Resource Management For Scalability and Performance

448430
Spring 2009
4/27/2009
Kyoung Shin Park
Multimedia Engineering
Dankook University

Large-scale NVE

- Military war gaming is a prevalent example.
- The military envisions support for up to 100,000 simultaneous entities.
- To support the large-scale NVE, NVE designers must use new methods of managing resources such as bandwidth and processor capacity.

Improved Interactive Performance

- Reduced processor load results in quicker processing of user actions at the local machine.
- Excess processor capacity allows for higher-quality graphics generation for the user.
- Reduced network load can lower transmission latencies caused by network congestion and ensure shared state information is exchanged quicker.
- By demanding fewer resources, NVE software can co-exist with other applications on participant's machines.

NVE Designer Goal

- Strive to avoid wasting available resources.
- This chapter provides methods to achieve improved NVE size and performance by reducing bandwidth and processor resources.
  - Scalability
  - Performance

Resource Management

- Explain close relationship between resources and information requirements in NVE.
- 4 broad types of resource management:
  - Communication protocol optimization
    - Packet compression and aggregation
  - Data flow restriction
    - Area-of-interest filtering
  - Leveraging limited user perception
    - Altering visual and temporal perceptions
  - Modifying system architecture

Resource Management

- Network bandwidth increases as the number of new users increases.
- This growth occurs for 3 reasons:
  - Each additional user must receive the initial NVE state as well as updates that other users are receiving.
  - Each additional user introduces new updates to the existing shared state.
  - Each new user introduces an additional shared state to the NVE.

Resource Management

- As more users enter the NVE, additional processor cycles are required at each of the existing users' hosts.
- Each additional user
  - Introduces more elements that the processor must render.
  - Increases the amount of caching (new shared state)
    - Since each user introduces new shard state to the NVE, the processor must cache this additional state, receive updates to this new state, and apply those updates to the cache.
  - Increases the number of updates to receive and handle
    - Because each user introduces additional updates, the processor must be prepared to receive and handle the increased volume of updates and support increased interactions with the local user

Resource Management Summary

- New users increase amount of shared data and level of interaction in the environment.
- More network bandwidth is required to maintain the data and disseminate the interactions.
- As more users enter the NVE, additional processor cycles are required at each of the existing users' hosts.

Networked VE Information Principle

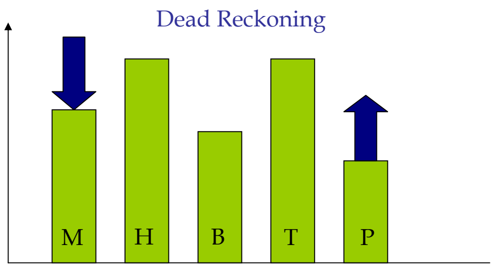
- The resource utilization of a new NVE is directly related to the amount of information that must be sent and received by each host and how quickly that information must be delivered by the network.

Networked VE Information Principle

- The relationship between networking and processing in NVE.
- The key is to improve scalability and performance
- Resources = M * H * B * T * P
  - M = the number of **messages** exchanged
  - H = the average number of destination **hosts** for each message
  - B = the **bandwidth** required for a message
  - T = the **timeliness** in which the packets must be delivered to each destination
  - P = the number of **processor** cycles required to receive and process each message

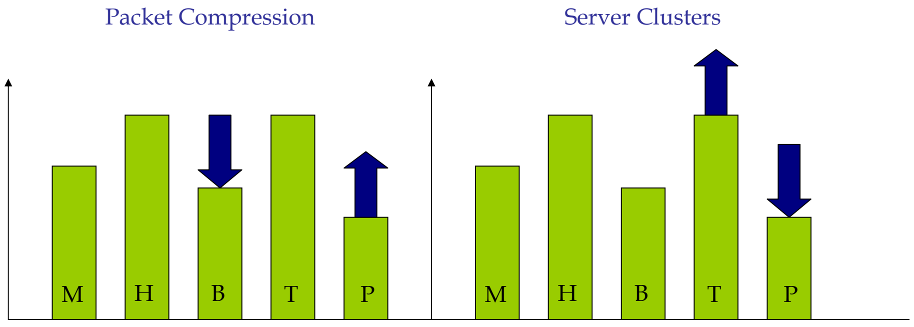
Information Principle Equation as a Tool

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



Information Principle Equation as a Tool

- How to modify depends on the application (See also Table 7-1)

