
CSCI 8220

Parallel & Distributed Simulation

PDES: Distributed Virtual Environments
Dynamic Data Distribution



Outline

- **Introductory Comments**
- **Dynamic (Value based) Data Distribution: HLA Data Distribution Management**
 - » Routing space
 - » Publication Region
 - » Subscription Region
- **DDM Implementation**
 - » Cell-Based
 - » Region-Based
 - » Combining Cells and Regions

Review: Background

- **Basic question:** When a simulator generates information (e.g., state updates) that may be of interest to other simulators, who should receive the message?
- **Example:** Moving vehicles in a virtual environment
 - » Moving vehicle sends “update” messages indicating new position
 - » Each vehicle that can “see” the moving vehicle should receive a message
 - » **How does the sender/RTI know which other federates should receive the message?**
 - Data distribution is essentially a message routing problem

Review: Communication Primitives

- **Unicast**

- » One sender, message received by one destination

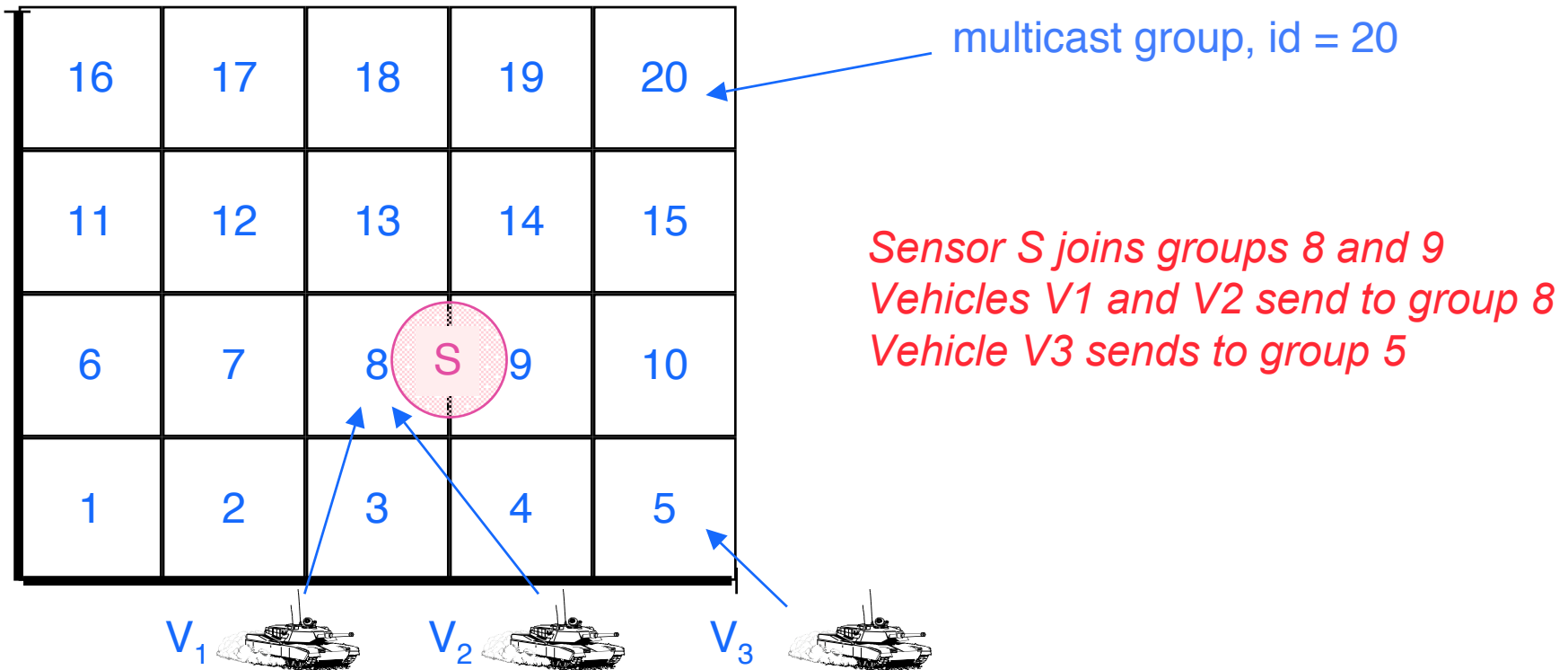
- **Broadcast**

- » One sender, message received by all destinations

- **Multicast**

- » One sender, message received by multiple (but not necessarily all) destinations
- » Operations (analogous to newsgroups)
 - Join group
 - Leave group
 - Send message to group
- » Can be implemented by unicast, or network multicast
- » Best effort vs. reliable multicast

Using a Grid to Capture Locality



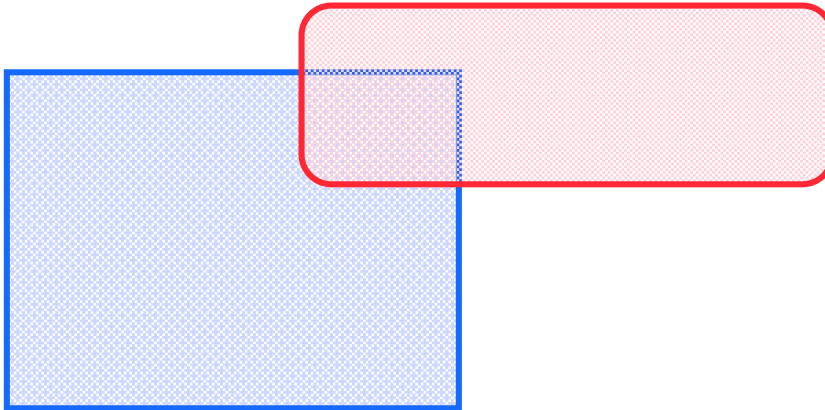
- Divide play-box into non-overlapping (rectangular, hexagonal) grid cells
- Create one multicast group for each cell
- Subscribe to cell(s) you can “see”
- Send message to cell where the vehicle resides
- Requires additional filtering at the receiver

