Peer-Server Systems

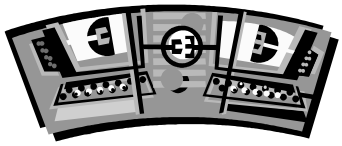
- Peer-to-peer: minimizes latency, consumes bandwidth
- Client-server: effective aggregation and filtering, increases latency
- Hybrid peer-server:
  - over short-haul, high-bandwidth links: peer-to-peer
  - over long-haul, low-bandwidth links: client-server
- Each entity has own multicast group
- Well-connected hosts subscribe directly to a multicast group (peer-to-peer)
- Poorly-connected hosts subscribe to a *forwarding server*
- Forwarding server subscribes to the entities' multicast groups
  - aggregation, filtering

Recapitulation: Resource Management Methods

1. Optimizing the communication protocol
   - packet compression and aggregation
2. Controlling the visibility of data
   - area-of-interest filtering
3. Exploiting perceptual limitations
   - altering visual and temporal perceptions
4. Enhancing the system architecture

§7 Other Issues

- Taxonomy of online cheating
- Analysis of denial-of-service activity
- Synchronized simulation in *Age of Empires*



Network Security

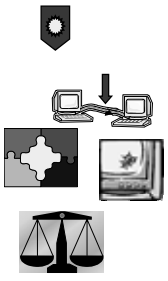
- Military
  - private networks → no problem

- Business, industry, e-commerce,…
  - 'traditional' security problems

- Entertainment industry
  - multiplayer computer games, online games
  - specialized problems

Security and Cheating in Multiplayer Computer Games

- Protect the sensitive information
  - cracking passwords and user accounts
  - pretending to be an administrator
- Provide a fair playing field
  - tampering with the network traffic
  - colluding with other players
- Uphold justice inside the game world
  - abusing beginners
  - ganging up

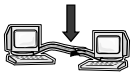


Taxonomy of Online Cheating 1 (4)

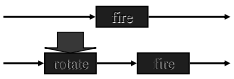
- Cheating by compromising passwords
  - dictionary attacks

- Cheating by social engineering
  - password scammers
- Cheating by denying service from peer players
  - denial-of-service (DoS) attack
  - clog the opponent's network connection

Taxonomy of Online Cheating 2 (4)



- Cheating by tampering with the network traffic
  - reflex augmentation
  - packet interception
  - look-ahead cheating
  - packet replay attack



- Cheating with authoritative clients
  - receivers accept commands blindly
    - requests instead of commands
    - checksums from the game state

Taxonomy of Online Cheating 3 (4)

- Cheating due to illicit information
  - access to replicated, hidden game data
  - compromised software or data

- Cheating related with internal misuse
  - privileges of system administrators
  - logging critical operations into CD-ROMs
- Cheating by exploiting a bug or design flaw
  - repair the observed defects with patches
  - limit the original functionality to avoid the defects
  - good software design in the first place!

Taxonomy of Online Cheating 4 (4)

- Cheating by collusion
  - two or more players play together without informing the other participants
  - one cheater participates as two or more players

- Cheating related to virtual assets
  - demand ⇒ supply ⇒ market ⇒ money flow ⇒ cheating

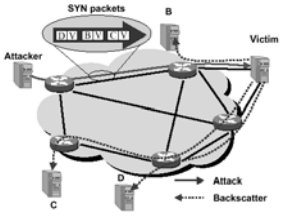
- Cheating by offending other players
  - acting against the 'spirit' of the game
    - players handle the policing themselves → militia
    - systems records misconducts and brands offenders as criminals
    - players decide whether they can offend/be offended

Denial-of-Service (DoS) Attack

- Attack types:
  - logic attack: exploit flaws in the software
  - flooding attack: overwhelm the victim's resources by sending a large number of spurious requests
- Distributed DoS attack: attack simultaneously from multiple (possibly cracked) hosts
- IP spoofing: forge the source address of the outgoing packets
- Consequences:
  - wasted bandwidth, connection blockages
  - computational strain on the hosts

Analysing DoS Activity

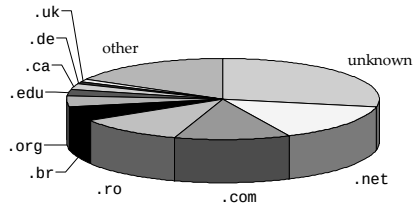
- Backscatter analysis
- Spoofing using random source address
- A host on the Internet receives unsolicited responses
- An attack of *m* packets, monitor *n* addresses
- Expectation of observing an attack: $E(X) = nm/2^{32}$



DoS: Selected Results

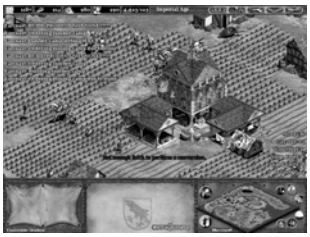
- Three week-long logging periods, February 2001
- >12,000 attacks, >5,000 distinct targets
- Significant number of attacks were directed against
  - home machines
  - users running Internet Relay Chat (IRC)
  - users with names that are sexually suggestive or incorporate themes of drug use
  - users supporting multiplayer games
- In addition to well-known Internet sites, a large range of small and medium sized businesses were targeted

