

Spatial Publish Subscribe

IEEE Virtual Reality workshop MMVE'09

Shun-Yun Hu

National Central University, Taiwan

2009/03/15



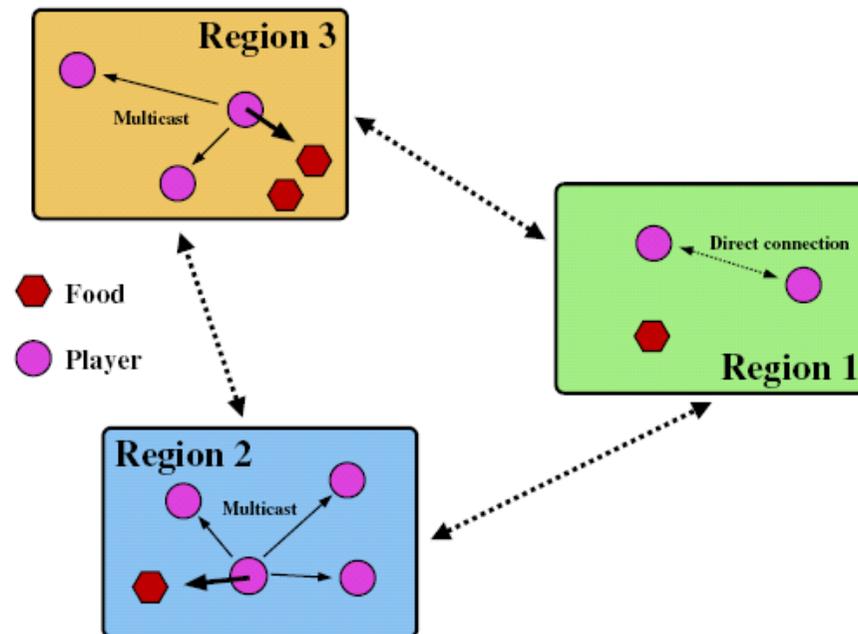
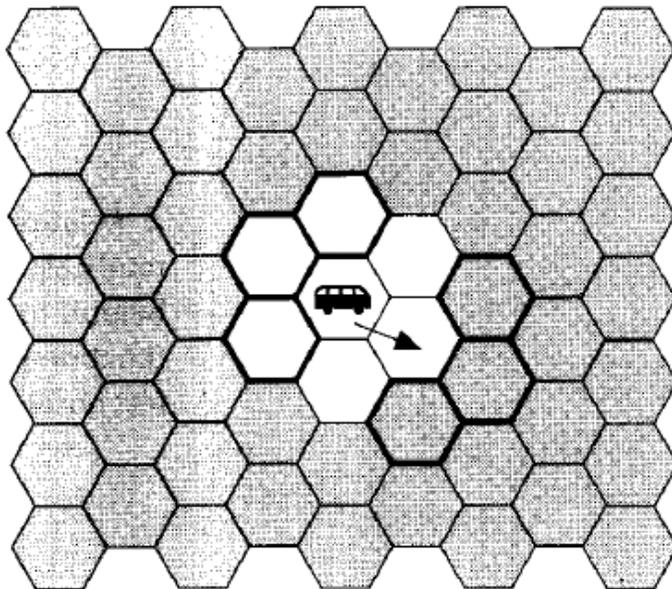
A basic primitive for virtual worlds?

- Many existing designs for building scalable virtual environments (VEs)
- Any common requirements / features?
- A common interface / API?



