Tech Topic #5

February 8, 2010

Gaming

- Lots of different kinds of games, but the most challenging are the interactive games
  - Typically racing or first person shooter (FPS) games
- The most interactive of the interactive games are
  - Online PC games (e.g., World of Warcraft)
  - Console games (e.g., Xbox, PS3, Wii)
- Most console games have a single player component and an interactive component
  - The interactive component is either a subscription model (Xbox) or free (PS3 and Wii)

Issues in Gaming

- Scale (Bandwidth)
  - The number of players and amount of bandwidth that can be supported
  - Limits are generally driven by minimum available bandwidth
  - With dial-up modems, hard to have any kind of scale
  - Xbox, for example, requires broadband connection
- Delay
  - Opinions on delay vary, but 50ms-100ms seems common
  - Some allowances go as high as 200ms
- Processing
  - Used to be an issue but not really anymore, or rather, is more of an issue for graphics quality but not networking

Scale

- Scale is not so hard to manage
  - Rather, requirements are limited based on strict limits on the number of players
  - Other techniques to reduce bandwidth requirements: reduce complexity, less frequent updates, aggregate updates, limited Areas of Interest, compression, etc.
- Consoles are often limited to 8, 16, or 32 players
  - Much beyond that and it is hard to have rich interactions
- PC games have unlimited players, but only a small number can co-exist in an area
  - For popular areas, replicated servers providing identical areas are used

Delay

- Games are even more sensitive to delay than real-time voice conversations
  - Even with minimal buffering and processing, physical distance propagation times are often too slow

- Consider for example, highly accurate first person shooter games

- Quite a few techniques to avoid dealing with delay
  - Fairly sophisticated efforts to measure delay between players (one of the components of matchmaking)
  - Other, in-game, techniques (e.g., dead reckoning)

Precision and Deadline

- Claypool papers talks about these two ways of describing how important delay is

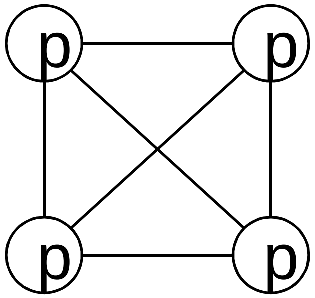
- Precision: accuracy of action
  - Precise v. Imprecise
- Deadline: responsiveness
  - Tight v. Loose

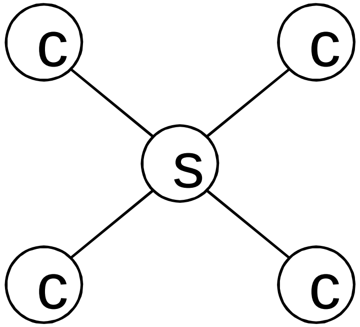
- Game phases…

- But, we want to look at the really challenging parts

Architecture Types

- Two different types of architectures





  - Peer-to-peer
  - Distributed

  - Client/server
  - Centralized

Architecture Types

- Impacts of architecture type

- Where control decisions are made
  - Single point versus distributed fashion
  - Single point of control generally better for security
  - Single point is single point of failure (may be security issue)
  - Single point means server is likely a traffic hotspot
  - Single point may require more capability to handle updates

- Where data has to be sent
  - In single point, players only have to communicate with server
  - In single point, server has to communicate with all players

Client Capabilities

- Distributed architecture
  - All clients have to be able to perform data processing and game status updating
- Centralized architecture
  - A spectrum of possibilities
  - At one end: clients just past actions to server and server responds with updated game state (player location, status, inventory, etc.)
  - At the other end: clients just past game state (all of the processing was done locally)
    - Server still has to verify correctness of actions
- For either architecture, may need to re-sync periodically

Hybrid Architecture

- For some types of data, p2p distribution is used
  - Player-to-player voice communication
  - Non-critical game state
- For other types of data, central server is used
  - Game critical data is sent to the server and processed there (for correctness check)