HLA Data Distribution Management (DDM)

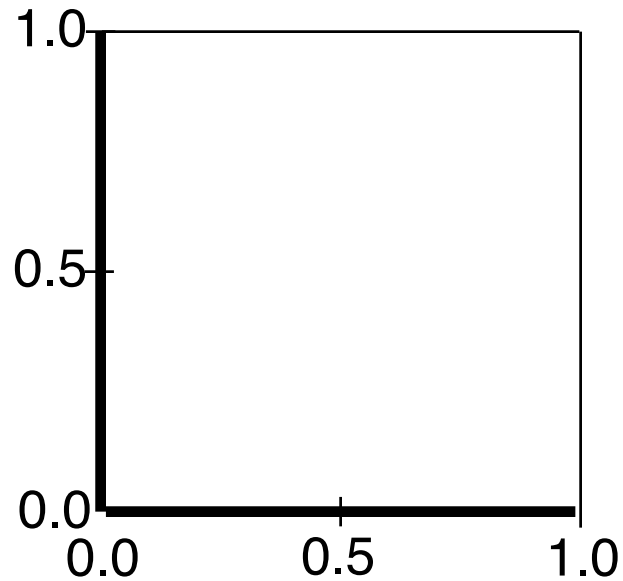
- HLA DDM provides a general mechanism
- **Name Space**
 - » **Routing space**: N-dimensional coordinate system
 - » **Separate from simulation state, used solely for routing**
- **Interest expressions**
 - » **Subscription region**: N-dimension rectangular in routing space
 - » **Associate region with subscription requests**
- **Description expressions**
 - » **Update region**: N-dimensional rectangle in routing space
 - » **associated with each object instance**

HLA Data Distribution Management (DDM)

- A message updating an attribute of an object instance is routed to a federate if:
 - » The federate is subscribed to the object's class and attribute, **and**
 - » The update region associated with the updated attribute overlaps with the federate's subscription region for that class/attribute



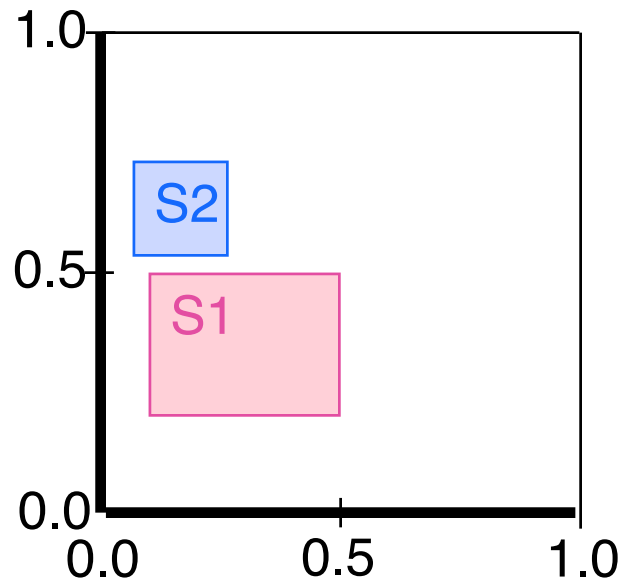
HLA Data Distribution Management



Name Space

- **N dimensional routing space**
- **Playbox in virtual environment**
- **Radio channels for wireless communication**

HLA Data Distribution Management

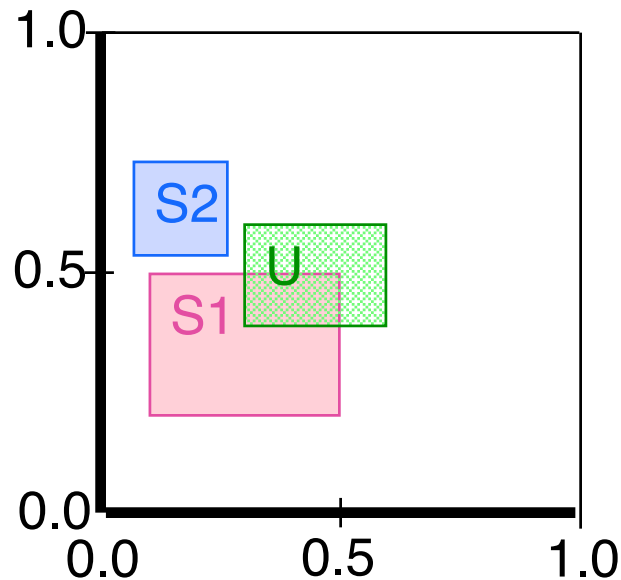


- Federate 1 (sensor): subscribe to S1
- Federate 2 (sensor): subscribe to S2

Interest expressions

- Subscription region in routing space ($S1=[0.1,0.5], [0.2,0.5]$)
- Specifies portion of routing space of interest to federate