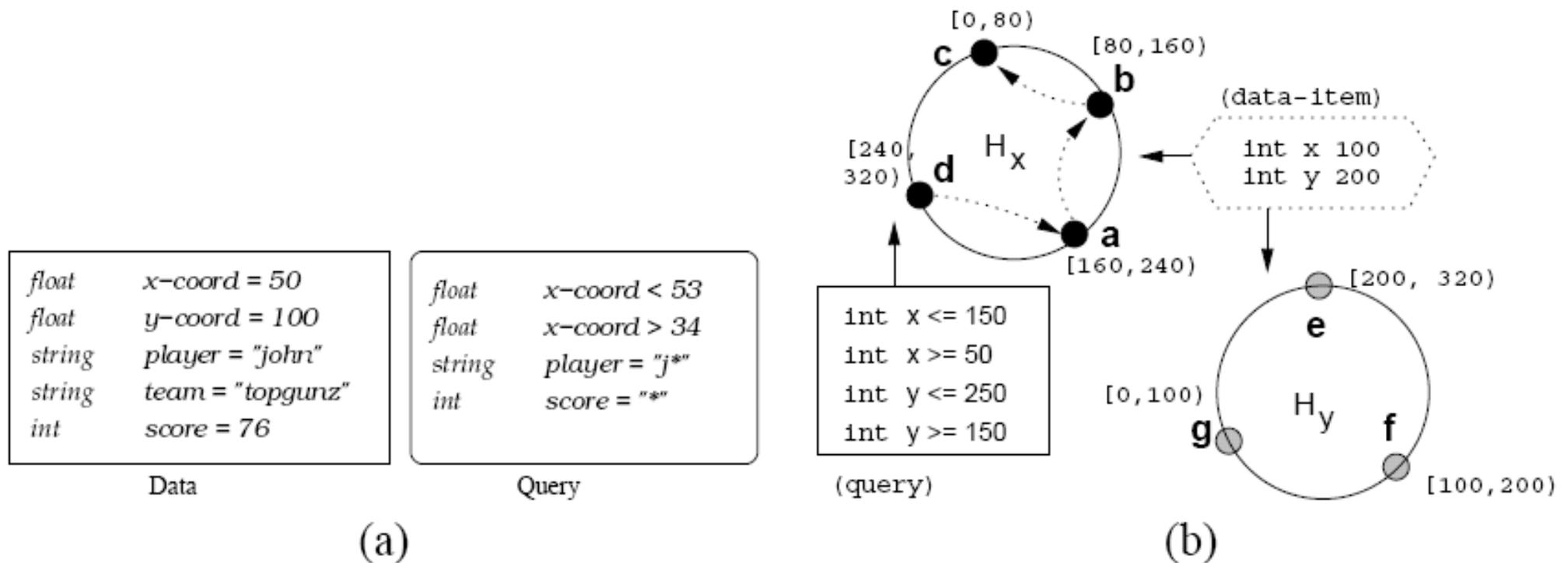
Spatial multicast (know thy neighbors)

- NPSNet (Macedonia et al. '95)
- SimMud (Knutsson et al. '04), Solipsis, VON, COVER,...
- Simple to understand, easy to implement



Spatial query (know thy neighbors' goods)

- OpEN (Tanin et al., '04)
- Colyseus (Bharambe et al, '06), GP3, ...
- Persistent states that allow multiple queries





Spatial Publish / Subscribe (SPS)

- Why not support both?

- Spatial multicast → publications
- Spatial query → subscriptions

- Flexibility

- publication / subscription of any shape / size

- Advantages

- No continuous query ← spatial query
- Fine-grained filtering ← spatial multicast