DoS: Most Attacked Top-Level Domains



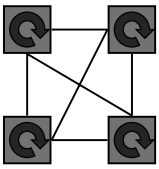
Synchronized Simulation in *Age of Empires*

- *Age of Empires* game series by Ensemble Studios
- Real-time strategy (RTS) game
- Max 8 players, each can have up to 200 moving units
  ⇒ 1600 moving units
  ⇒ large-scale simulation
- Rough breakdown of the processing tasks:
  - 30% graphic rendering
  - 30% AI and path-finding
  - 30% running the simulation and maintenance



Synchronized (or Simultaneous) Simulation

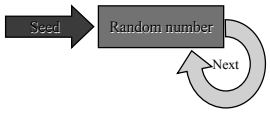
- Large simulation ⇒ a lot of data to be transmitted
- Trade-off: computation vs. communication
  - 'If you have more updating data than you can move on the network, the only real option is to generate the data on each client'
- Run the *exact* same simulation in each client



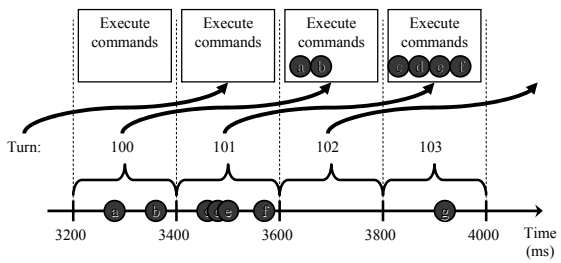
Handling Indeterminism

- 'Indeterministic' events are either
  - predictable (computers) or
  - unpredictable (humans)
- Only the unpredictable events have to be transmitted
  ⇒ communication
  - apply an identical set of commands that were issued at the same time
- The predictable events can be calculated locally on each client
  ⇒ computation
- Pseudo-random numbers are deterministic
- All clients use the same seed for their random number generator
  - disseminate the seed

Pseudo-random number generator

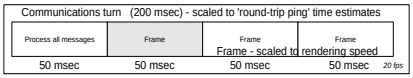


Communication Turns

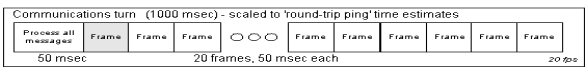


Division of the Communication Turn

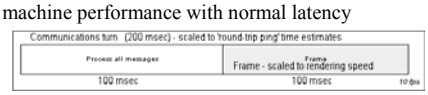
Single communication turn



High Internet latency with normal machine performance



Poor machine performance with normal latency



Features

- Guaranteed delivery using UDP
  - message packet:
    - execution turn
    - sequence number
  - if messages are received out of order, send immediately a resend request
  - if acknowledgement arrives late, resend the message
- Hidden benefits
  - clients are hard to hack
  - any simulation running differently is out-of-sync
- Hidden problems
  - programming is demanding
  - out-of-sync errors
  - checksums for everything
    - 50 Gb message logs

Lessons Learned

- Players can tolerate a high latency as long as it remains constant
  - for an RTS game, even 250–500 ms latencies are still playable
- Jitter (the variance of the latency) is a bigger problem
  - consistent slow response is better than alternating between fast and slow
- Studying player behaviour helps to identify problematic situations
  - hectic situations (like battles) cause spikes in the network traffic
- Measuring the communication system early on helps the development
  - identify bottlenecks and slowdowns
- Educating programmers to work on multiplayer environments

§8 Final Remarks

Outline of the Course (Revisited)

1. Introduction
2. Background
   - history
   - past projects and applications
3. Networking
   - data transfer and protocols
   - communication architectures
4. Managing dynamic shared state
   - consistency-throughput trade-off
   - centralized information repositories
   - frequent state regeneration
   - dead reckoning
5. System design
   - threads
   - polygon culling and level-of-detail
6. Resource management
   - packet compression and aggregation
   - area-of-interest filtering
   - exploiting perceptual limitations
7. Other issues
   - security
   - case examples

Examinations 1 (2)

- examination dates
  1. March 15, 2004
  2. April 5, 2004
  3. May 10, 2004
- check the exact times and places at `http://www.it.utu.fi/opetus/tentit/`
- if you are *not* a student of University of Turku, you must register to receive the credits
  - further instructions are available at `http://www.tucs.fi/Education/Information/regcredits.php`

Examinations 2 (2)

- questions
  - based on the lectures and additional literature (3 articles)
  - four questions à 8 points
  - to pass the examination, at least 16 points (50%) are required
  - questions are in English, but you can answer in English or in Finnish
- remember to enrol in time!