HLA Data Distribution Management



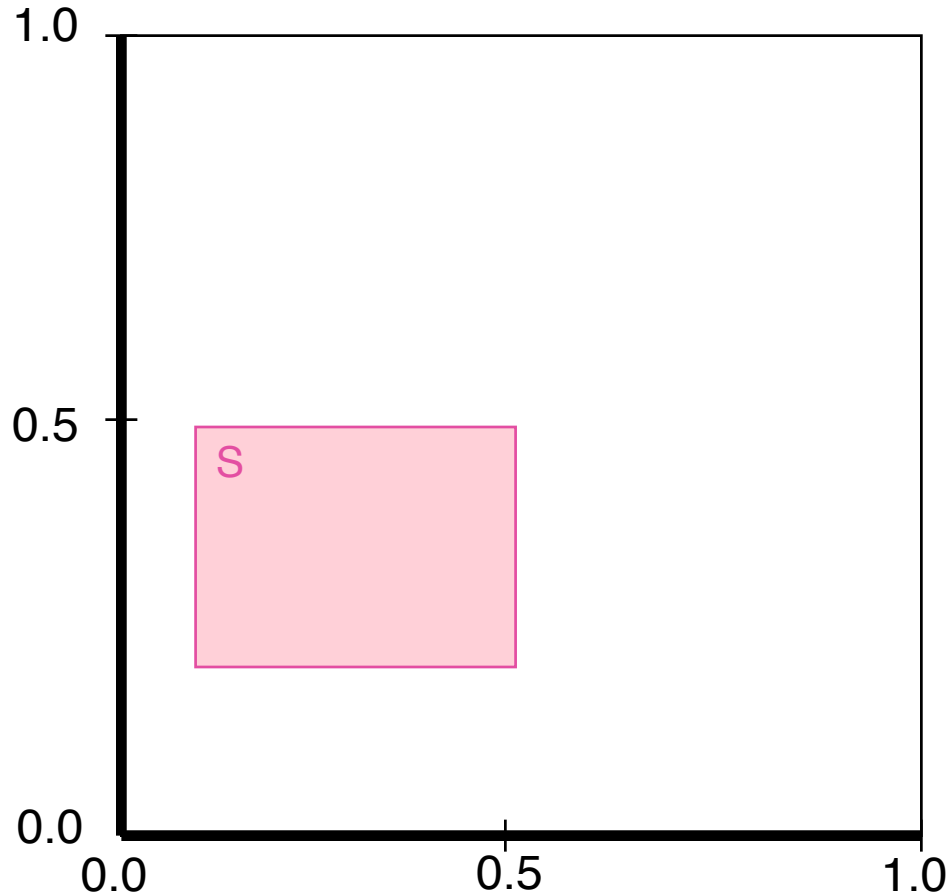
- Federate 1 (sensor): subscribe to S1
- Federate 2 (sensor): subscribe to S2
- Federate 3 (target): update region U

update messages by target are sent to federate 1, but not to federate 2

Description expressions

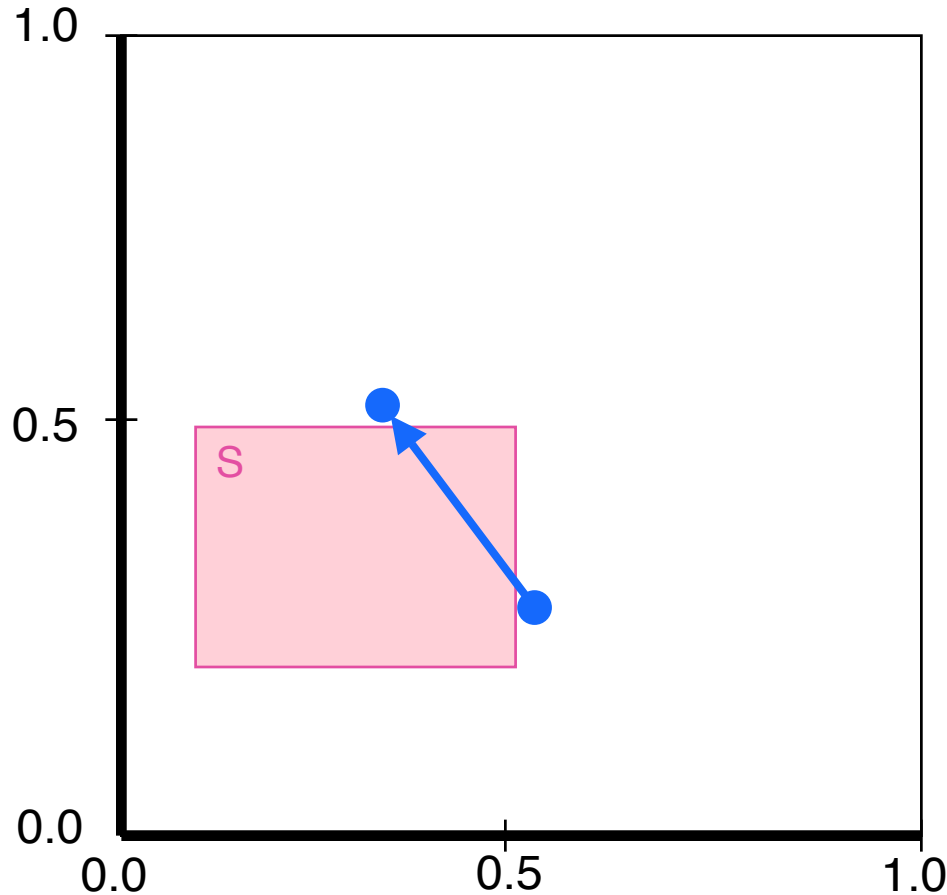
- Update region in routing space (U)
- Associated an update region with **each** attribute update
- a federate receives a message if
 - » It has subscribed to the attribute(s) being updated, and
 - » its subscription region overlaps with the update region

Update Regions vs Points



- Routing space represents playbox
- Subscription region represents sensor
- Updates correspond to position of a moving vehicle

