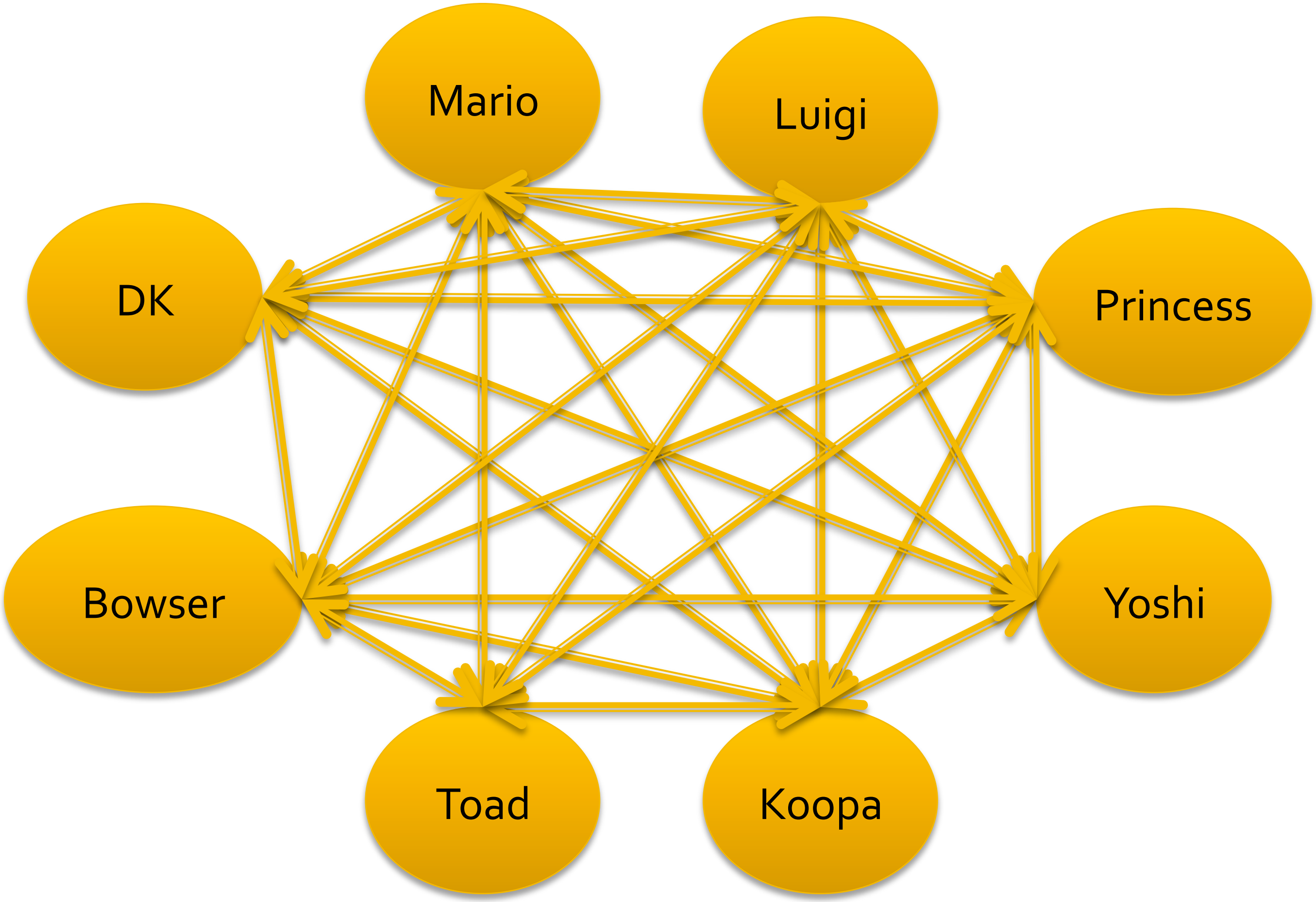
DHCP

- DHCPDISCOVER
  - from 0.0.0.0 port 68 to 255.255.255.255 port 67
- DHCPOFFER
  - From Server IP port 67 to 255.255.255.255 port 68
- DHCPREQUEST
  - Client requests one of multiple offers
- DHCPACK
  - Server acknowledges offer granted, sends final config info (including lease time)

Network Topology

- More than two hosts- how are all your hosts connected?
  - Simple Peer to Peer
  - Peer to Peer with master host
  - Peer to Peer with rendezvous server
  - Client/Server

Peer To Peer



Peer to Peer Internals

- Each peer tracks each other peer
  - IP and address of each other peer
  - One socket per other peer, if TCP
- Each peer's data comes in at different times
  - Need separate threads or non blocking sockets so that there's never a stall waiting for another peer's data
- Each peer runs part of the game, or the whole game

Joining a peer to peer game

- Need to know the IP of every host in the game so that you can send data to all of them

Peer to Peer with Master

- Master can relay all peer IPs and Ports
- Connect to the master peer first and then the master tells the addresses of all other peers
- Protocol change: 3 messages now
  - “Hi! I want to play. Who wants my data?”
  - “Oh hello, here’s everybody who wants your data”
  - Kart and Shell positions (same as before)
  - First two need some kind of reliability

