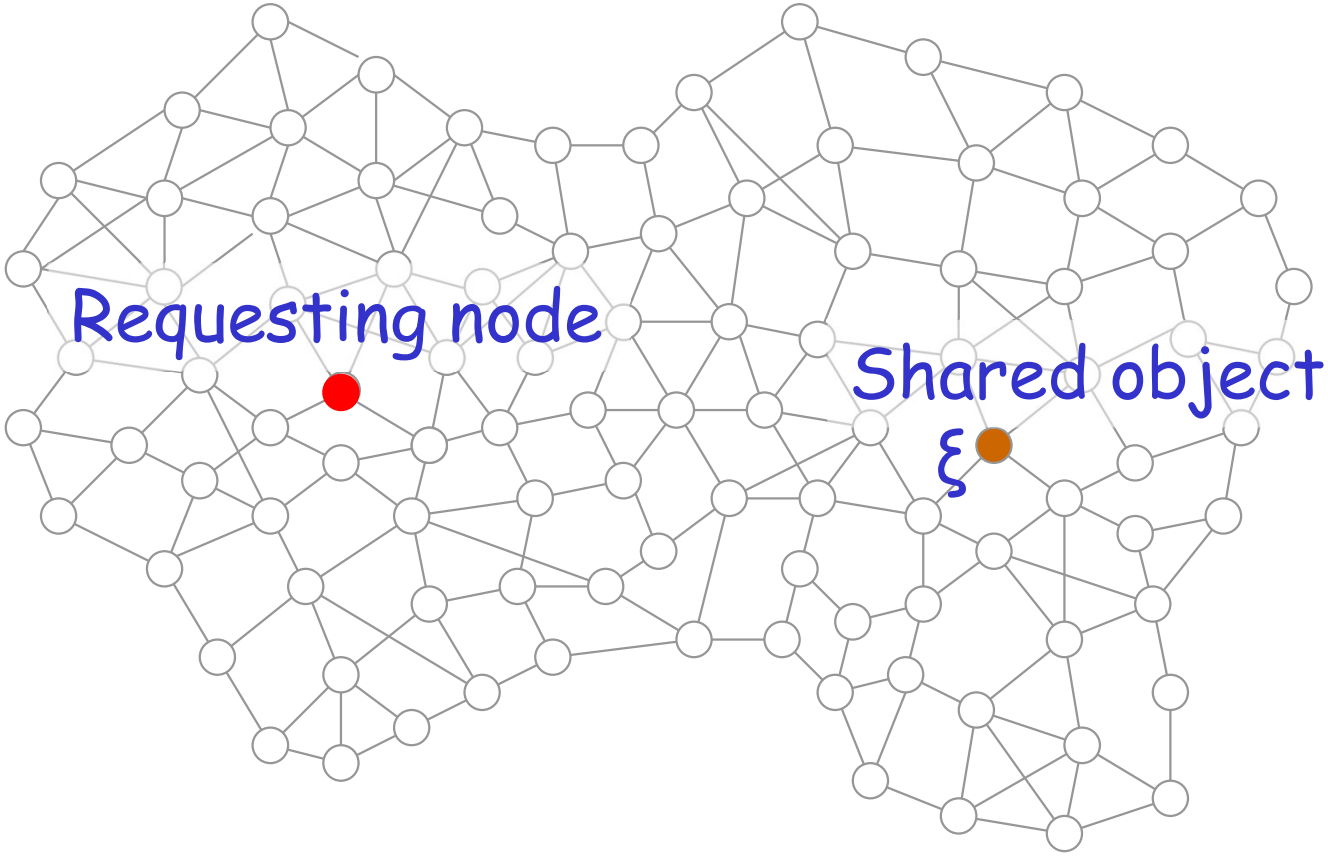
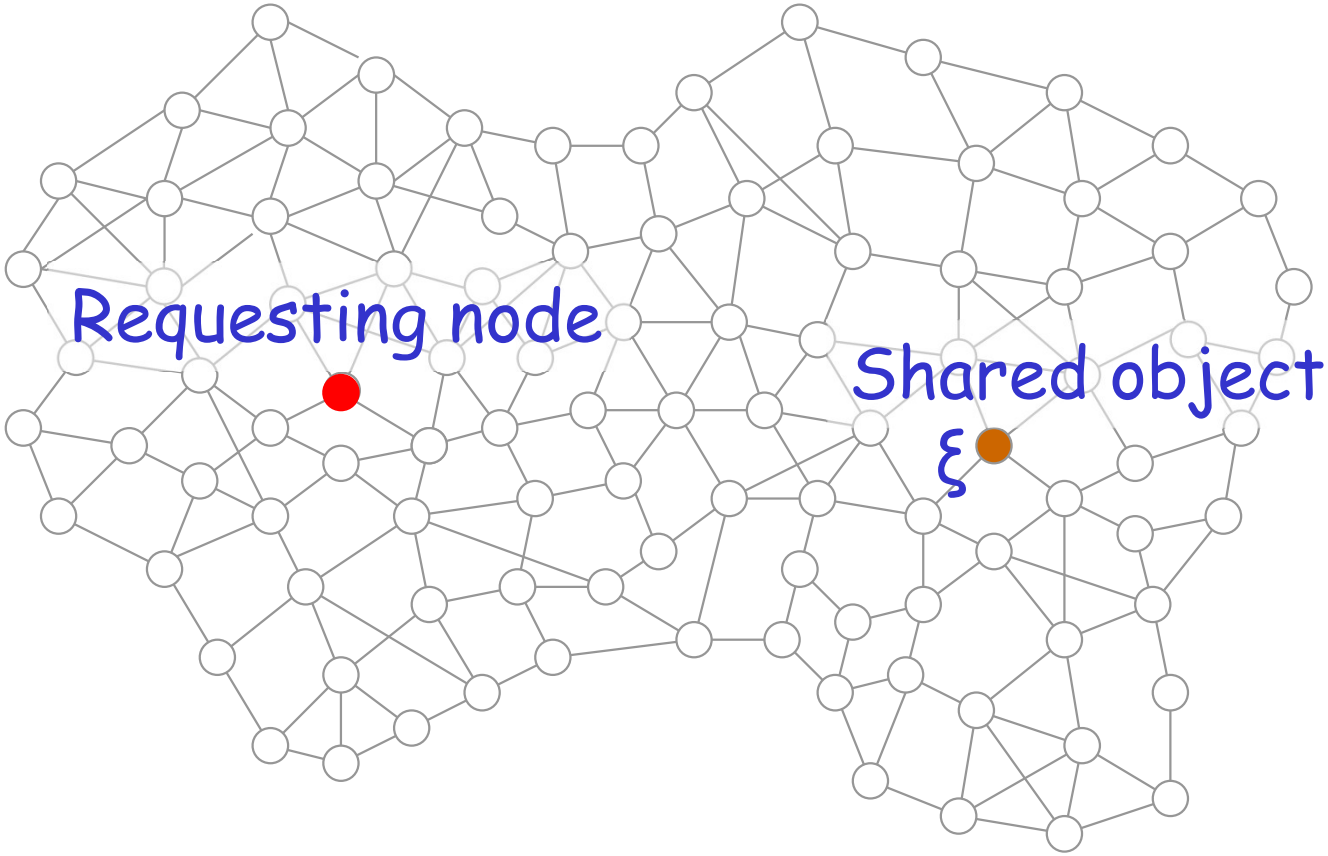
Learning-Augmented Distributed Directories