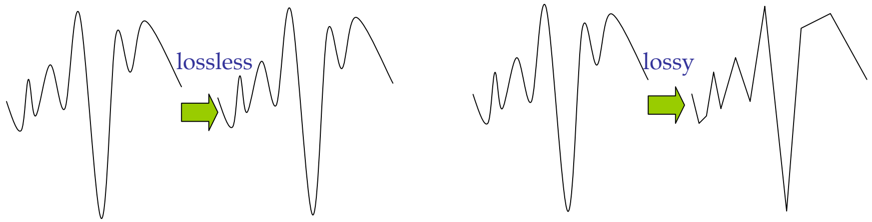


Optimizing Communications Protocol

- Every network packet incurs a processing penalty at the sender and receiver, and every packet must include header information as per TCP/IP and UDP/IP.
- **Network packet optimizations**:
  - **Reduce packet size (packet compression)**
  - **Reduce the number of packets (packet aggregation)**
- How can we reduce M (number of messages) and B (bandwidth per message) while increasing P (processor requirements per packet)?

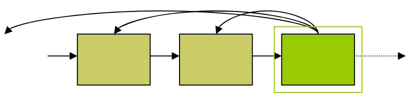
Optimizing Communications Protocol

- **Compression may be either 'lossless' or 'lossy.'**
  - Lossless simply shrinks the format used to transmit the data, but does not affect the information transmitted. e.g. Run-Length Encoding (RLE)
  - Lossy may eliminate some of the information as part of the compression process. e.g. 10.000000001 => 10



Optimizing Communications Protocol

- **Compression can be either 'internal' or 'external.'**
  - Internal compression manipulates a packet based solely on its content, no reference to the previous packets.
  - External manipulates the packet based on what has been transmitted. (delta information)
    - Better compression
    - Dependency between packets
    - Need for reliable transmission



External compression

Optimizing Communications Protocol

- The right choice of compression algorithms tends to depend on the particular NVE application.
- The decision depends on:
  - Frequency of packet updates.
  - Content of packets.
  - Communication architecture and protocols used for packet distribution.
- The choice is derived by analyzing traffic generated earlier in prior executions of the NVE to detect data duplication and other inefficiencies.

Optimizing Communications Protocol

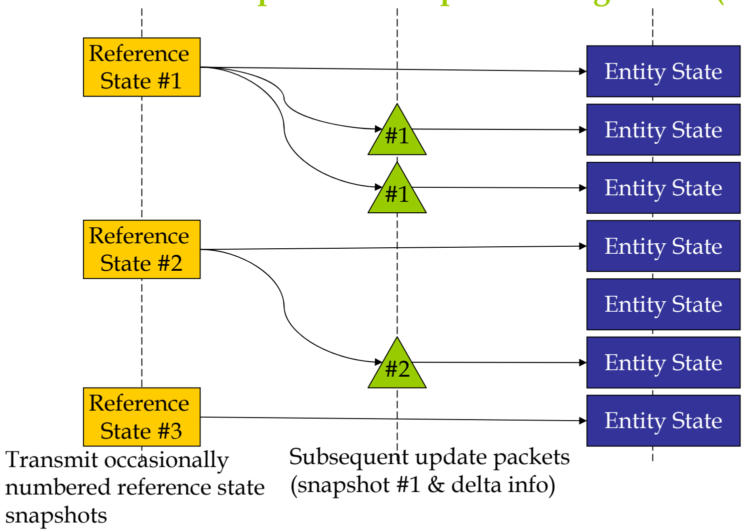
- Network packet compression
  - Protocol independent compression algorithm (PICA)
  - Localized compression using application gateways
- Network packet aggregation
  - Aggregation trade-offs and strategies
  - Aggregation servers

Optimizing Communications Protocol

- Protocol independent compression algorithm (PICA)
  - A lossless, external compression algorithm
  - Used in early versions of the U.S. Military's Strategic Theater of War (STOW) program.
  - Eliminates redundant state information from successive update packets.
  - Can be used with unreliable data transmission protocols.
- PICA Module
  - The compression module is responsible for constructing the update packet by selecting the differences between the current state "snapshot" and the current entity state.
  - When the number of byte differences exceeds a threshold, the PICA compression module transmits a new reference "snapshot (sequence number & entity state)."