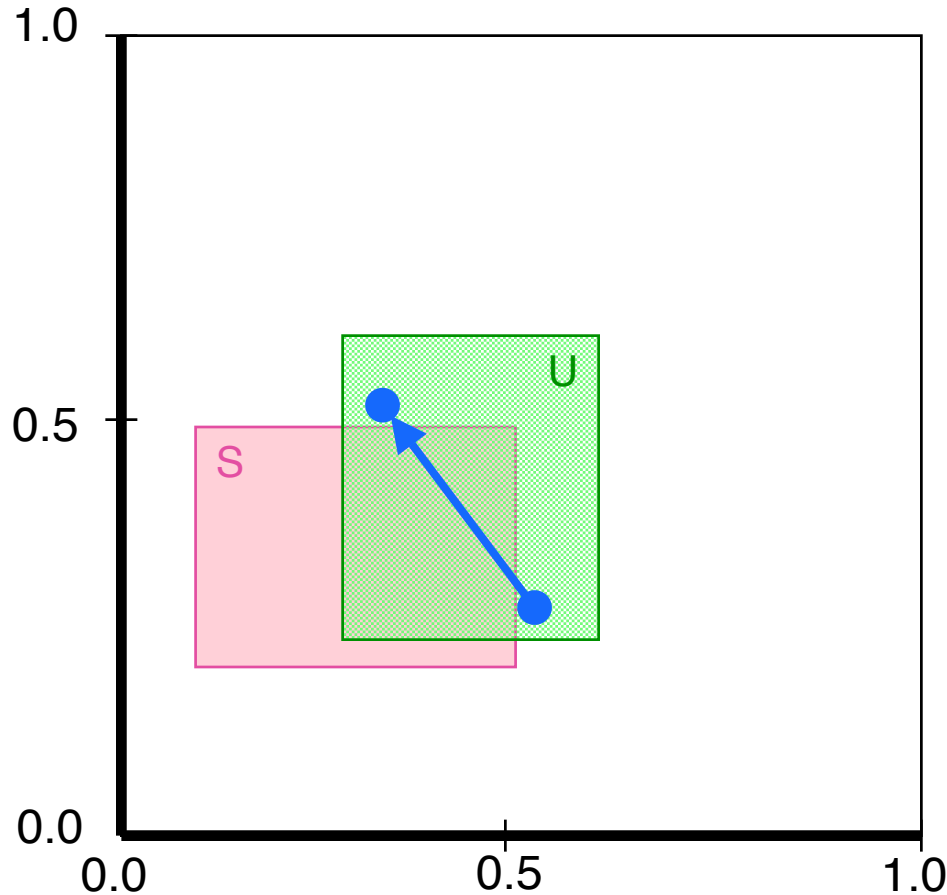
Update Regions vs Points



- Routing space represents playbox
- Subscription region represents sensor
- Updates correspond to position of a moving vehicle

● **Update points:** Sensor not notified of vehicle

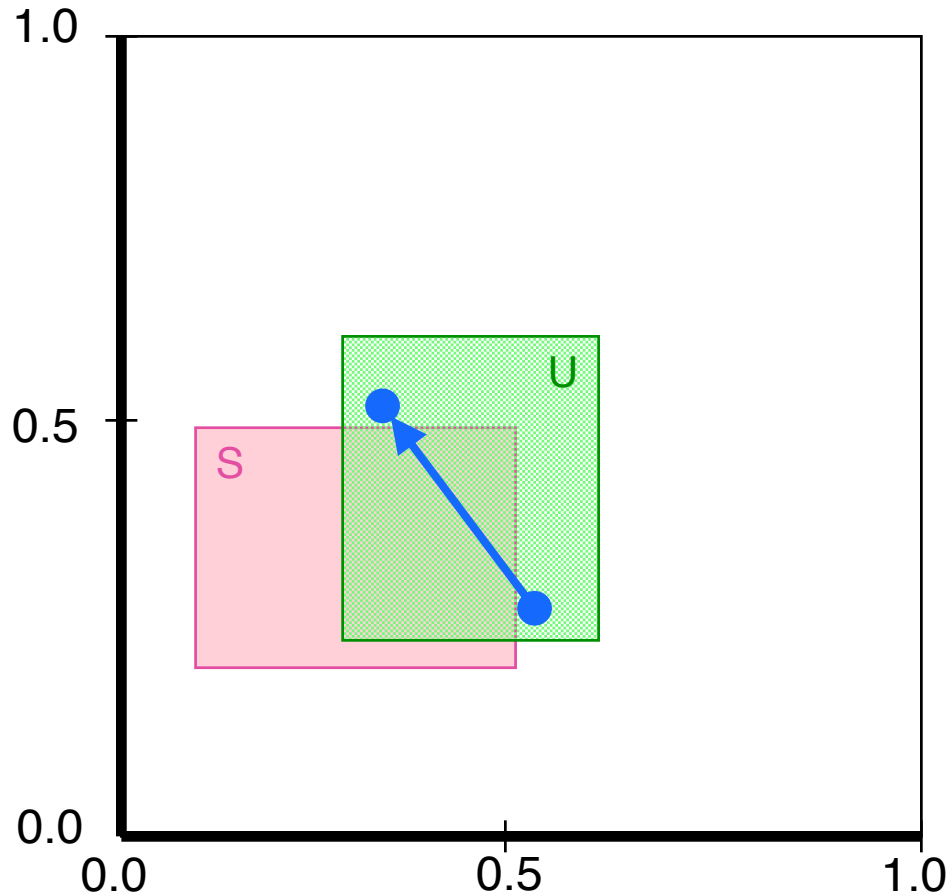
Update Regions vs Points



- Routing space represents playbox
- Subscription region represents sensor
- Updates correspond to position of a moving vehicle