Swapnil Guragain, Bibek Maharjan, Sushant Bhattarai, Gokarna Sharma, and Pavan Poudel

Kent State University, USA
University of Houston-Clear Lake, USA

November 5, 2025

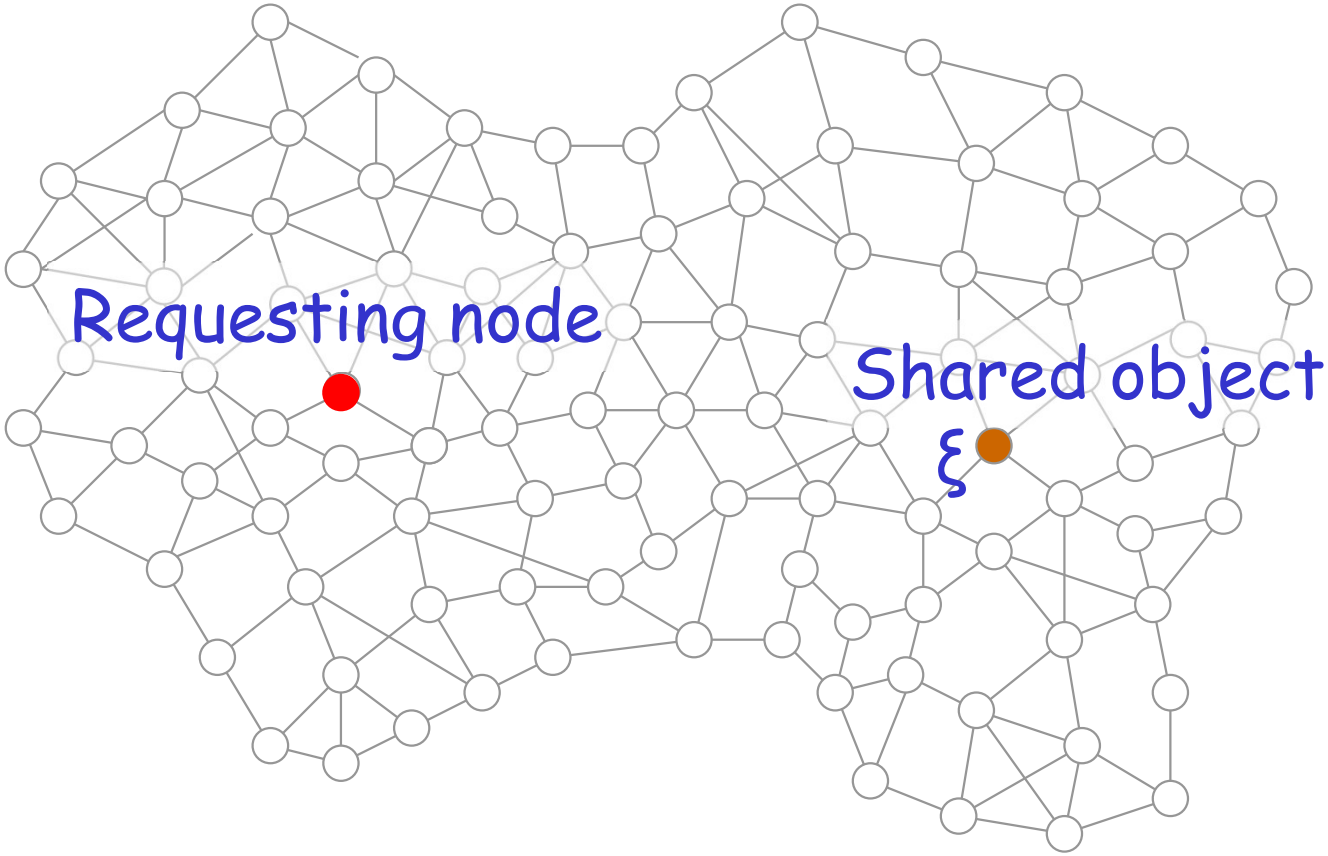

Requesting node
Shared object
ξ



Naïve Implementation #1: Inform all other nodes

Update Cost: O(m)

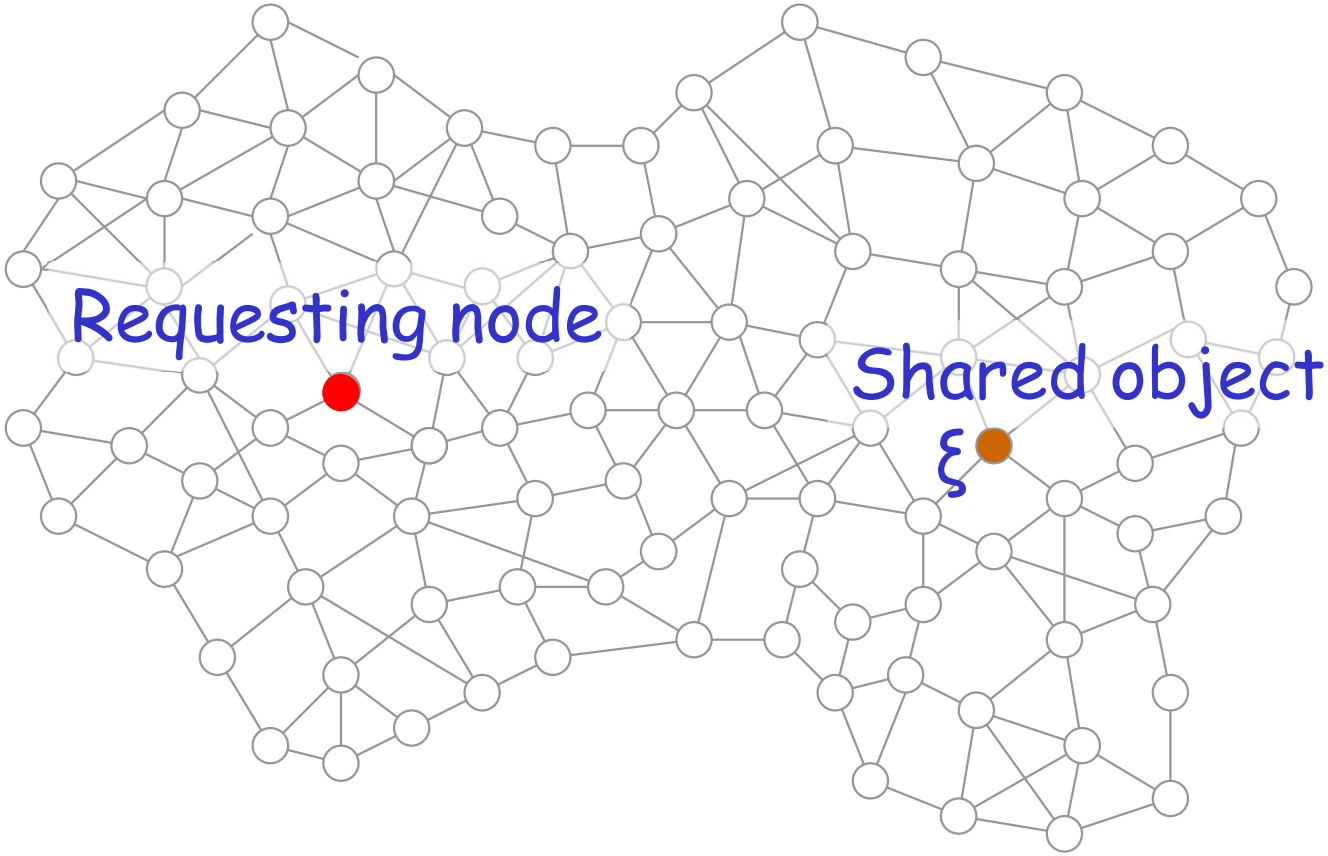
Lookup Cost: O(1)



