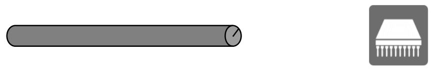
§9 Compensating Resource Limitations

- aspects of compensation
  - information principle equation
  - consistency and responsiveness
  - scalability
- protocol optimization
- dead reckoning
- local perception filters
- synchronized simulation
- area-of-interest filtering

Information-Centric View of Resources



- Bandwidth requirements increase with the number of players
- Each additional player
  - must receive the initial game state and the updates that other users are already receiving
  - introduces new updates to the existing shared state and new interactions with the existing players
  - introduces new shared state
- Additional players require additional processor cycles at the existing player's host
- Each additional player
  - introduces new elements to render
  - increases the amount of caching (new shared state)
  - increases the number of updates to receive and handle

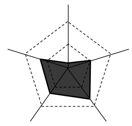
Information Principle

The resource utilization is directly related to the amount of information that must be sent and received by each host and how quickly that information must be delivered by the network.

- The most scalable networked application is the one that does not require networking
- To achieve scalability and performance, the overall resource penalty incurred within a networked application must be reduced

Information Principle Equation

Resources = $M \times H \times B \times T \times P$



M = number of messages transmitted

H = average number of destination hosts for each message

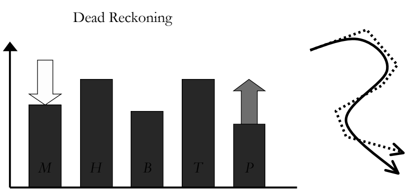
B = average amount of network bandwidth required for a message to each destination

T = timeliness in which the network must deliver packets to each destination

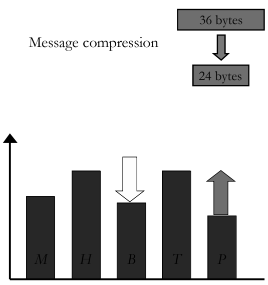
P = number of processor cycles required to receive and process each message

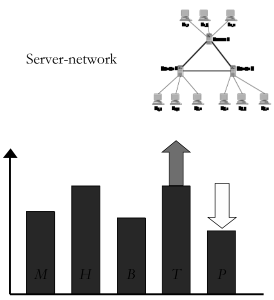
Information Principle Equation as a Tool

- Each reduction ⇒ a compensating increase or a compensating degradation in the quality
- How to modify depends on the application



Information Principle Equation: Examples





Consistency and Responsiveness

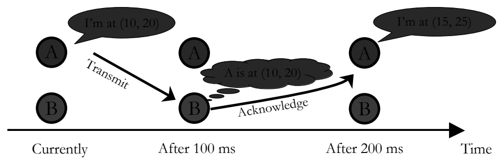
- consistency
  - similarity of the view to the data in the nodes belonging to a network
- responsiveness
  - delay that it takes for an update event to be registered by the nodes
- traditionally, consistency is important
  - distributed databases
- real-time interaction ⇒ responsiveness is important and consistency can be compromised

⇒ the game world can either be
  - a *dynamic world* in which information changes frequently or
  - a *consistent world* in which all nodes maintain identical information

but it cannot be both

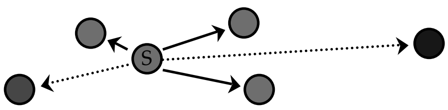
Absolute Consistency

- To guarantee *absolute consistency* among the nodes, the data source must wait until everybody has received the information before it can proceed
  - delay from original message transmission, acknowledgements, possible retransmissions
- The source can generate updates only at a limited rate
- Time for the communication protocol to reliably disseminate the state updates to the remote nodes



High Update Rate

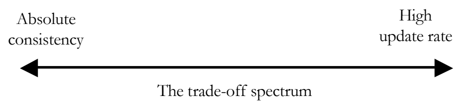
- There is a delay before the state change is received by other nodes
- If the state information is updated often, it might be updated while the previous update messages are still on the way
- Whilst some nodes see new values, others may still see older ones
- Because of the inherent transmission delay, one cannot update the shared state frequently and still ensure that all remote hosts have already received all previous state updates



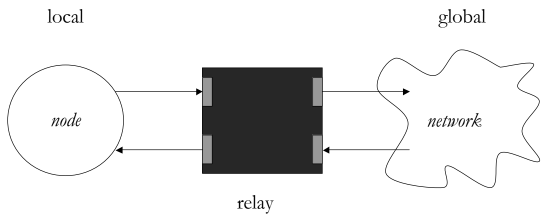
Trade-off Spectrum

- Available network bandwidth must be allocated between
  - messages for updating the state information and
  - messages for maintaining a consistent view of the state information

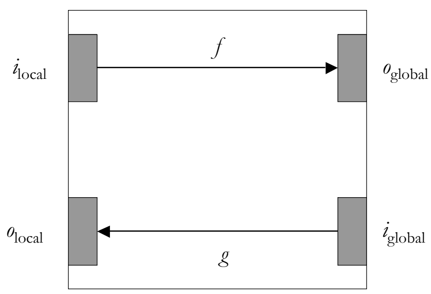
among participants.



