Aspects of Networking in Multiplayer Computer Games

By Jouni Smed, Timo Kaukoranta,
and Harri Hakonen c. 2001

Presented by Cody Olivier

Intro / Motivation

- Virtual reality realm of computing:
  - c1980 Military Simulations
    - Distributed Interactive Simulation (DIS)
  - c1990 Virtual Reality
    - Distributed virtual environments (DVE)
  - c1990 Computer Supported Collaborative Work

Intro / Motivation

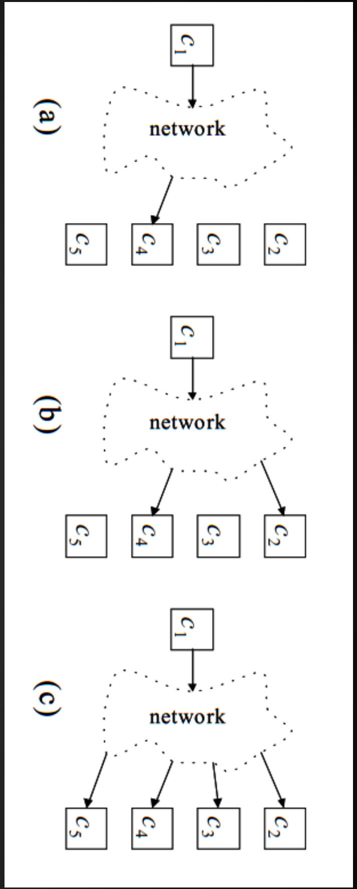
- Share similarities with online games
- Games not seen in scientific literature
- Game companies starting to publishing ideas
- Bridge gap: Layout explicit problems

Topics

- Networking Resources
- Distributed Concepts
- Scalability
- Security and Cheating

Networking Resources

- Bandwidth
  - Amount of data sent per unit time
- Latency
  - Jitter: variance in latency
  - Rule of Thumb: 0.1 - 1 sec
- Computational Power
  - Handling traffic
  - 100,000 objects ⇒ 80 percent CPU
    500Mhz processor

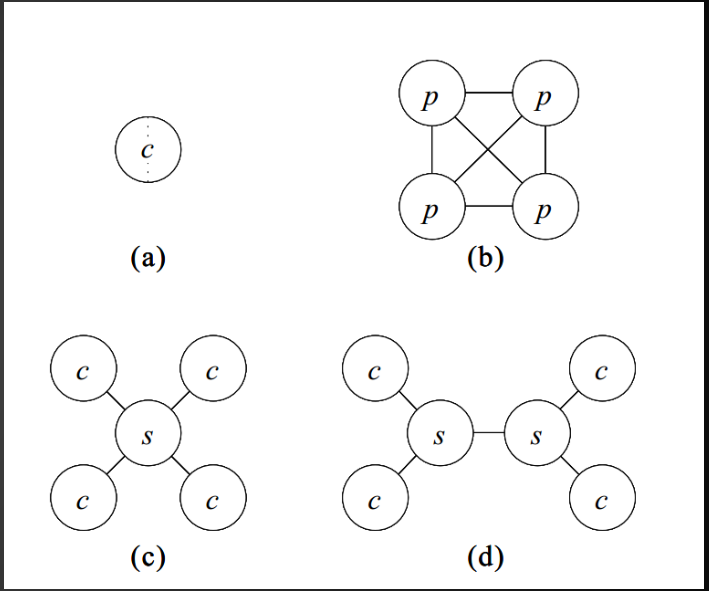


Topics

- ~~Networking Resources~~
- Distributed Concepts
- Scalability
- Security and Cheating

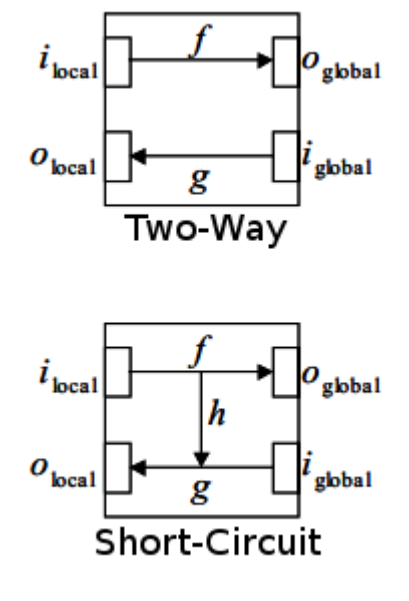
DC - Communication Architectures

- Split Screen
- Peer-to-Peer
  - Great for LAN
- Client / Server
  - Very common today
- Server - Network
  - Upgrade: hierarchy of server-network