- **Update points:** Sensor not notified of vehicle
- **Update regions:** Sensor is notified of vehicle

Filtering Precision



- Vehicle out of range, but updates are still routed to sensor federate
- Messages must be filtered at the receiver
- Sensor range may not be rectangular

In general, DDM is a compromise among:

- Filtering accuracy
- Implementation considerations (mapping to multicast groups)
- Ease of use

HLA DDM Services

Routing spaces (name space) and regions

- Define routing spaces in federation initialization file
- Create region, Modify Region, Delete Region
- Used for both subscription and update regions

Subscription regions (interest expressions)

- Subscribe/Unsubscribe Object Class Attributes with Region
- Used in addition to class-based filtering

Update regions (description expressions)

- Register Object Instance with Region or Associate Region with Updates
- Unassociate Regions for Updates
- Update Attribute Values

Outline

- **Introductory Comments**
- **Dynamic (Value based) Data Distribution: HLA Data Distribution Management**
 - » Routing space
 - » Publication Region
 - » Subscription Region
- **DDM Implementation**
 - » Cell-Based
 - » Region-Based
 - » Combining Cells and Regions

Communication Services

Recall:

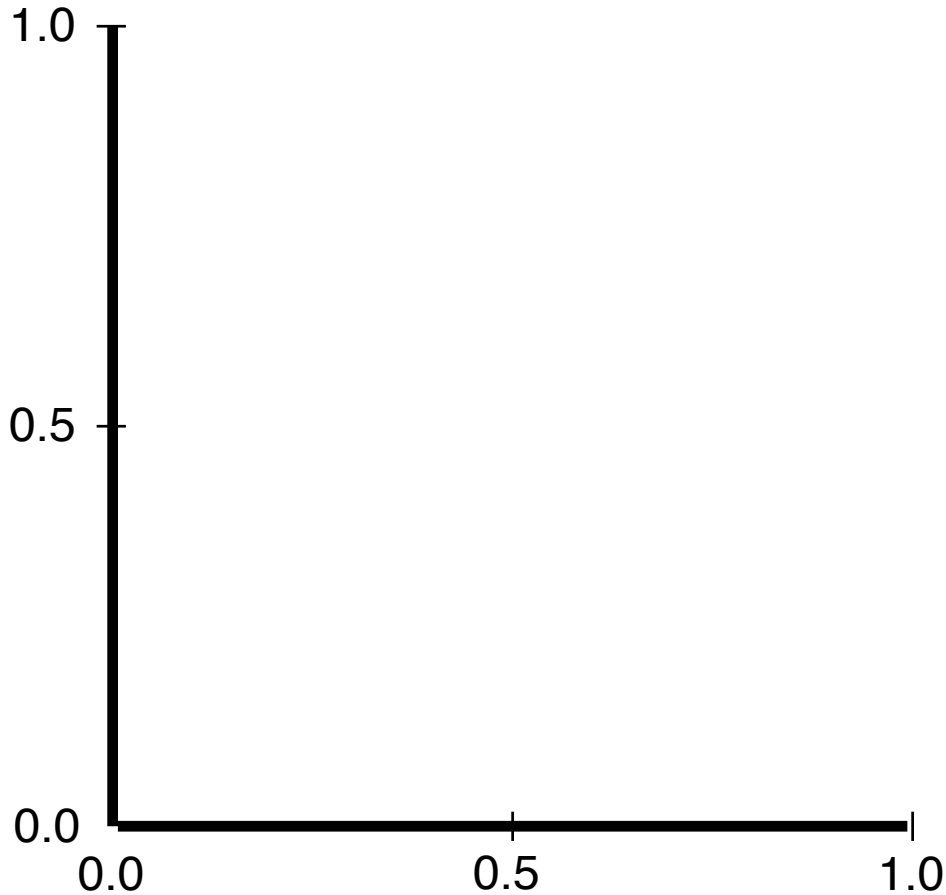
- **Unicast: point-to-point communication**
- **Broadcast: send to all**
- **Multicast: send to multiple destinations (not necessarily all)**
 - » **Multicast group**
 - » **Join group, Leave group**
 - » **Send: transport message to every member of group**

Data distribution software must map name space, interest expressions, and description expressions to group communication services