Optimizing Communications Protocol

- Protocol independent compression algorithm (PICA)

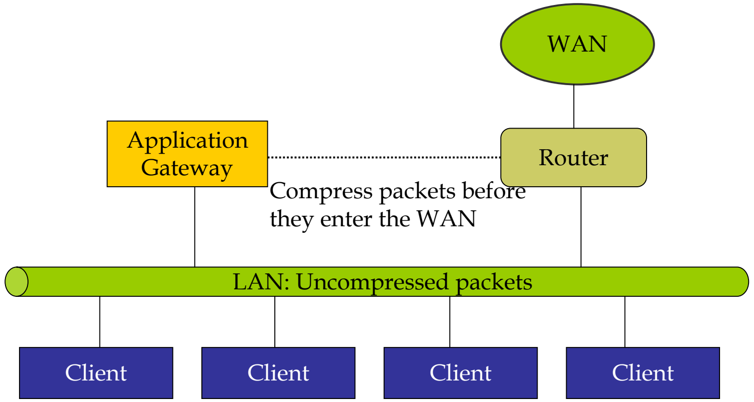


Optimizing Communications Protocol

- Localized compression using application gateways
- Compression and decompression usually occur on the source and destination hosts. However, compression only needs to be localized to areas of the network having limited bandwidth availability.
  - Application gateways (AG) compress data at limited bandwidth areas.
  - Quiescent Entity Service (QES) eliminates repeated update transmissions from low bandwidth WANs.
  - Application gateways (AG) detect and filter inactive entities and generate periodic state updates on behalf of inactive entities located on remote LANs.

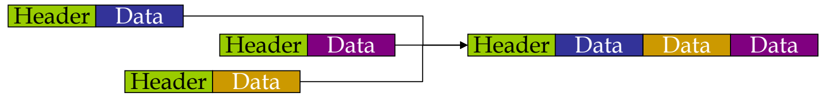
Optimizing Communications Protocol

- Localized compression using application gateways



Optimizing Communications Protocol

- Packet aggregation
- Reducing the number of packets that are actually transmitted by merging information from multiple packets into a single packet.
  - Merging saves network bandwidth by reducing the number of packet headers sent over the network.
  - **Example:**
    - **Each UDP/IP packet includes a 28 byte header**
    - **Each TCP/IP packet includes a 40 byte header**
    - **With the two packets merged, one of the headers is eliminated**.
  - Aggregation can eliminate as much as 50% of bandwidth requirements in a NVE.



Optimizing Communications Protocol

- Aggregation tradeoffs and strategies
  - Aggregation may artificially delay transmission of update packets by waiting until it has enough packets to merge.
  - The receivers must rely on stale information longer than they otherwise would have.
- Tradeoff:
  - By waiting longer to transmit an aggregated packet,
    - it offers greater potential for bandwidth savings (more packets might get merged),
    - but reduces the value of the data (by delaying transmission).

Optimizing Communications Protocol

- Timeout-based transmission policy
- The packet aggregator collects individual packets and transmits them after waiting a fixed period of time.
  - Guarantees an upper bound on the delay introduced on an update packet.
- Drawback:
  - It is possible during the timeout period, only one entity generates an update, so no aggregation will occur even though the transmitted packet still paid the delay penalty.

Optimizing Communications Protocol

- Quorum-based transmission policy
- The packet aggregator merges individual updates until the aggregated packet contains a certain number (quorum) of updates.
  - Guarantees a particular bandwidth and packet rate reduction.
  - Bandwidth reduction is controlled by the number of updates that are merged.
- Drawbacks to the quorum:
  - No limitation on how long the entity update is delayed.
  - Could be delayed indefinitely while waiting for more updates to merge.

Optimizing Communications Protocol

- Hybrid transmission approach (Merging timeout- and quorum-based policies)
- The quorum-based approach can transmit fewer packets than timeout-based transmission, but its delay characteristics are less predictable.
- Wait until enough packets or timeout expired
- After transmission of an aggregated packet, reset timeout and packet counter
- Adapts to the dynamic entity update rates
  - At slow update rates, the timeout prevails preventing indefinite waiting.
  - At fast update rates, the quorum prevails and achieves the target level of update aggregation.

Optimizing Communications Protocol

- Aggregation servers
- Servers collect updates from multiple source hosts and disseminate aggregated update packets using one of the strategies.
- Good for managing inactive entity updates.
- Quiescent Entity Service (QES)
- Generate aggregated updates on behalf of entities that are dead or just inactive.