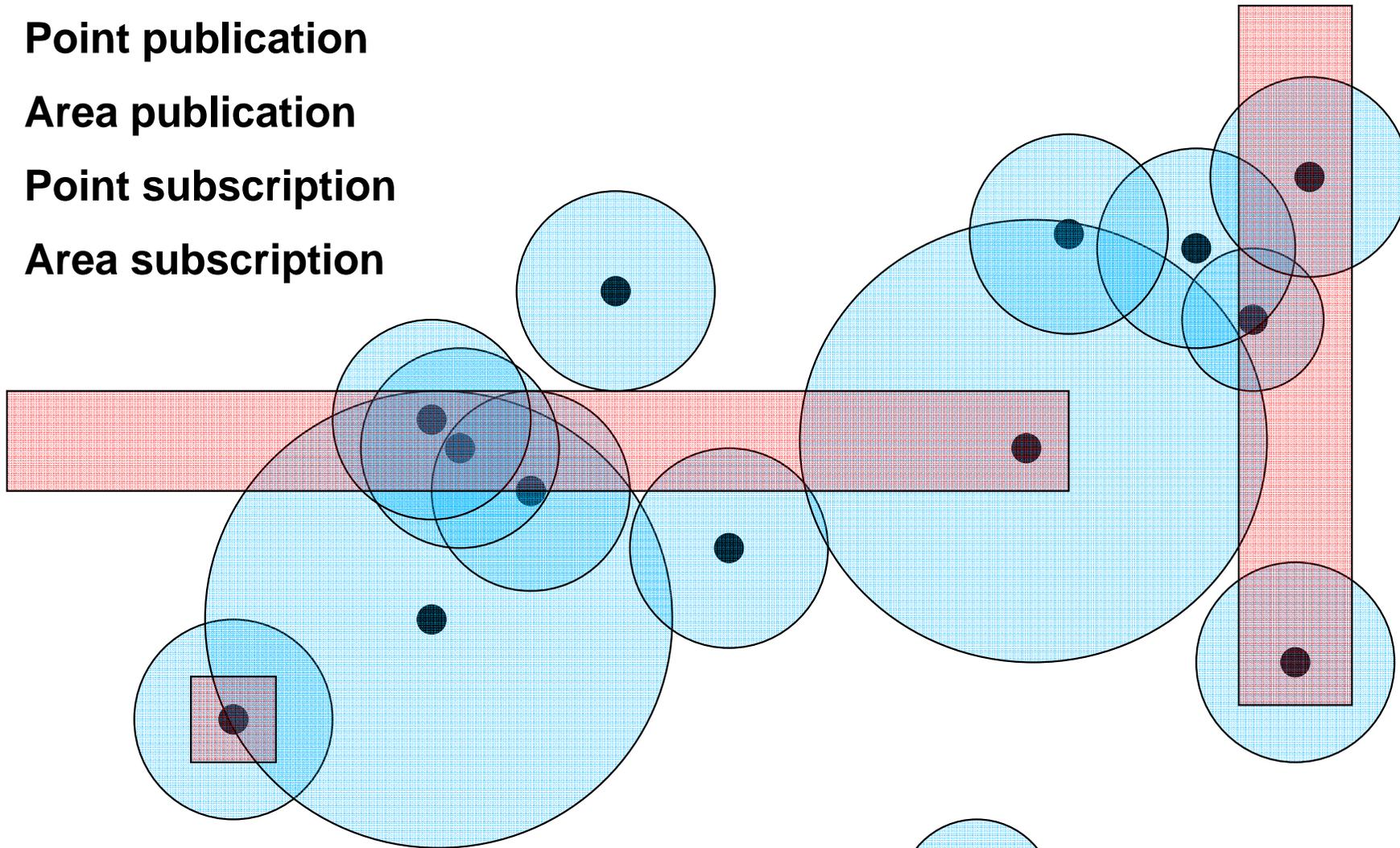
SPS: basic operations

Point publication

Area publication

Point subscription

Area subscription



pub

sub



Usage scenario

- Existing research (in the P2P space)
 - Overlay management
 - How do peers connect?
 - Solipsis, VON, N-tree, COVER, OPeN, APOLO, VoroCast
 - State management
 - How to manage objects?
 - SimMud, HYMS, Colyseus, VSM, Hydra
 - Content management
 - How to deliver content?
 - Voice (QuadCast, PartyPeer), 3D (FLoD, LoDDT, HyperVerse)



Overlay management

- Goal: provide a list of AOI neighbors
- SPS approach
 - Area subscriptions + point publications
 - Point subscriptions + area publications
 - State managers for late joiners



State management

- Goal: arbitrators to update & distribute states
- SPS approach:
 - Arbitrators (i.e., servers):
 - Area subscription in *event layer*
 - Point publications in *update layer*
 - Actors (i.e., clients):
 - Point publications in *event layer*
 - Area subscriptions in *update layer*
 - Arbitrators to update late joiners



Content management

- Goal: content delivery for same-view users
- SPS approach
 - Discovery: subscription of AOI neighbors
 - Exchange: peer-exchange with AOI neighbors



Conclusion

- SPS may be a flexible primitive for VEs
- Implementation challenges
 - Scalable (low message overhead)
 - Responsive (publications → subscribers)
 - Topology-aware (considers physical proximity)
- Important research topic for VE community



Q & A

Thank you!



What's missing?

■ More generic functionalities

□ Flexibility

- Any publication / subscription areas (a 30 km gun!)
- Any mix of direct / forward connections

□ Practicality

- Network environment (topology / NAT-aware)
- Client environment (capacity-matching, superpeers)



Related work

- Communication architecture (Fiedler et al. '02)
 - Grid partitioning + channel-based pub/sub
- Mercury (Bharambe et al, '06)
 - Spatial query to support pub/sub
 - Too flexible (any object field can be range-queried)
- DiGAS (Bonotti et al. '07)
 - Flood all publications to all brokers
- HLA's Data Distribution Management (DDM) ('97)
 - Too flexible (similar to Mercury)