Implementation Approach

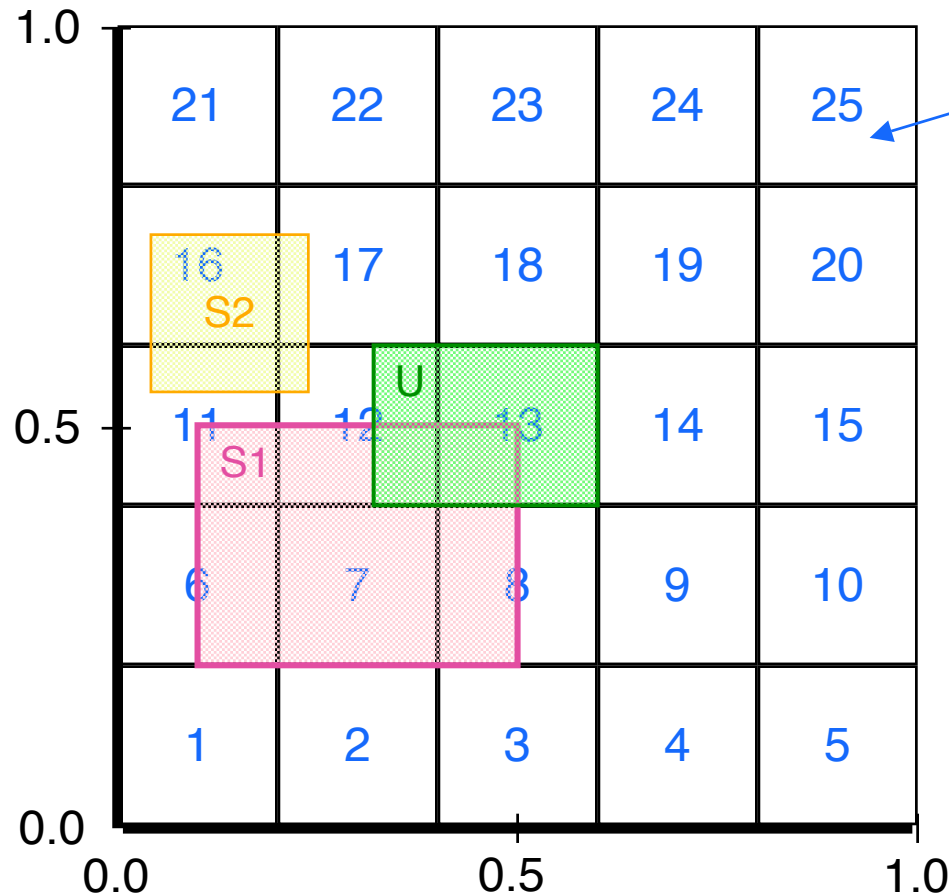
- **Map name space to multicast groups**
 - » Not all points in name space need be mapped to groups
 - » A point in name space could map to **multiple** groups
- **Interest expression**
 - » Interest expression defined as points of name space
 - » Join groups that overlap with interest expression
- **Description expression**
 - » Description expression defined as points in name space
 - » Send messages to groups that overlap with description expression

Grid-Based Implementation



- partition routing space into grid cells, map each cell to a multicast group
- subscription region: *Join* each group overlapping subscription region
- attribute update: send *Update* to each group overlapping update region
- need additional filtering to avoid unwanted messages, duplicates

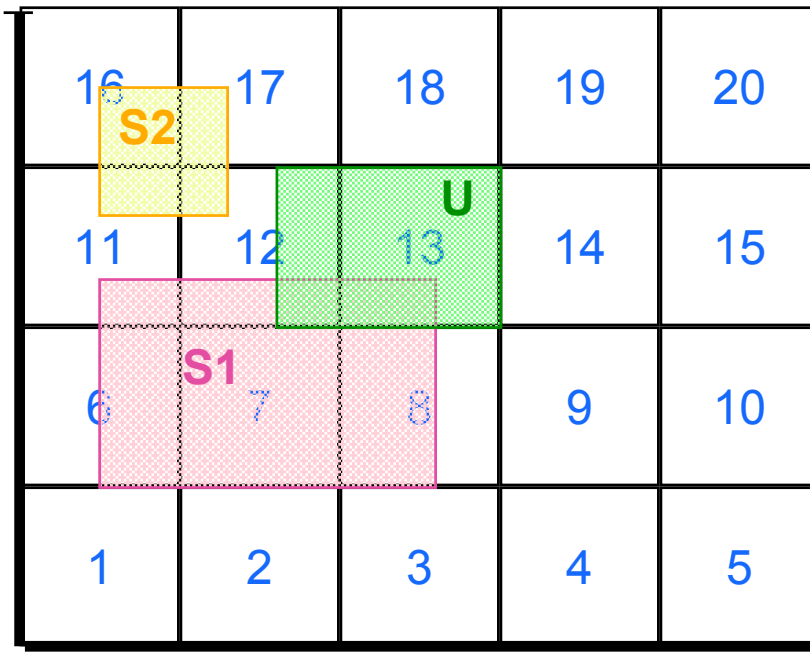
Grid-Based Implementation



- S1 subscribes to 6,7,8,11,**12,13**
- S2 subscribes to 11,**12**,16,17
- U publishes to **12, 13**
- Unwanted messages to S2
- Duplicate messages to S1

- partition routing space into grid cells, map each cell to a multicast group
- subscription region: *Join* each group overlapping subscription region
- attribute update: send *Update* to each group overlapping update region
- need additional filtering to avoid unwanted messages, duplicates