DC - Data and Control Architectures

- Data and Control Attributes
  - Consistency: same data everywhere
  - Responsiveness: how responsive are changes
- 2 setups:
  - Two-way relay
    - Commands sent directly to server
  - Short-Circuit
    - Send to server and locally updated



DC - Data and Control Architectures

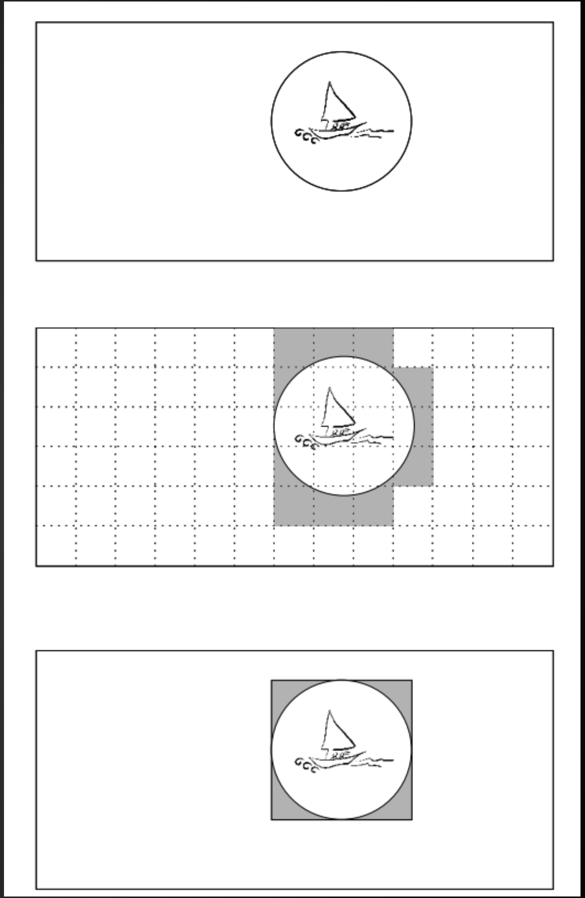
- Architectures:
  - Centralized
    - All data on one server
    - Consistency, two-way relay
  - Distributed
    - Subsets of data on a server
    - Responsiveness, short-circuit relay
  - Replicated
    - Duplicates of data
    - Responsiveness, short-circuit relay

DC - Compensatory Management

- Reducing traffic between nodes
- Message Compression and Aggregation
  - Compress data
  - Aggregating messages
    - Less header info
- Tradeoff - less data sent, more computation

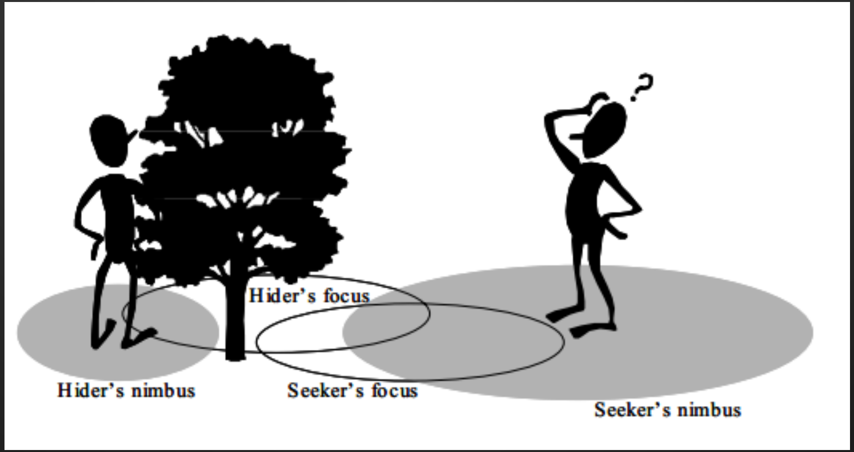
Interest Management

- Nodes express interest in what they want
- Aura or area of interest
    - Only send relevant data to clients
    - 2 entity auros intersect, made aware of each other
    - Symmetric