Optimizing Communications Protocol

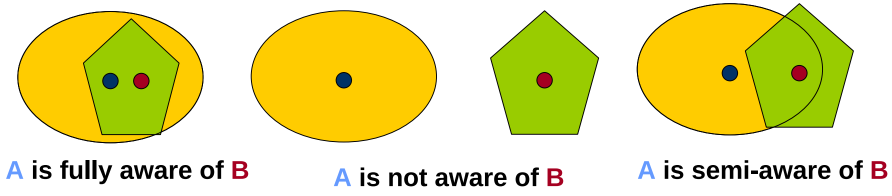
- Advantages of aggregation servers
  - Reduces the need for a single high-power machine by distributing the workload.
  - Improved fault tolerance characteristics of the NVE should a server fail.
  - Use of multiple servers can improve the overall performance of the aggregation process.

Controlling the Visibility of Data

- *The goal of the NVE designer should be to send information to those hosts who really need to receive it.*
- **Area-of-interest filters**
  - Each host provides explicit data filters
  - Filters define the interest in data
- **Multicasting**
  - Use subscription-based routing protocols to restrict the flow of data
  - Divide the entities or the region into multicast groups
- **Subscription-based aggregation (hybrid)**
  - Group available data into fine-grained 'channels' to which destination hosts may subscribe
  - Hosts subscribe the appropriate channels

Controlling the Visibility of Data

- **Aura-Nimbus Information Model**
  - Aura - entity data should only be available to those entities that are capable of perceiving that information.
  - Nimbus – entity data should only be available to those who are interested in that information.
  - Aura-Nimbus has the disadvantage in that it does not scale to large numbers of entities.
  - Each packet has a custom set of destination entities – hard to utilize multicasting



Controlling the Visibility of Data

- **Area-of-interest (AOI) filtering subscriptions**
  - Hosts transmit information to a set of subscription managers (or area-of-interest managers, filtering servers).
  - Managers receive subscription descriptions from the participating hosts.
  - For each piece of data, the managers determine which of the subscription requests are satisfied and disseminate the information to the corresponding subscribing hosts.
  - AOI filtering is restricted form of the pure aura-nimbus model.
    - Subscription descriptions specify the entity's aura
    - Reduces the processing requirements of the pure aura-nimbus model

Controlling the Visibility of Data

- **Subscription interest language**
  - Allows the hosts to express formally their interests in the NVE.
  - Subscription description can be arbitrarily complex.
    - A sequence of filters or assertions
    - Based on the values of packet fields
    - Boolean operators
    - Programmable functions

```
(OR
    (EQ TYPE "Tank")
    (AND
        (EQ TYPE "Truck")
        (GT LOCATION-X 50)
        (LTE LOCATION-X 75)
        (GT LOCATION-Y 83)
        (LTE LOCATION-Y 94)
        (EQ PACKET-CLASS INFRARED)))
```

Controlling the Visibility of Data

- **Joint Precision Strike Demonstration (JPSD)**
  - **Filtering subscription-based system example**
  - Military NVE built to train army tactical commanders.
  - Most of the entities are artificially constructed with no human controller.
  - Demonstrated with up to 6000 participating entities with 80 hosts.
- Subscription management at each source host
- Each host manages subscriptions for all local entities
- The host sends packets directly to the interested clients using peer-to peer unicast.
- Interest subscriptions
  - Logical predicates, operators (equality, 'within range')
  - External function modules in a library

Controlling the Visibility of Data

- When to use customized AOI filtering data flows?
  - Hosts cannot afford the cost of receiving and processing unnecessary packets
  - Hosts are connected over an extremely low-bandwidth network
  - Multicast or broadcast protocols are not available (i.e., only available on point-to-point connection)
  - Client subscription patterns change rapidly
  - No a priori categorizations of data
- AOI filtering has a problem when a large number of hosts are interested in the same piece of information
  - Customized data streams => unicast => the same data travels multiple times over the same network

Controlling the Visibility of Data

- Intrinsic and extrinsic filtering
  - Intrinsic filtering
    - The filter must inspect the application content of each packet to determine if it should be delivered to a particular destination host. (Area of interest filtering)
    - Can dynamically partition data based on fine-grained entity interests
  - Extrinsic filtering
    - Filters packets based on external properties such as the network address to which the packet was transmitted. (Multicasting)
    - Implementation efficient
    - Filtering cannot be as sophisticated