Dealing with Delay

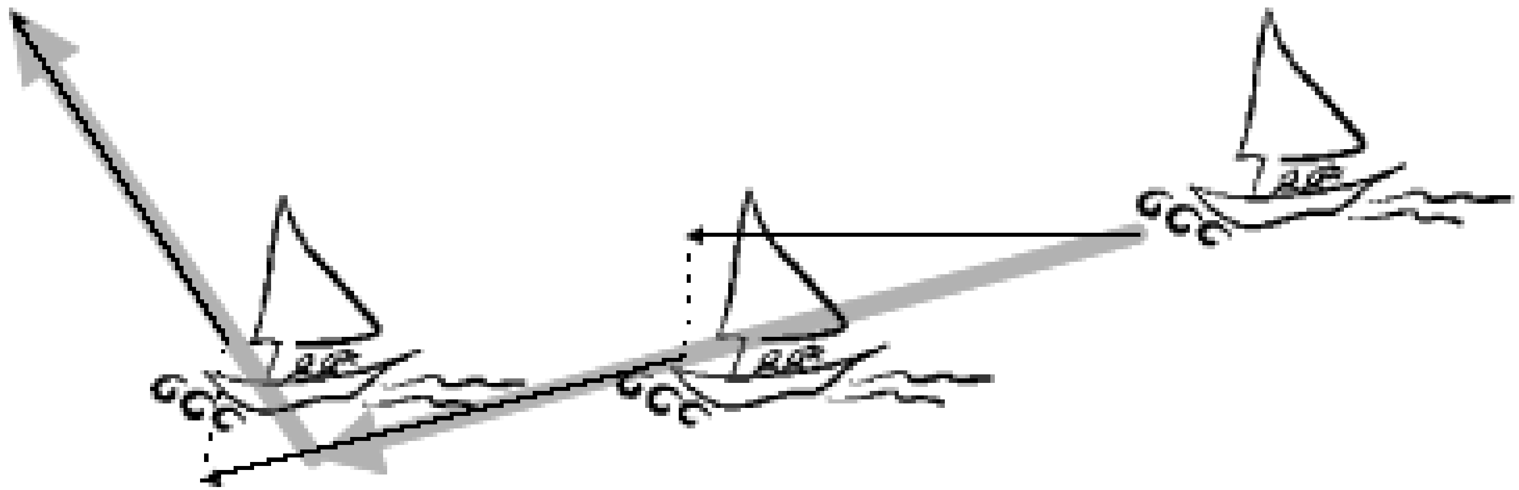
- Given the laws of Physics, how to overcome delay that must necessarily exist?

- Some reduction of the amount of data to send will help with delay, but these days, bandwidth constraints are well understood and can be managed

- Other technique is a combination of local processing and “dead reckoning”

Dead Reckoning

- Allow local player to estimate new position of remote player based on previous position and history
- Tradeoff between: delay and data consistency
  - The longer the time without an update, the more inconsistent the data becomes
  - Use one node (central server?) to deal with inconsistencies



One Possible Solution

- Use a centralized approach
  - Though each of the players has the potential to be the server
- Have some local processing
  - Can receive user inputs and display those on the screen
  - Can also do limited dead reckoning for remote player actions
    - Expecting only modest amounts of delay
- Send game state updates to the server
  - Server responds with whether action was allowed
    - Server runs “virtual world” to compare possibilities versus realities
  - If action is not possible (likely security hack), fix it
  - If conflict occurs (because of dead reckoning), fix it
  - Fixes may result in on-screen “hitch”
- An appearance of minimal delay and fairly good accuracy can be achieved

Server Also Checks for Cheats

- Packet and traffic tampering
  - reflex augmentation
    - proxy replaces human to produce superior results (aiming proxy)
    - e.g., aiming proxy
  - packet interception
    - proxy prevents packets from reaching the cheating player
    - e.g., suppressing the packets containing damage information
  - packet replay
    - the same packet is sent repeatedly (fire command)
      - e.g., the fire command packet
- Information exposure
  - gives information to player he/she is not supposed to have
- Design defects
  - can create loopholes which the cheaters are apt to exploit

Learning About How Games Work

- Traffic traces as a way of reverse-engineering what is happening
  - Typically hard to do because traffic is carefully encrypted

- Can also inject arbitrary delay and observe how game reacts

- And now… more on tracing traffic…