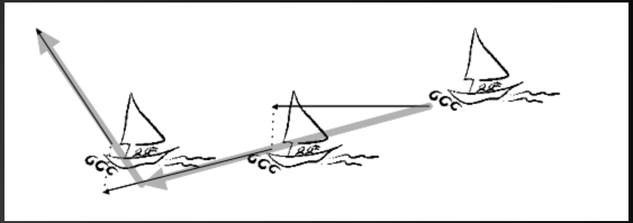


Interest Management

- Finer grade: nimbus and focus
    - When Focus intersects other entity nimbus, update



Dead Reckoning



- Predicting future data
  - Based on navigational techniques
- Position and velocity known, extrapolate future positions
- Increase time between transmissions
- Update when new messages are arrive

Topics

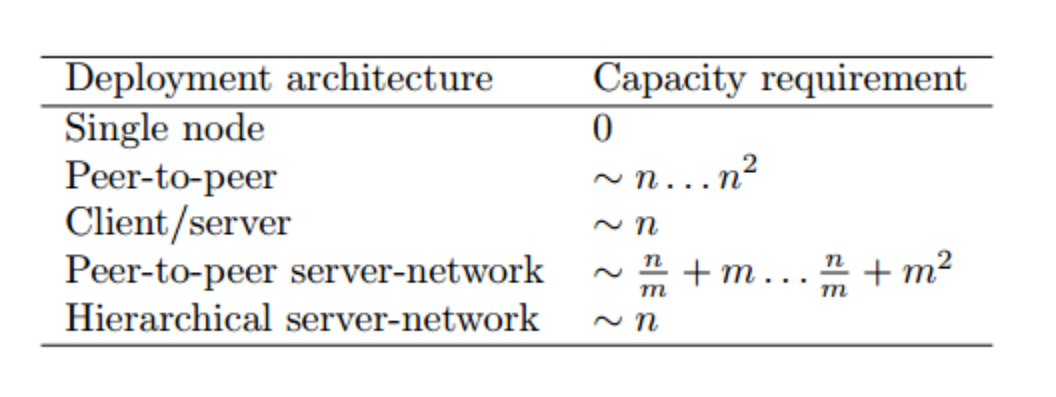
- ~~Networking Resources~~
- ~~Distributed Concepts~~
- Scalability
- Security and Cheating

Scalability - Serial & Parallel Execution

- Speedup based on # nodes and parallel part
- Serializable part immutable
- Extreme 1: no serialize part exists
- Extreme 2: no parallelizable part
- Real-time must be between these
- Client / Server works best

Scalability - Communication Capacity

- Serialized scalability communication limited
- P2P Server-Network: n clients, m servers
- Hierarchical Server-Network: ~n capacity



Deployment architecture	Capacity requirement
Single node	0
Peer-to-peer	$\sim n \ldots n^2$
Client/server	$\sim n$
Peer-to-peer server-network	$\sim \frac{n}{m} + m \ldots \frac{n}{m} + m^2$
Hierarchical server-network	$\sim n$

Ultimate Hierarchical Server-Network

- Sublinear communications
- k-ary tree of hierarchical server-network
- Server sends 1/kth of client data up the tree
  - Aggregation and compression
    - Very unlikely to reduce data to 1/kth
  - Instead, perform interest management
  - Use dead reckoning to limit messages sent

Topics

- ~~Networking Resources~~
- ~~Distributed Concepts~~
- ~~Scalability~~
- Security and Cheating

S&C - Packet and Traffic Tampering

- Issues
  - Reflex Augmentation
  - Packet Interception
  - Replay Attack
- Solutions:
  - Use checksums (MD5) to detect packet tampering
  - Encrypt and add state information (such as random numbers)

S&C - Information Exposure

- Crack clients
    - RTS: see where other players’ units are
    - FPS: wallhacks
- Solution:
    - Data encrypted, memory locations hard to find
    - Make sure commands coming into server are valid
    - In replicable arch, determine what is valid by majority

S&C - Design Defects

- Exploits
- Backend design choices
  - Client trust
  - Solution: Server commands (not checksums)
- Unexpected issues:
  - Latency is high
  - DDOS

Topics

- ~~Networking Resources~~
- ~~Distributed Concepts~~
- ~~Scalability~~
- ~~Security and Cheating~~

Conclusion

- Provide overview of multiplayer game networking topics
- Future work:
  - Test different aspects of released games
  - Look into upcoming encryption for better security
  - Inform game makers of advances in research