Controlling the Visibility of Data

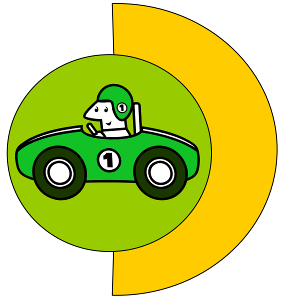
- **Multicasting**
  - A network protocol technique whereby the application sends each packet to a "multicast group" by supplying a special "multicast address" as the destination for the UDP/IP packet.
  - The packets are delivered exclusively to those hosts who have subscribed to the multicast group.
  - A host must explicitly subscribe (Join) to receive and unsubscribe (Leave) to stop receiving.
  - A host can subscribe to multiple groups simultaneously.
  - Transmission to a group does not require subscription.
  - Relatively efficient compared to broadcast protocols.
  - **Challenge: How to partition the available data among a set of multicast groups?**
  - **Worst case: Each host is interested in a small subset of information from every group => must subscribe to every multicast address => broadcasting**

Controlling the Visibility of Data

- **Multicasting approaches:**
  - Group per entity approach
    - Allows each host to receive information about all hosts that lie within its nimbus. Each host executes its subscription filter locally, based on entities that exist in the NVE.
  - Group per region approach
    - Partitions the NVE world into regions and assigns each region to one or more multicast groups. Each entity transmits its data to groups corresponding to regions that cover its current location.
  - Hybrid multicast aggregation approach
    - Strikes a balance between the two approaches. The hybrid scheme aims to partition the NVE data into fine-grained multicast groups as well as ensuring that the data is not so fine grained that the approach does not degenerate into simple unicast. The data must be broad enough to include multiple source entities.

Controlling the Visibility of Data

- **Group-per-entity approach**
  - A difference multicast address to each entity
  - Each host receives information about all entities within its nimbus.
  - Subscription filter is executed locally
  - Subscribe to the groups which have interesting entities
  - Entities cannot specify their nimbus; no control over which hosts receive the information
  - Example: PARADISE
    - Each entity subscribes to nearby entities
    - Control directional information interests
      - Nearby entities that are behind
      - Nearby and distant entities that are in front

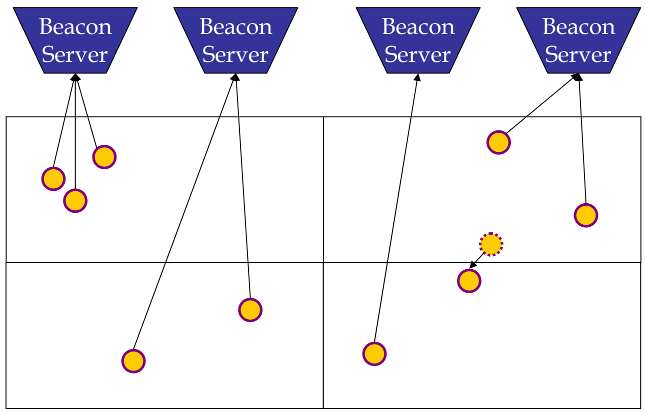


Controlling the Visibility of Data

- **Group-per-entity approach**
  - Multiple multicast group addresses to each entity
    - Position updates
    - Infrared data
  - Information at a finer granularity
  - More accurate nimbus by group subscriptions
  - Host need a way to learn about nearby entities
  - Entity directory service tracks the current state of the entities
    - Entity transmits periodically state information
    - Directory servers collect the information and provide it to the entities when requested

Controlling the Visibility of Data

- Beacon servers

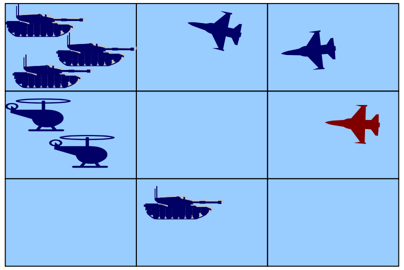


Controlling the Visibility of Data

- Group-per-entity approach drawback
  - Consumes a large number of multicast addresses
  - Address collisions become quite probable
  - Network routers have to process the corresponding large number of join and leave requests
  - Group search induces network traffic
  - Network cards can only support a limited number of simultaneous subscriptions
    - Too many subscriptions => promiscuous mode

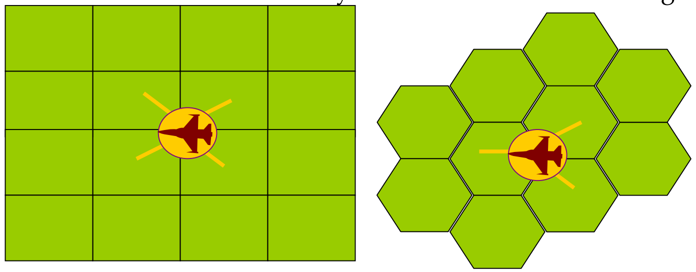
Controlling the Visibility of Data

- Group-per-region approach
  - Partitions the NVE world into regions and assigns each region to one or more multicast groups.
  - An entity transmits to groups corresponding to the regions that cover its location.
  - The entity subscribes to groups corresponding to interesting regions.
  - Entities have limited control over their aura but less control over their nimbus.



