Local Perception Filter
With Time Sync
Without Time Sync
Maintaining tightly synchronized states
States can go out of date. A player sees a state that happened $t$ seconds ago.
Hybrid Model: 

**Render** objects within real-time interaction range in real time, other objects in delayed time.
A happening now

B happened time $t$ ago
Question:
What if a player A throws a ball at player B?
happening now

A

happened time t ago

B
Question:
What if a player B throws a ball at player A?
happening now

happened time t ago
Two Kinds of Entities

Active:
players (unpredictable)

Passive:
ball, bullet (predictable)
Question:
What if a player A throws a ball at player B?
happened $t$ ago
happened $t - 1$ ago
happened $t - 2$ ago
Question:
What if a player B throws a ball at player A?
happened $t$ ago
happened $t - 1$ ago
happened $t - 2$ ago
B throws ball at A. A sees that the ball’s speed increases as it gets nearer to A.
A throws ball at B. A sees that the ball’s speed decreases as it gets away from A.
Server not shown in this example.
From perspective of a player A, the other player is surrounded by a “temporal distortion field” defined by the communication delay between that player and A.
1D temporal distortion field from A’s perspective.
1D temporal distortion field from B’s perspective.
Extension to multi players

temporal distortion

rendered position of ball

A C B
Limitations

Delay jitter leads to fluctuating field.

Can’t interact directly with other players.
Local Perception Filter
Bullet Time
Slow down time to allow more reaction time.

E.g. “Max Payne”
Difficult in multiplayer game -- naive implementation slows every player down.
Ideally, players who invoke “bullet time” slow down their game, but others play on.
“bullet time” -- bullet slows down as it comes near the player; increases speed as it moves away.
bullet time distortion field from B’s perspective.

temporal distortion

rendered position of ball
overall distortion field

temporal distortion

rendered position of ball

A

B
overall distortion field from B’s perspective.

temporal distortion

rendered position of ball

A  B
bullet time distortion field from A's perspective.
You Are Here

- CS4344
  - Client/Server Architecture
  - Synchronization Protocols
You Are Here

- CS4344
  - Client/Server Architecture
    - Synchronization Protocols
  - Interest Management
Bandwidth Requirement
A Measurement Study of Shen Zhou Online, an MMORPG.

7 kbps

Average bandwidth per client
2.5 : 1

Peak-to-Mean Ratio for Bandwidth
370,000

Simultaneous Number Of Players
(Ragnarok Online, December 2004)
6.5 Gbps

Peak Server Bandwidth
860 Terabyte

Amount of data transferred per month
Need to reduce bandwidth overhead
Dead Reckoning
Interest Management
Relevance Filtering
**Idea:** only need to update another player $p$ if the update matters to $p$. 
Aura / Area-of-Interest
Update of $p$ matters to $q$ if the auras of $p$ and $q$ intersect.
Foci

(what a player can see)
Nimbi
(where a player can be seen)
Update of $p$ matters to $q$ if the foci of $p$ intersects nimbi of $q$. 
Calculating aura/foci/nimbi can be costly.
Idea: approximate use bounding boxes
or approximate using cells
Large cell: Redundant messages.
Small cell: Large management overhead.
The white player will receive many messages he/she is not interested in.
Idea: we can dynamically partition the cells into smaller ones as needed.
Generalization: an entity may specify any other events/entity it is interested in.
Communication Abstraction
**Multicast**: send a message to a set of subscribers
Group: a channel to publish messages
A client can subscribe to/join a group to start receiving messages from that group.
A client can unsubscribe from/leave a group to stop receiving messages from that group.
Anyone can send a message to a group (need not be a subscriber).
Each cell is a group. A subscriber can subscribe to multiple cells. A group can have multiple publishers.
Implementation:

IP Multicast
Multicast groups are identified using class D IP addresses

(224.0.0.0 to 239.255.255.255)
Any message sent to a multicast address will be sent to all its subscriber.
Anyone listening to a multicast address will receive messages sent to it.
Problems with IP Multicast
Not reliable
join/leave takes time
not widely deployed
need states at the router
and many others..