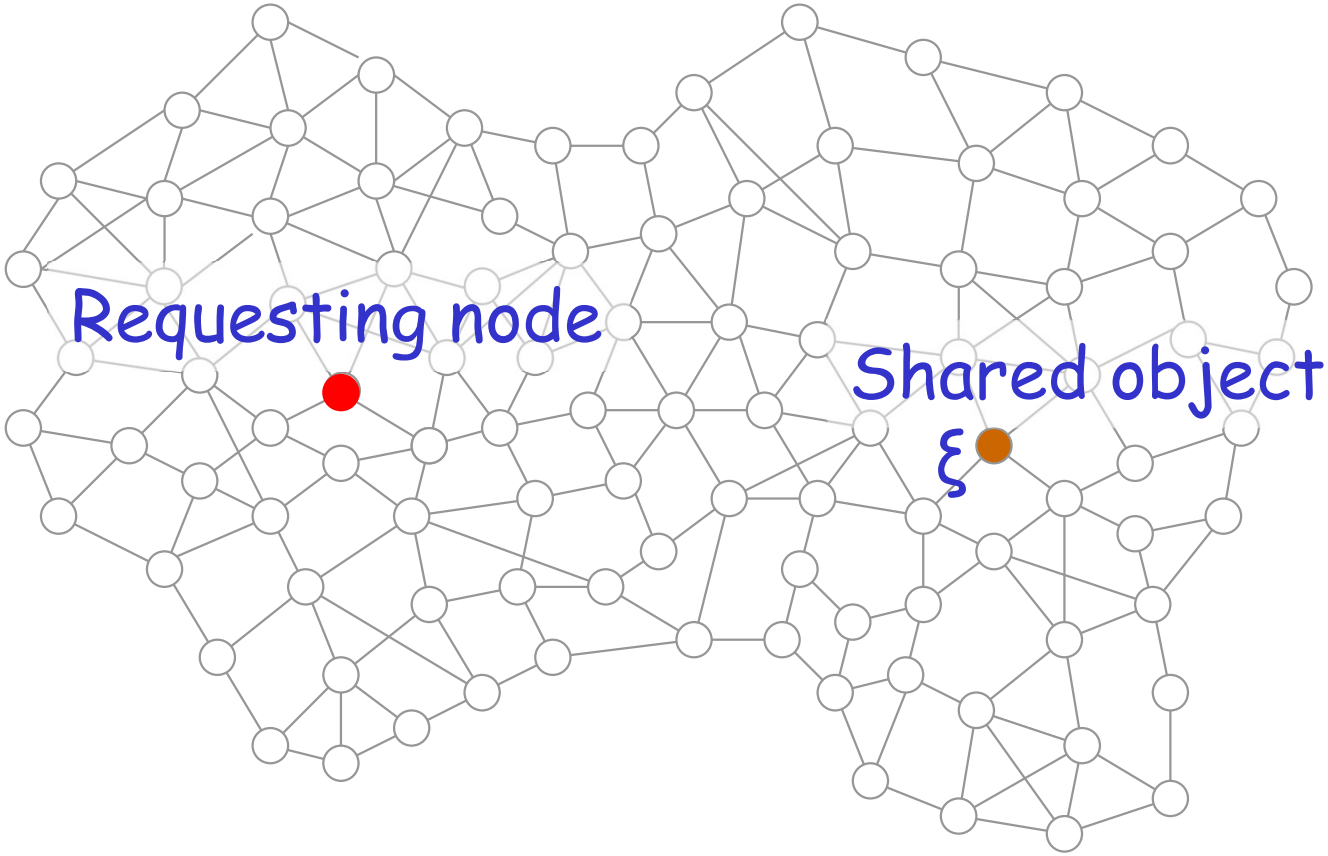
Naïve Implementation #2: Search the network

Update Cost: O(1)

Lookup Cost: O(m)



Question: Can both update and lookup costs be simultaneously minimized?