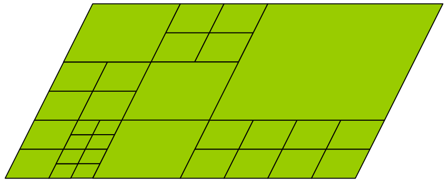
Controlling the Visibility of Data

- Regions bounds
  - An entity has to change its target groups throughout its lifetime.
    - Track the bounds of the current regions
    - Learn the multicast address of a new region
    - Boundaries and addresses assigned to the regions are often static
  - In grid-based region assignment there are many points at which multiple grids meet.
  - Near these corners an entity has to subscribe to several groups



Controlling the Visibility of Data

- Hybrid multicast aggregation
  - Balance between fine-grained data partitioning and multicast grouping.
  - Three-tiered interest management system
    - Group-per-region scheme segments data based on location
    - Group-per-entity scheme allows receiver to select individual entities
    - Area-of-interest filter subscriptions



Controlling the Visibility of Data

- **Projection-based multicast aggregation**
  - A system whereby information is partitioned based on multiple parameter values. These groups are referred to as a "projection."
    - Each projection is associated with its own multicast group address.
  - Projection example:
    - Suppose each entity in a NVE has a type and location. Each projection is specific to entity types and locations. One projection might include tanks located between (10,25) and (35,40) while another projection includes cars located between (85,70) and (110,85) in the NVE.
  - Instead of sending data directly to the projection's multicast group, source hosts can send data to a projection aggregation server.
    - Each server is responsible for managing one or more projections, collecting data and sending aggregated packets that contain multiple updates.
  - Advantage: Less network bandwidth consumed.

Taking Advantage of Perceptual Limitations

- Humans have inherent perceptual limitations
  - Delay in human reaction to events that occur
  - Humans cannot discern intricate details about appearance or location if the object is distant.
  - Humans are relatively forgiving about small inaccuracies in a rendered scene.
- Two approaches to exploit
  - Information can provided at multiple levels of detail (LOD) and at different update rates
  - Mask the timeless characteristics of information

Taking Advantage of Perceptual Limitations

- **Exploiting LOD perception**
- Only the users who are located near the entity in the VE need to receive the high-detail information.
  - Accurate structure, position, orientation
  - Update rate in a local frame rate
- Distant viewers can tolerate less detail and less information about current structure, position, and orientation.
  - Transmitting high-resolution to distant viewers imposes unnecessary bandwidth burdens on the network and processing burdens on the receiving hosts.
- Many inaccuracies cannot even be detected on a fine-resolution display.

Taking Advantage of Perceptual Limitations

- **Multiple channel architecture**
- Each entity transmits multiple independent data channels each with a different LOD and frequency.
- Example:
  - The low resolution channel might provide updates once every 20 seconds and contain only the entity's position.
  - The high resolution channel might provide updates every 3 seconds and include the entity's position, orientation, and dynamic structure.
  - Viewers subscribe to the channel that provides their required LOD.
- Though more packets are transmitted, bandwidth requirements are reduced across the network. (Distant viewers shift to low bandwidth)

Taking Advantage of Perceptual Limitations

- System reliability
- Channels that generate low-frequency updates, a reliable multicast scheme may be used to prevent receivers from operating with stale data.
- Channels that generate high-frequency updates, will quickly replace lost updates.

Taking Advantage of Perceptual Limitations

- Channel choices tradeoff
- How many channels should a source host provide?
  - The more choices, the greater the chance of matching rendering accuracy and computational requirements.
  - Too many choices imposes high cost in terms of computation at the source host and in terms of bandwidth for the host's local network links.