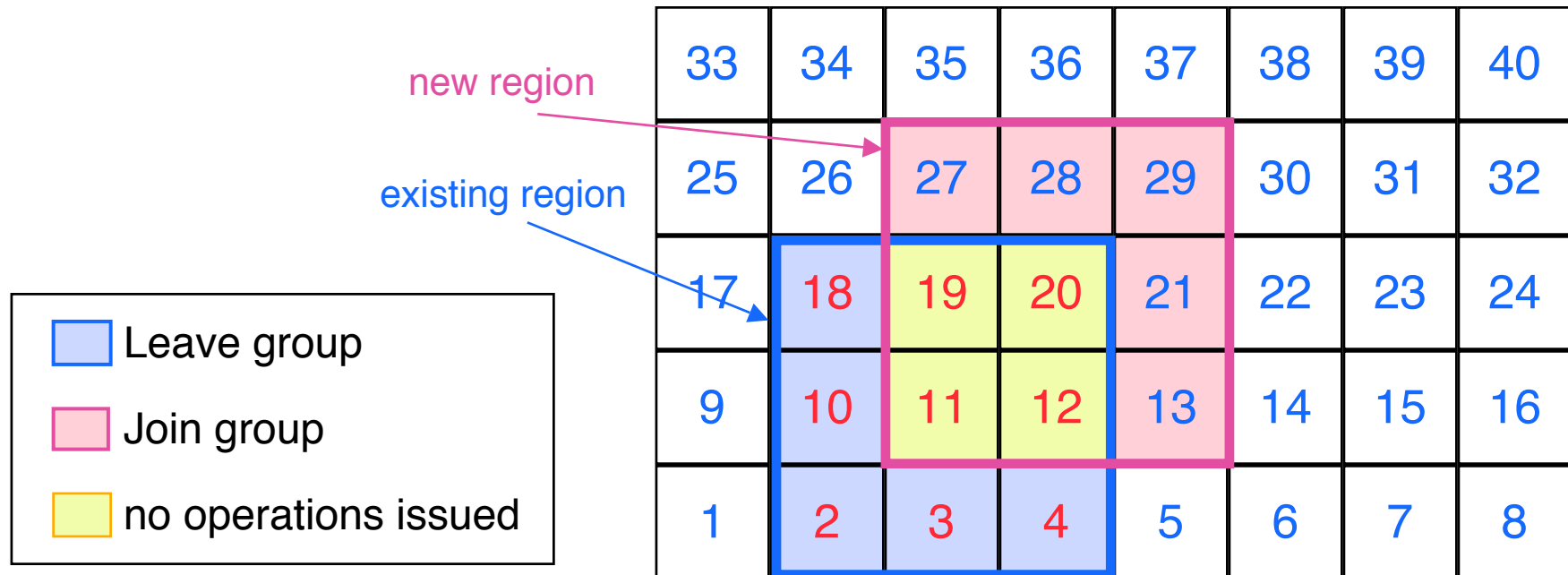
Additional Messages



- S1: Subscription Region
- S2: Subscription Region
- U: Update Region

- F1 subscribes to 6, 7, 8, 11, 12, 13 (F1's interest expression **S1**)
- F2 subscribes to 11, 12, 16, 17 (**S2**)
- Updates associated with publication region (description expression **U**) ⇒ a message sent to 12 & 13.
- **Irrelevant Messages**: F2 receive message too because of group 12 (interest and description regions does not map perfectly to multicast groups).
- **Duplicate Messages**: F1 receives 2 copies because message is sent to both 12 and 13 (receiver filtering).

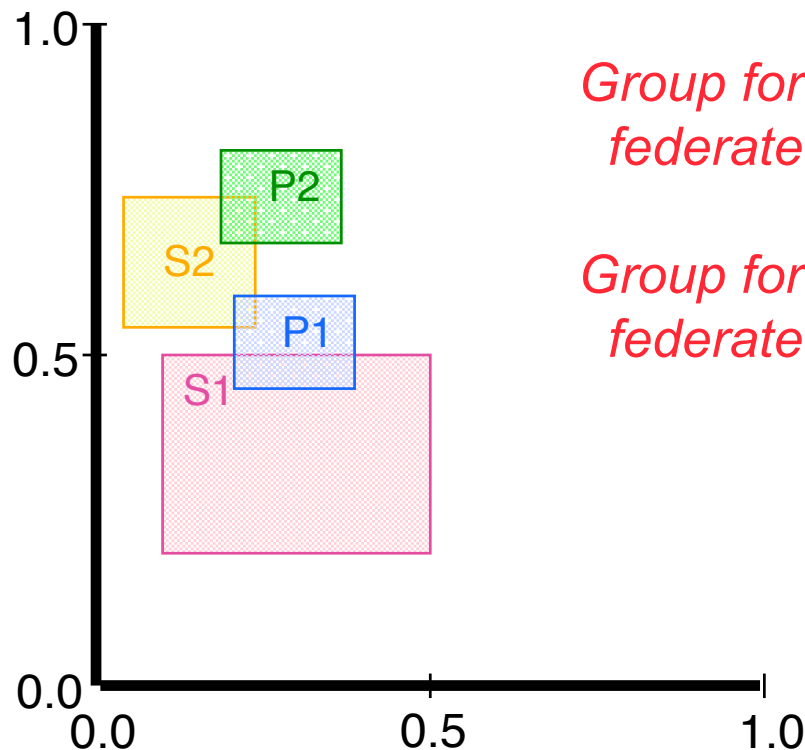
Changing a Subscription Region



- issue Leave operations for (cells in old region - cells in new region)
- issue Join operations for (cells in new region - cells in old region)

Approach 2: Region-Based Groups

- Define one multicast group per **publication** region (uses matching)
 - » Publishing federates that are in intersecting q subscription region are forced to join multicast groups and to send data to that group.
- **Group membership**: Any federate subscribed to a region that overlaps the publication region is a member of group
- **Update**: Send message to group associated with publication region



Group for P1:

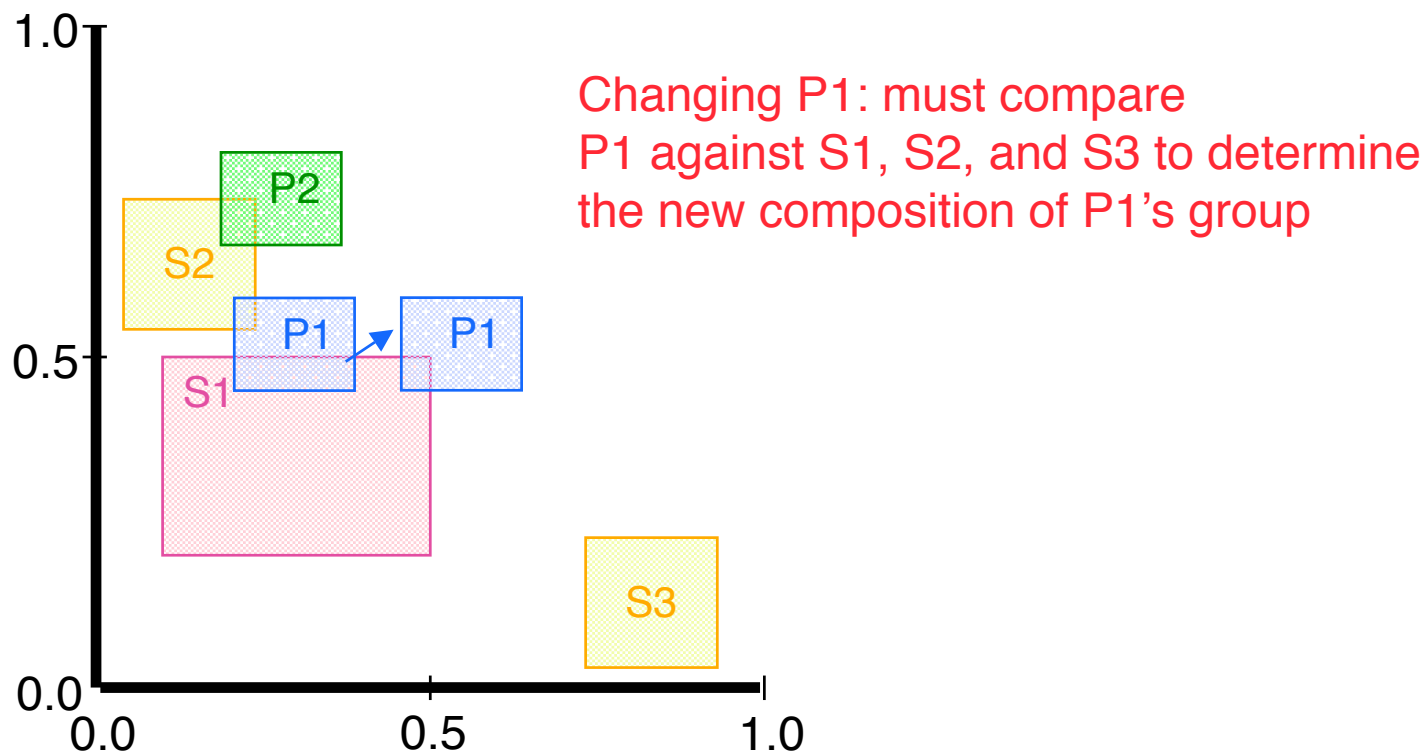
federates subscribed to regions S1, S2

Group for P2:

federates subscribed to region S2

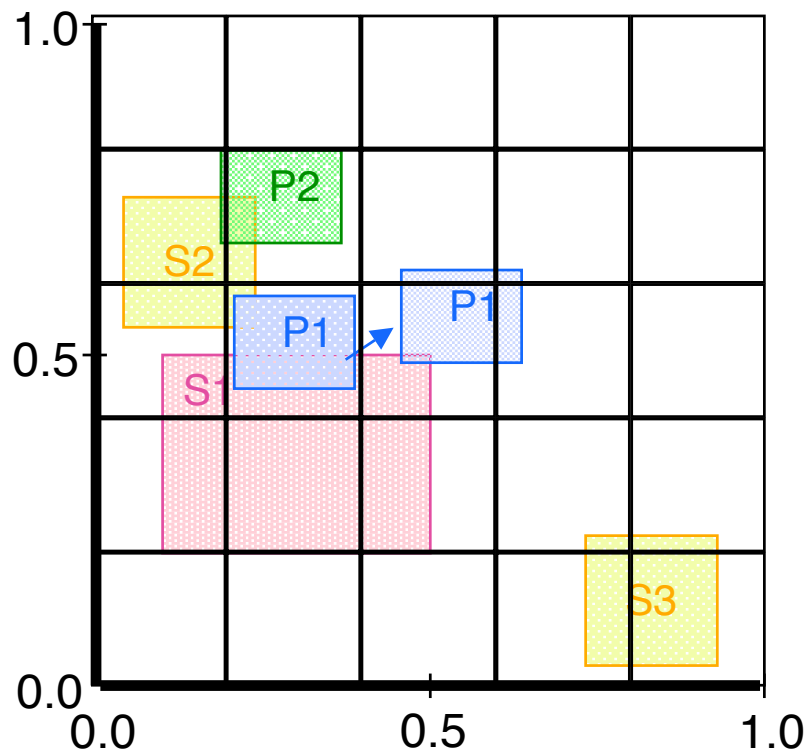
Approach 2: Region-Based Groups

- When a subscription region changes, it must be compared against **all** publication regions to determine if the federate should join/leave multicast groups
- When a publication region changes, it must be compared against **all** subscription regions to determine the new composition of its multicast group
- Not scalable to large numbers of regions



Approach 3: Regions with Grids

- A group is defined for each publication region (same as region-based approach)
- A grid is superimposed over routing space
- **Matching**: need only check publication/subscription regions in the grid cell(s) overlapping the original and new regions



Changing P1: must compare P1 against S1 and S2, but need not compare against S3

Practical Problems

- **Limited number of multicast groups**
- **Fast movers: rapid joins and leaves**
 - » Join/leave times may be large
 - » Predict and initiate group operations in advance
- **Wide area viewers: too much traffic!**
 - » need less detailed information to reduce traffic
 - » Multiple routing spaces with different grid sizes and detail of information covering playbox

Summary

- **Data distribution management provides value-based filtering of data**
 - » **Dynamic interest, description expressions**
 - » **Design involves many tradeoffs**
 - Filtering efficiency
 - Ease of use
 - Implementation complexity
- **Implementation**
 - » **Map name space to multicast groups**
 - » **Map interest expressions to multicast group joins**
 - » **Map declaration expression to multicast group sends**
 - » **Interest expression changes map to group joins and leaves**