Optimal cost: Shortest path distance between requesting node and object node

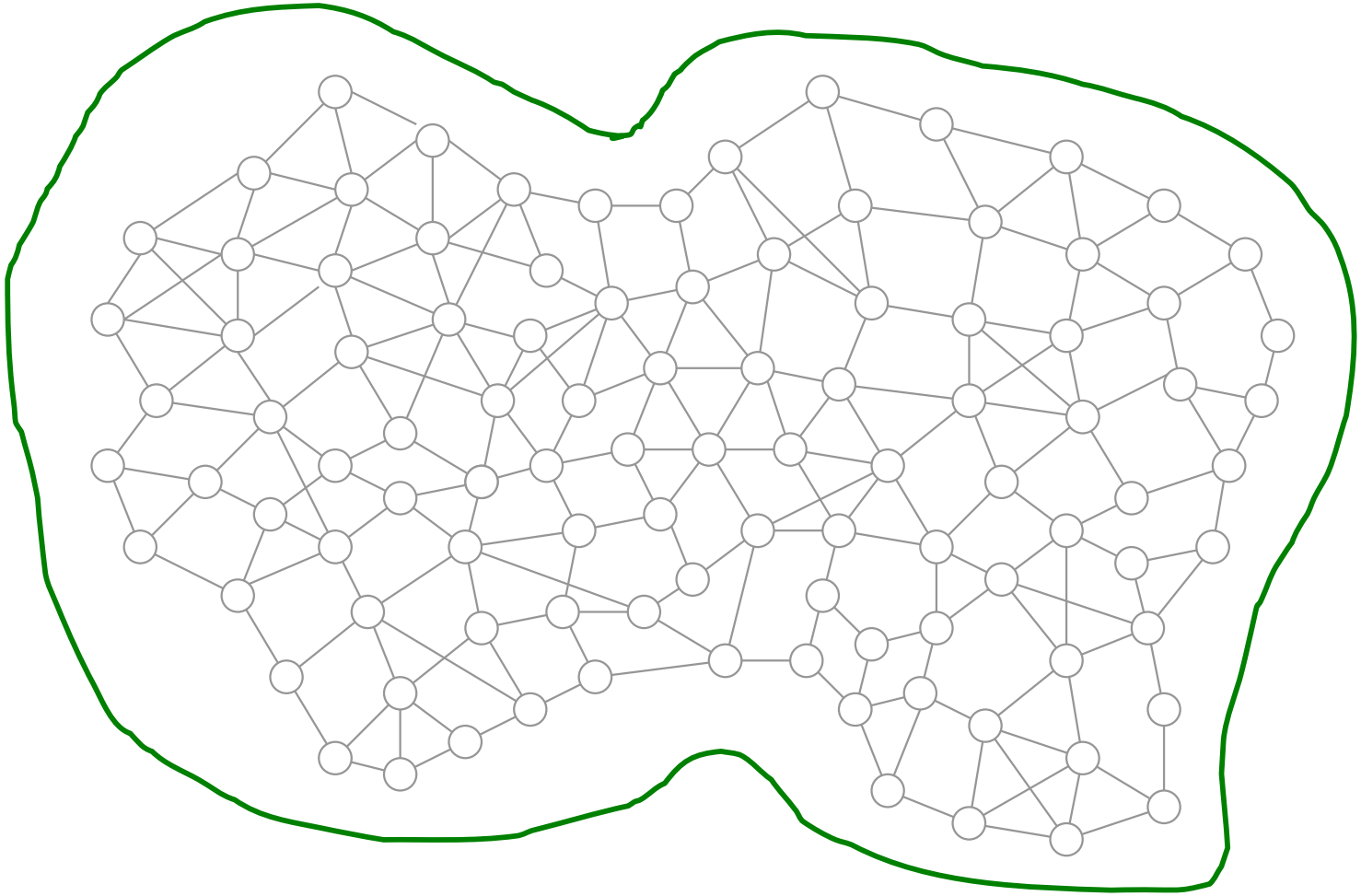
Protocol cost: Protocol's path distance from requesting node to object node

Existing Approaches

Protocol	Update Stretch	Network Kind	Runs on
Arrow [DISC'98]	$O(S_{ST})=O(D)$	General	Spanning tree
Relay [OPODIS'09]	$O(S_{ST})=O(D)$	General	Spanning tree
Combine [SSS'10]	$O(S_{OT})=O(D)$	General	Overlay tree
Ballistic [DISC'05]	$O(\log D)$	Constant-doubling	Hierarchical clustering with independent sets
Spiral [IPDPS'12]	$O(\log^2 n \log D)$	General	Hierarchical clusters

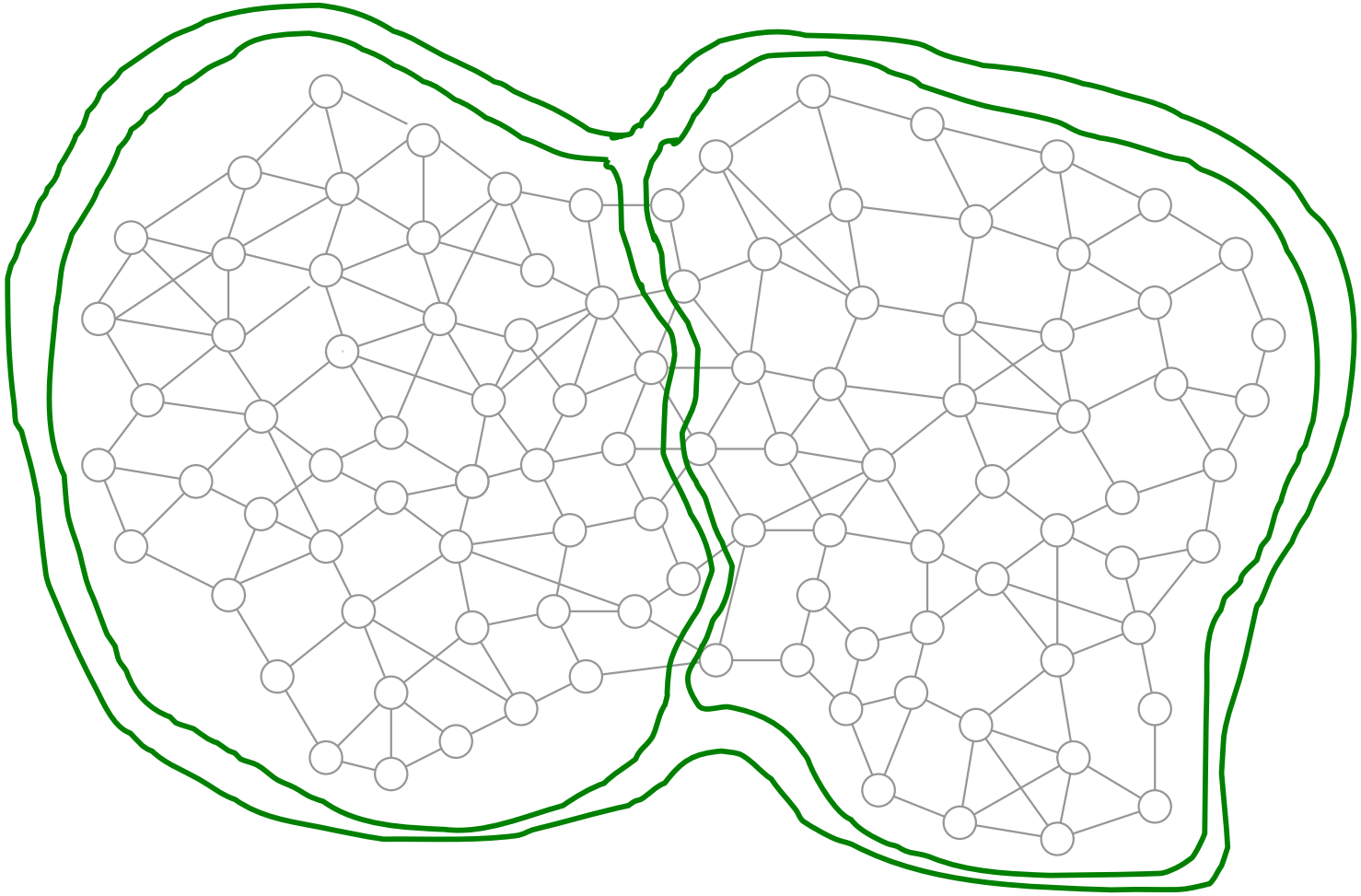
Stretch: Ratio of Protocol cost over Optimal cost