Taking Advantage of Perceptual Limitations

- How many channels to provide for an entity?
  - More channels, better service for subscribers
  - Each channel imposes a cost (bandwidth and computation)
- To satisfy the trade-off, the source provides 3 channels for each entity.
  - Rigid-body channel (for far-range viewers)
  - Approximate-body channel (for mid-range viewers)
  - Full-body channel (for near-range viewers)
  - Each channel provides an order-of-magnitude difference in structural and positional accuracy, and an order-of-magnitude difference in packet rate.

Taking Advantage of Perceptual Limitations

- Rigid-body channel
- Source host transmits enough information to allow remote hosts to represent the entity as a rigid body.
  - Used when the entity is distant from the local viewer.
- Demands the least network bandwidth and processor computation.
  - Updates position, orientation, and basic structure only.
  - Ignores changes in the entity's structure.

Taking Advantage of Perceptual Limitations

- Approximate-body channel
- Enables remote hosts to render a rough approximation of the entity's dynamic structure (i.e, appendages or articulated parts)
  - Remote hosts subscribe for non-rigid bodies that are close enough to notice structural differences yet far enough to tolerate inaccuracies in structural representation.
  - Consumes more bandwidth and computational resources
- Provided information is entity-specific
  - Corresponds to the dominant changes of the structure
- Common approximations
  - Radial length
  - Articulation vector
  - Local coordinate system points

Taking Advantage of Perceptual Limitations

- Full-body channel
- Provides the highest level of detail about the entity's dynamic position, orientation, and structure.
  - A viewer is restricted to subscribe to a limited number of channels at any one time.
- Required for nearby entity's or when the viewer needs to interact with the entity.
  - Imposes the highest bandwidth and computational requirements.

Taking Advantage of Perceptual Limitations

- Summary of multiple channels
  - The source host must simultaneously support the low, medium and high fidelity requirements of its entity viewers.
  - Multiple channels improves scalability by allowing each viewer to independently determine its rendering accuracy requirements.
  - Multiple channels reduce the aggregate traffic and computation throughout the NVE by shifting most subscriptions toward the lower frequency, lower data volume updates.

Taking Advantage of Perceptual Limitations

- Exploiting temporal perception
  - Since multiple-fidelity channels that can provide largely inaccurate representation due to dead reckoning schemes based on stale or incomplete information an approach that eliminates this problem is required.
  - Exploiting temporal perceptions alleviates this problem by rendering the entity in an accurate, but slightly out of date, location, though it was there at some time in the past.
  - Used when the local user is not interacting with the rendered entity and the temporal inaccuracies can be hidden inside the rendered NVE.
  - A formalization of the frequent state regeneration techniques.

Taking Advantage of Perceptual Limitations

- Advantages of exploiting temporal perceptions
  - Because packet recipients can explicitly hide the effects of network latency, the NVE can be safely deployed over the WAN having greater latency.
  - Network bandwidth requirements can be reduced by enhanced packet aggregation techniques that artificially delay transmission of data.
  - These techniques can enhance the use of dead reckoning and other prediction techniques by reducing the required prediction time interval and limiting the potential prediction error. When temporal techniques are used, prediction is used to hide the effects of network latency rather than provide an accurate model of the entity's position.

Taking Advantage of Perceptual Limitations

- Active and passive entities
  - Active entity
    - Takes actions on its own and generates updates
    - Includes human participants as well as computer controlled entities
    - Cannot be predicted typically
    - Rendered using state updates adjusted for the latency
  - Passive entity
    - Only reacts to events from its environment and does not generate it own actions.
    - Includes inanimate objects (e.g., rocks, books, etc).
  - Because active entities interact with passive entities, the NVE must render each passive entity according to the network latency of its nearest active entity, so users perceive that the passive entity reacts instantaneously to the actions of the active entity.

Enhancing the System Architecture

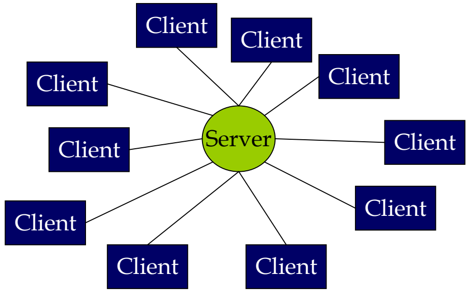
- *Optimizing by changing the logical structure of the NVE system to enable more efficient information dissemination.*
- NVE's system architectures are divided into:
  - Basic structure
    - Client-server
    - Peer-to-peer
  - Hybrid to support greater scalability and performance
    - Server clusters
      - Partition clients across multiple servers
      - Partition the NVE across multiple servers
    - Server hierarchies
    - Peer-server systems

Enhancing the System Architecture

- Two ways to improve efficiency in large scale virtual environments:
  - Client-server architecture can be expanded to include a cluster of servers to communicate in a peer-to-peer manner.
  - Client-server and peer-to-peer structures are merged to create the peer-server architecture.