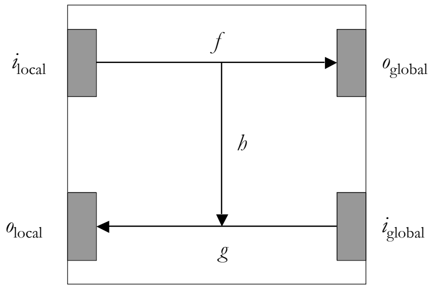
Relay Model



Two-Way Relay

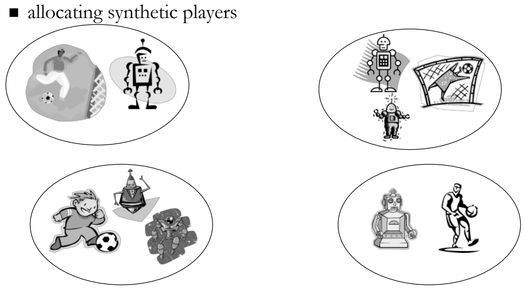


Short-Circuit Relay



Scalability

- ability to adapt resource changes
  - supporting a varying amount of human players
  - allocating synthetic players

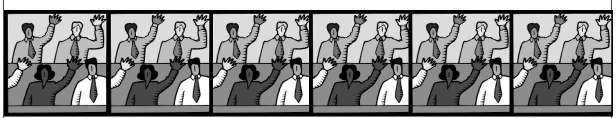


Amdahl's Law

- time required by serially executed parts cannot be reduced by parallel computation
- theoretical speedup:
  $S(n) = T(1) \;/\; T(n) \;\leq\; T(1) \;/\; (T(1) \;/\; n) = n$
- execution time has a serial part T_s and parallel part T_p
  - $T_s + T_p = 1$
  - $\alpha = T_s \;/\; (T_s + T_p)$
- speedup with optimal serialization:
  $S(n) = (T_s + T_p) \;/\; (T_s + T_p/n) \;\leq\; 1/\alpha$
- example: $\alpha = 0.05 \Rightarrow S(n) \leq 20$

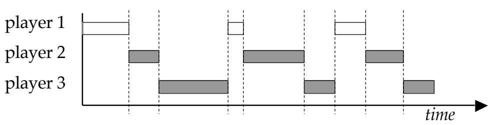
Serial and Parallel Execution

- ideally everything should be calculated in parallel
  - everybody plays their game regardless of others
- if there is communication, there are serially executed parts
  - the players must agree on the sequence of events

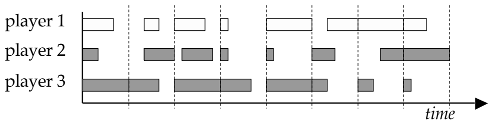


Interaction in a Multiplayer Game

Turn-based game



Real-time game



Communication Capacity: Example

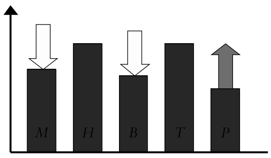
- client-server using unicasting in a 10 Mbps Ethernet using IPv6
- each client sends 5 packets/s containing a 32-bit integer value
  - bits in the message: $d = 752 + 32$
  - update frequency: $f = 5$
  - capacity of the communication channel: $C = 10^7$
  - number of unicast connections: $n = ?$
- $d \cdot f \cdot n \leq C \;\;\Rightarrow\;\; n \leq 2551$

Communication Capacity

Architecture	Capacity requirement
Single node	0
Peer-to-peer	$O(n) \ldots O(n^2)$
Client-server	$O(n)$
Peer-to-peer server-network	$O(n/m + m) \ldots O(n/m + m^2)$
Hierarchical server-network	$O(n)$