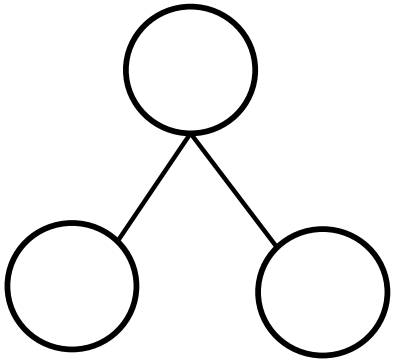
Approach: Hierarchical



Network graph

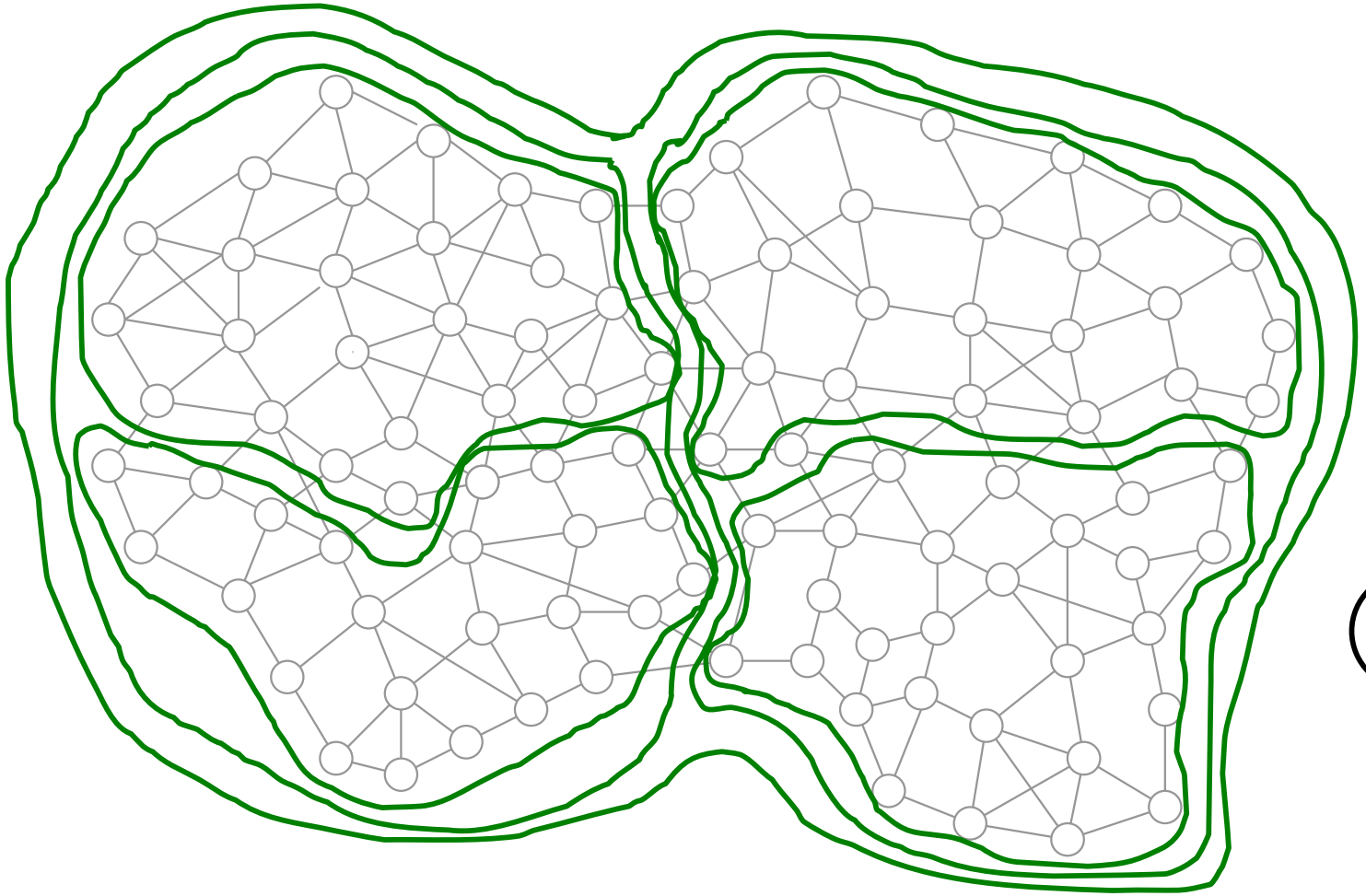
Approach: Hierarchical

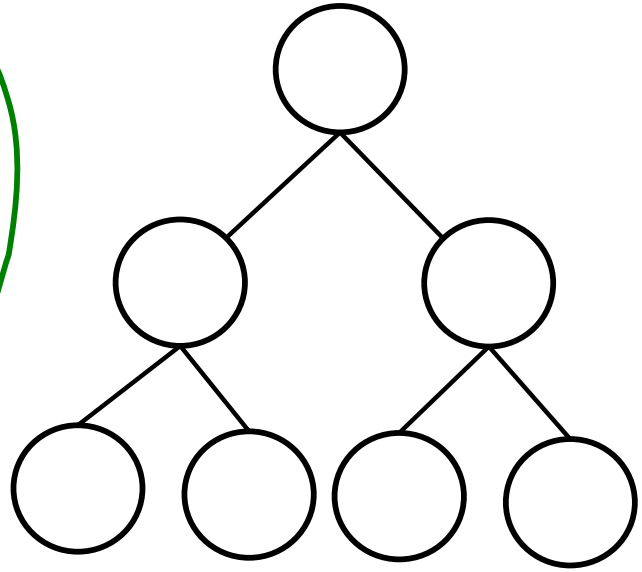




Alternative representation as a hierarchy tree with leader nodes