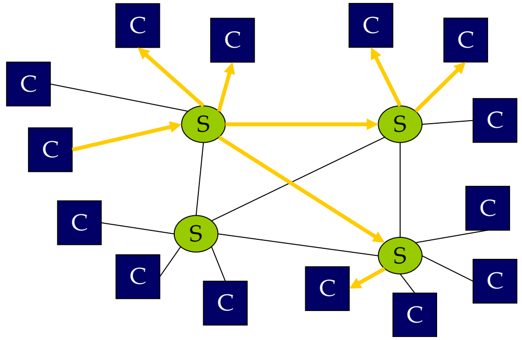
Enhancing the System Architecture

- Traditional client-server
  - Server may act as
    - Broadcast reflector
    - Filtering reflector
    - Packet aggregation server
  - Scalability problems
    - All traffic goes through the server (bottleneck).



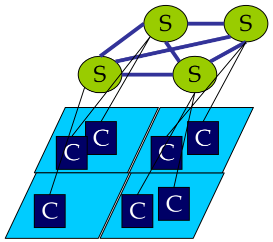
Enhancing the System Architecture

- Partitioning clients across multiple servers
  - The server exchange 'control messages' among themselves to inform the interest of their clients
  - Reduces the workload on each server
  - Incurs a greater latency
  - The total processing and bandwidth requirements are greater



Enhancing the System Architecture

- Partitioning the NVE across multiple servers
  - Each server manages clients located within a particular region
  - Each client communicates with different servers as it moves
  - Partitioning the NVE can eliminate almost 95% of the information exchange among servers
  - However this requires advanced configuration work to enable information exchanges among the servers, especially when one region can see into another.



Enhancing the System Architecture

- Server hierarchies
  - Server themselves act as clients in a client-server relationship with higher-level servers.
  - The higher level servers communicate information on behalf on NVE regions represented by 'client' servers.
  - The higher level servers send updates to its client servers who forward this information to its interested entities.
  - When a client server forwards updates up to the higher level server, the higher level server determines which areas require the information and forward the update to other high-level servers who then send the information to its interested client servers.

Enhancing the System Architecture

- **Peer-server systems**
  - The hybrid peer-server technique merges the best characteristics of the traditional peer-to-peer and client-server systems characteristics.
  - **'Forwarding server'** subscribes to multicast groups for entities of interest to the destination host, performs aggregation and filtering functions, and forwards updates to the destination hosts.
  - **'Monitoring directory server'** collects information about the environment and dynamically determines which hosts should receive transmissions from each entity in the environment.
  - **'Reachability testing'** determines whether a source host can communicate with a destination host.
  - The peer-server technique represents an example of an adaptive NVE system architecture that dynamically adjusts to account for network topology, host location, and network dynamics.

Conclusions

	Impact on Information Principle Equation	Description
Packet Aggregation	Reduce *M*, *B* Increase *P*, *T*	Merge multiple updates into a single packet using timeout- or quorum-based techniques
Multicasting	Reduce *H* Increase *M*	Transmit updates to a subset of participating entities employing a group for each source entity or for each net-VE region
Area-of-Interest Filtering	Reduce *H* Increase *M*, *P*, *T*	Define explicit information filters and process at a subscription manager
Projection Aggregations	Reduce *H*, *B* Increase *M*, *P*, *T*	Pre-define fine-grain multicast groups for particular subscription filters
Reduced Level-of-Detail	Reduce *H*, *B* Increase *M*, *P*	Provide low-fidelity data channels for distant viewers
Temporal Contour	Reduce *T* Increase *P*	Render remote entities using delays corresponding to perceived network latency
Server Clusters	Reduce *P* (per host) Increase *T*	Employ servers that communicate peer-to-peer, create server hierarchies
Peer-Server Communication	Reduce *T*, *B*, *M* Increase *T*, *B*, *M*	Select between peer-to-peer and client-server communication based on network characteristics

Recapitulation

- Optimizing the communication protocol
  - Packet compression and aggregation
- Controlling the visibility of data
  - Area-of-interest filtering
- Exploiting perceptual limitations
  - Altering visual and temporal perceptions
- Enhancing the system architecture