§9.2 Protocol Optimization

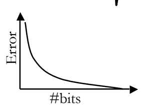
- To transmit data
  - allocate a buffer
  - write data into the buffer
  - transmit a packet containing the buffer contents
- Every network packet incurs a processing penalty
- To improve resource usage, reduce
  - the size of each network packet (message compression)
  - the number of network packets (message aggregation)



Message Compression

Lossless compression
- Change encoding
- No information loss
  - 10.0000001 ⇒ 10.0000001

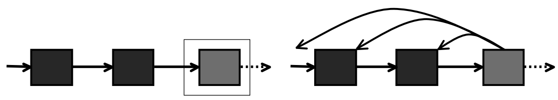
Lossy compression
- Some information may be lost
  - 10.000000001 ⇒ 10



Internal and External Compression

Internal compression
- Manipulates a message based solely on its own content
- No reference to the previous message

External compression
- Manipulates the message data within the context of what has already been transmitted
  - delta information
- Better compression
- Dependency between messages
- Need for reliable transmission

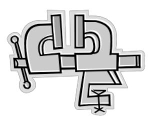


Compression Technique Categories

Compression technique	*Lossless compression*	*Lossy compression*
Internal compression	Encode the message in a more efficient format and eliminate redundancy within the message	Filter irrelevant information or reduce the detail of the transmitted information
External compression	Avoid retransmitting information that is identical to that sent in previous messages	Avoid retransmitting information that is similar to that sent in previous messages

Compression Methods

- Huffman coding
- Arithmetic coding
- Substitutional compression
  - LZ78, LZ77
- Wavelets
- Vector quantization
- Fractal compression



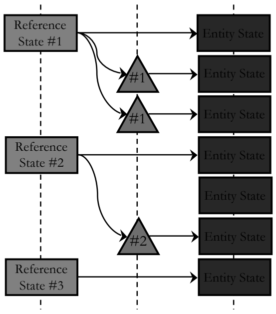
Protocol Independent Compression Algorithm (PICA)

- Lossless, external

Transmit occasionally numbered reference state snapshots

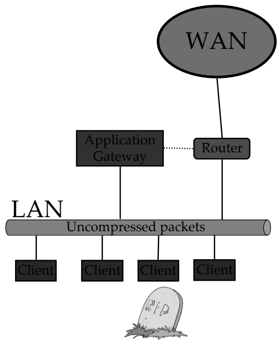
Subsequent update packets
- snapshot number
- delta information

Snapshots reliably
- easy retransmission



Application Gateways

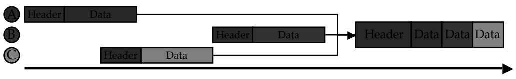
- Compression can be localized to areas of the network having limited bandwidth
- Packet in uncompressed form over the LAN
- Application Gateway (AG) compress them before they enter the WAN
- Quiescent entity service
  - handles dead or inactive entities



Message Aggregation

- Reduce the number of message by merging multiple messages
- Reduces the number of headers
  - UDP/IP: 28 bytes
  - TCP/IP: 40 bytes

Merge all messages of the local entities into a single message
- suits when messages are transmitted at a regular frequency
- does not decrease the quality
- if each entity generates updates independently, the host must wait to get enough messages

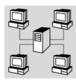


Aggregation Trade-offs and Strategies

- Wait longer
  - better potential bandwidth savings
  - reduces the value of data
- Timeout-based transmission policy
  - collect messages for a fixed timeout period
  - guarantees an upper bound for delay
  - reduction varies depending on the entities
    - no entity updates ⇒ no aggregation but transmission delay
- Quorum-based transmission policy
  - merge messages until there is enough
  - guarantees a particular bandwidth and message rate reduction
  - no limitation on delay
- Timeliness (timeout) vs. bandwidth reduction (quorum)

Merging Timeout- and Quorum-Based Policies

- Wait until enough messages or timeout expired
- After transmission of an aggregated message, reset timeout and message counter
- Adapts to the dynamic entity update rates
  - slow update rate ⇒ timeout bounds the delay
  - rapid update rate ⇒ better aggregation, bandwidth reduction



Aggregation Servers

- In many applications, each host only manages a single entity
- More available updates, larger aggregation messages can be quickly generated
- Large update pool ⇒ projection aggregation
  - a set of entities having a common characteristic
    - location, entity type
- Aggregation server
  - hosts transmit updates to aggregation server(s)
  - server collects updates from multiple hosts
  - server disseminates aggregated update messages
- Distributes the workload across several processors
- Improves fault tolerance and overall performance