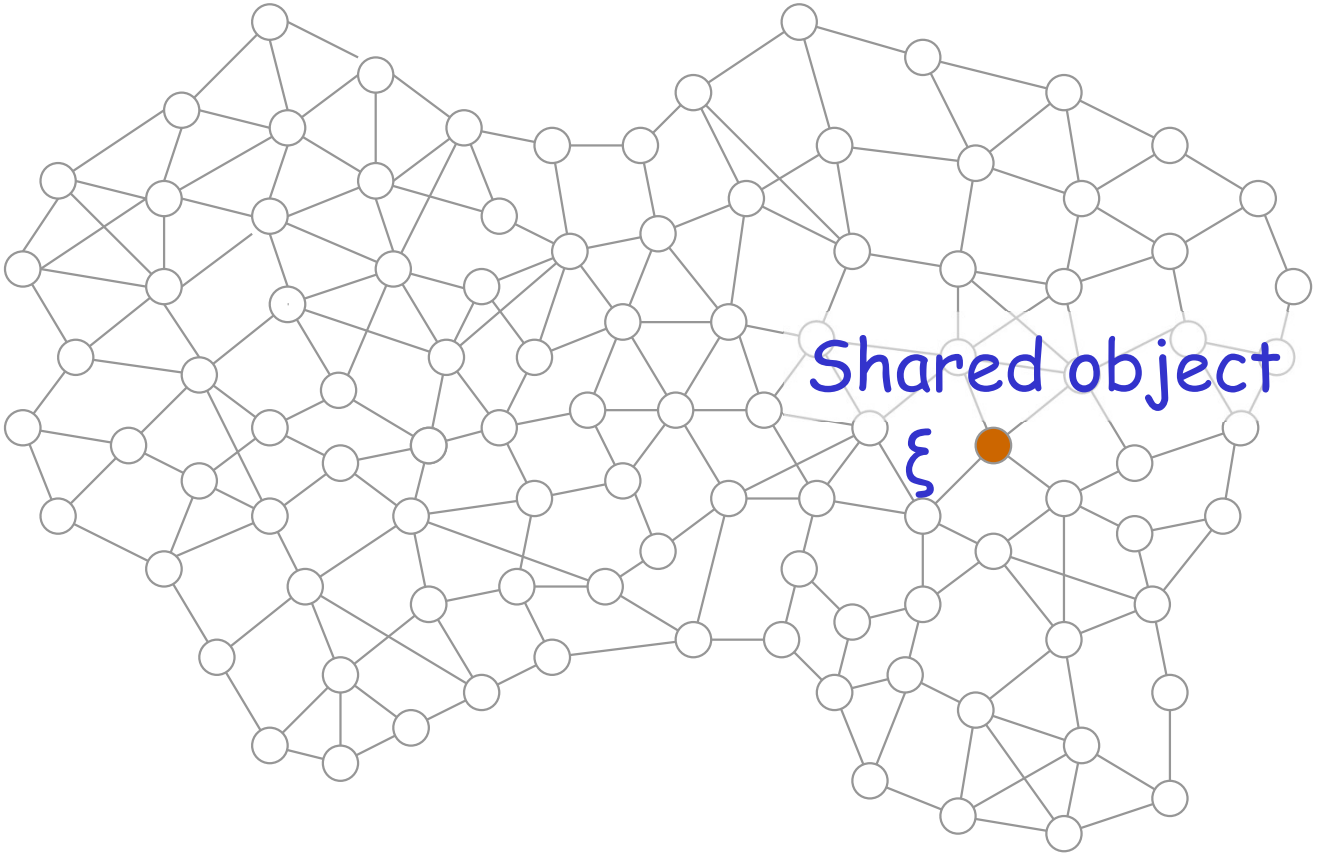
At the lowest level (level 0) every node is a cluster

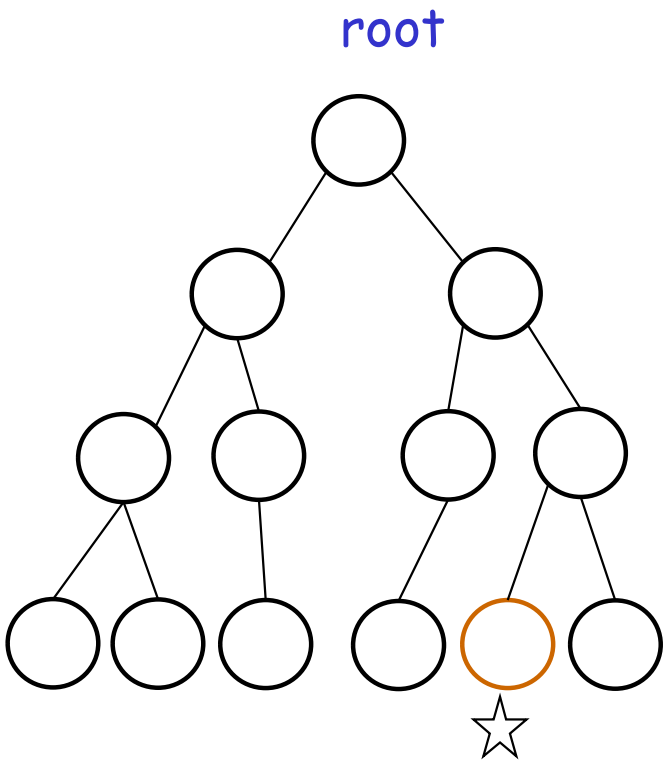




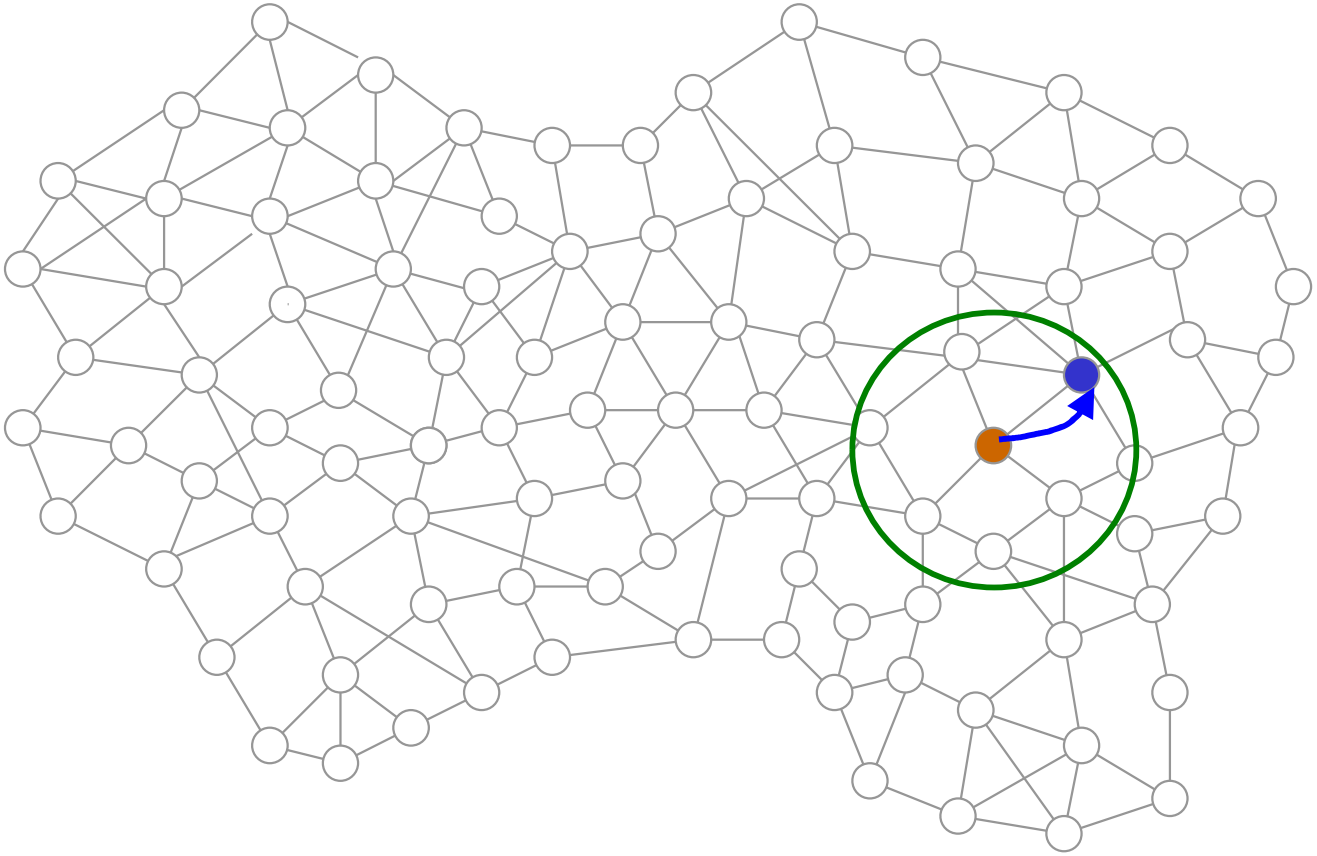
Directories at each level cluster, downward pointer if object locality known