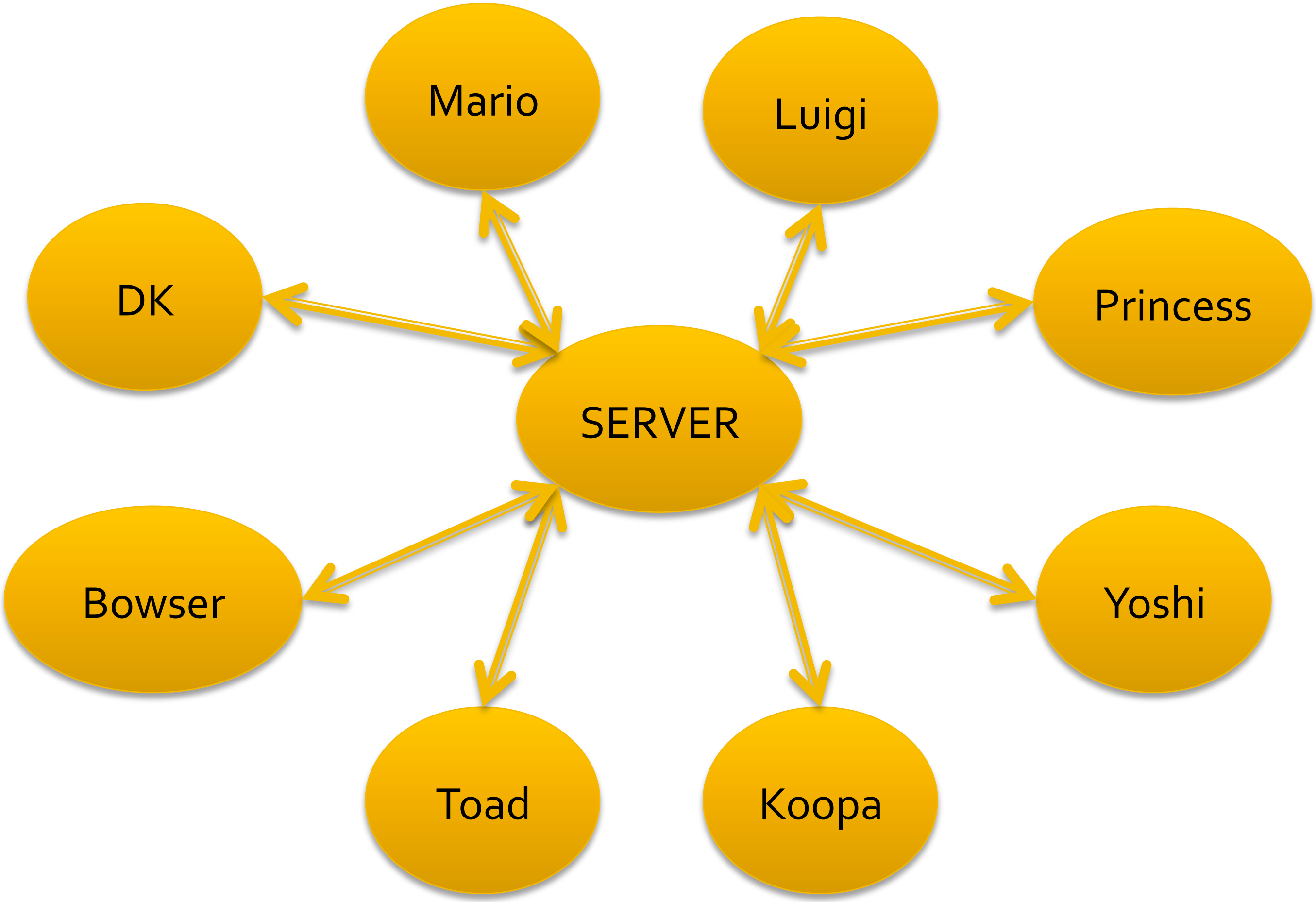
Everybody’s a master

- Everybody knows everybody else’s IP and Port, so anybody could be a master

Peer to Peer with Rendezvous

- Rendezvous server is a third party, agreed upon ahead of time
- Connect to rendezvous server, get IP and port of all players in game
- Allows all peers to connect without first knowing the address of any other peers

Client / Server



Client / Server

- Clients send data to server
- Server sends data to clients
- Server could run part of sim, all of sim, or none of sim

Peer To Peer Distinguishing Features

- Uses n^2 bandwidth
- Symmetric bandwidth requirements
- No designated peer is “hosting the game”
- If any peer drops, rest of peers can keep simulating
- Worst lag between peers is ½ Round Trip Time

Client / Server Distinguishing Features

- Uses 2n bandwidth
- Asymmetric bandwidth requirements- clients use less upstream
- Provides central host for authoritative game logic
- Worst lag a client sees is ½ RTT of other client + ½ RTT of itself + possibly more if server is running at a low framerate
- If the server host drops, then the game is terminated

Host Migration

- How do you handle a server failure
  - Hybrid: Client/Server + each Peer shares data
  - Peers prearrange list of fallback servers
  - Prearrange recovery plan from known data

Wireshark Demo!

- Let’s sniff some packets!