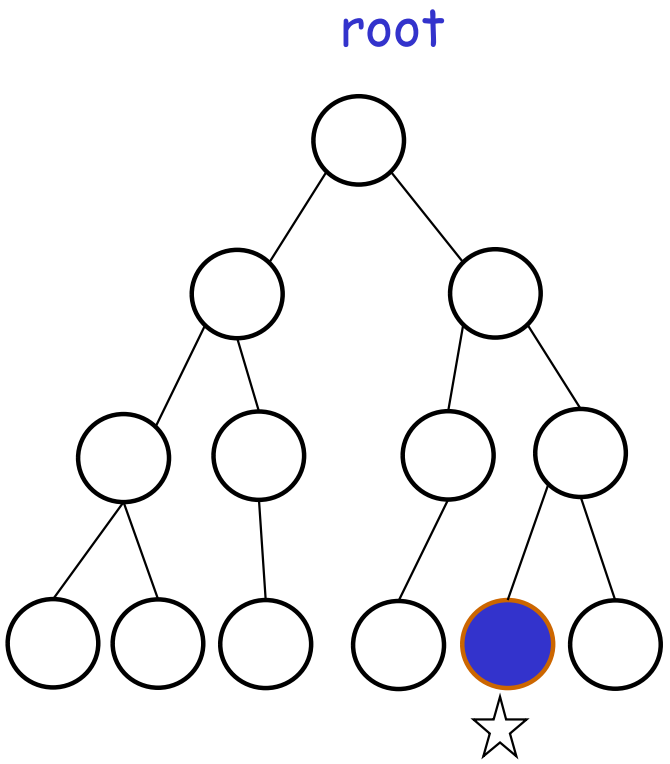
Publish object



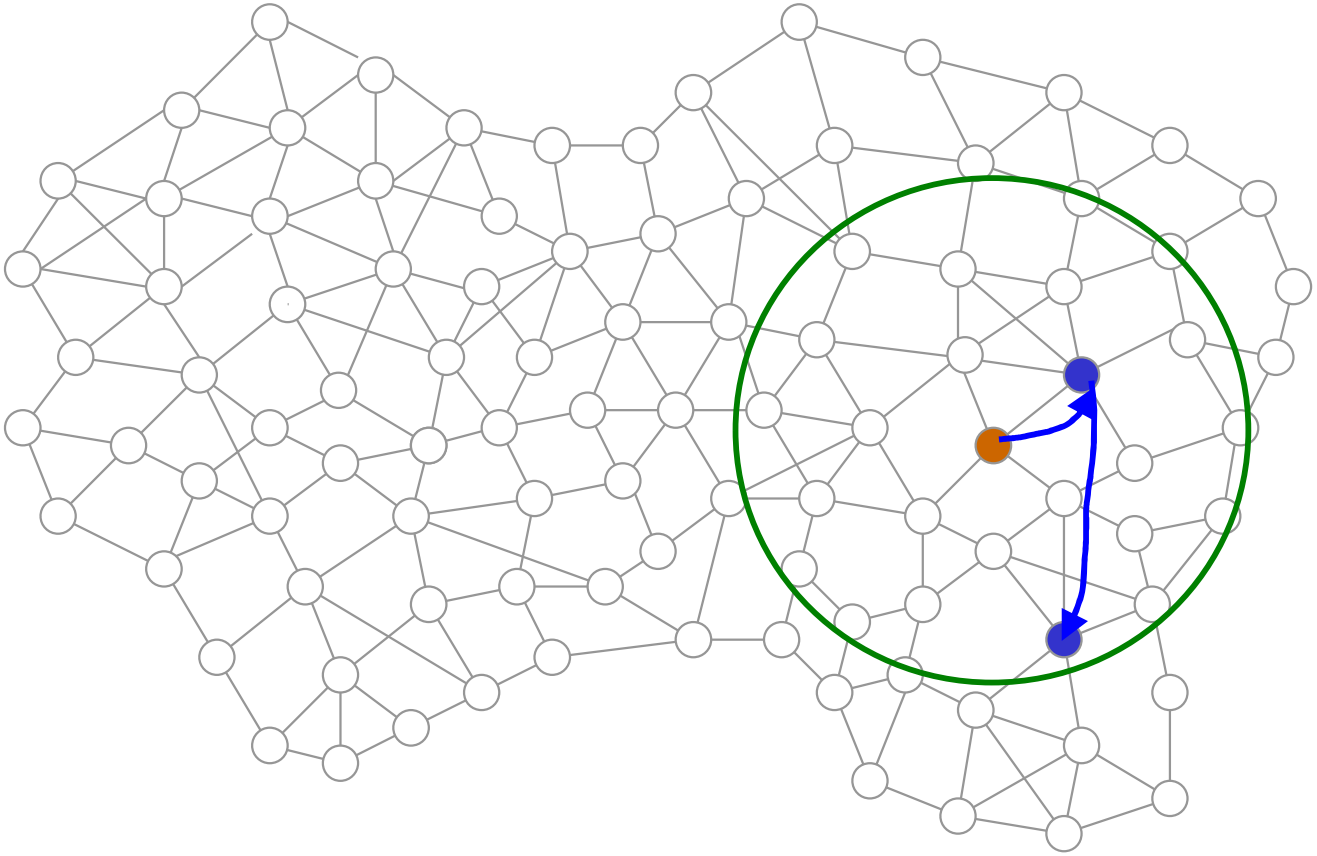


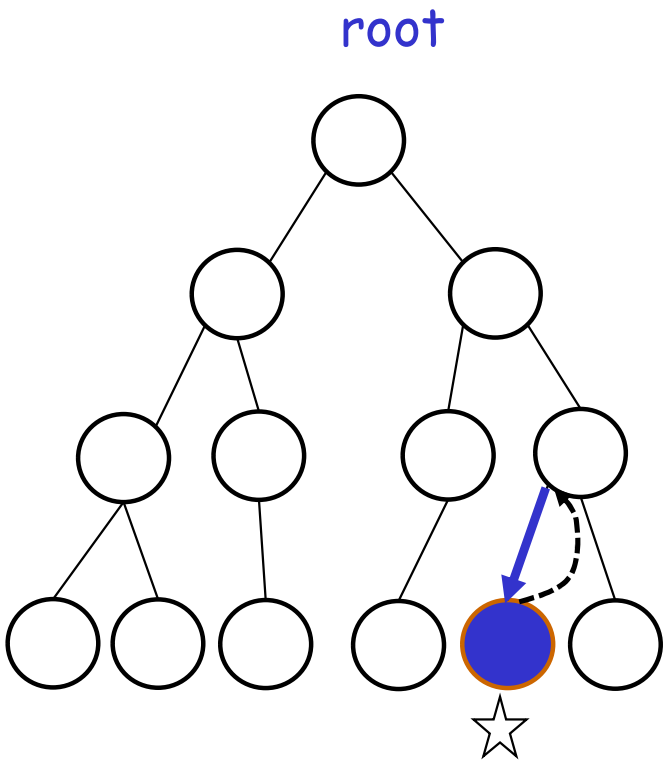
Send request to the leader





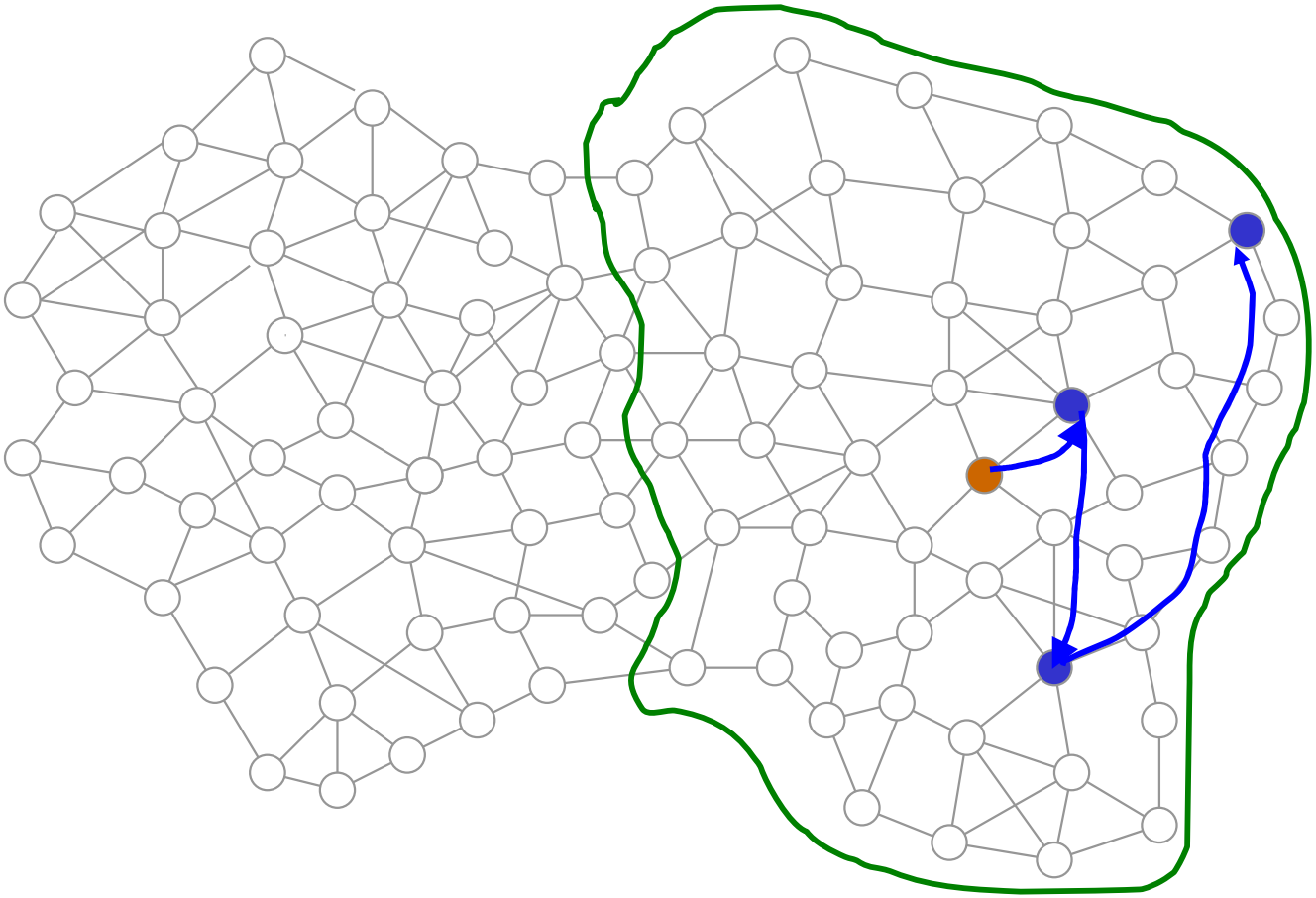
Continue up phase

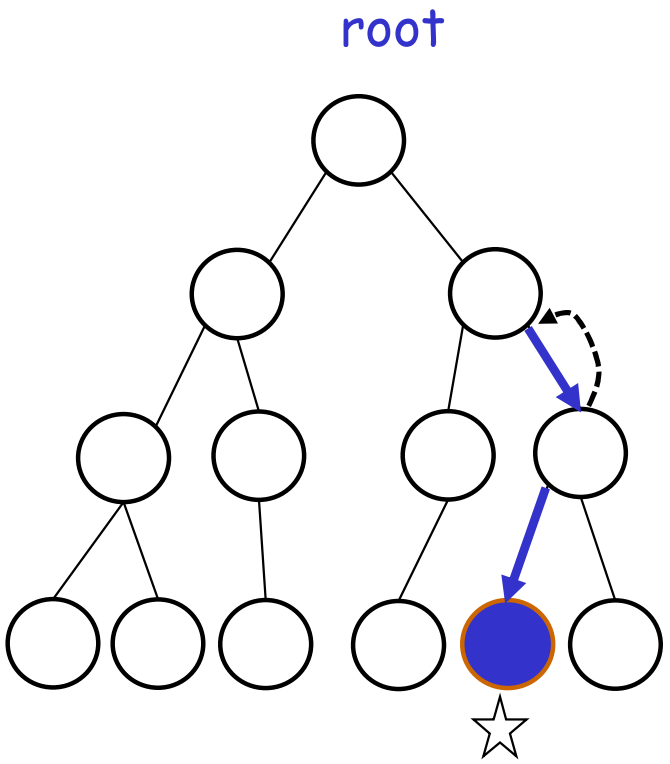




Set downward pointer while going up

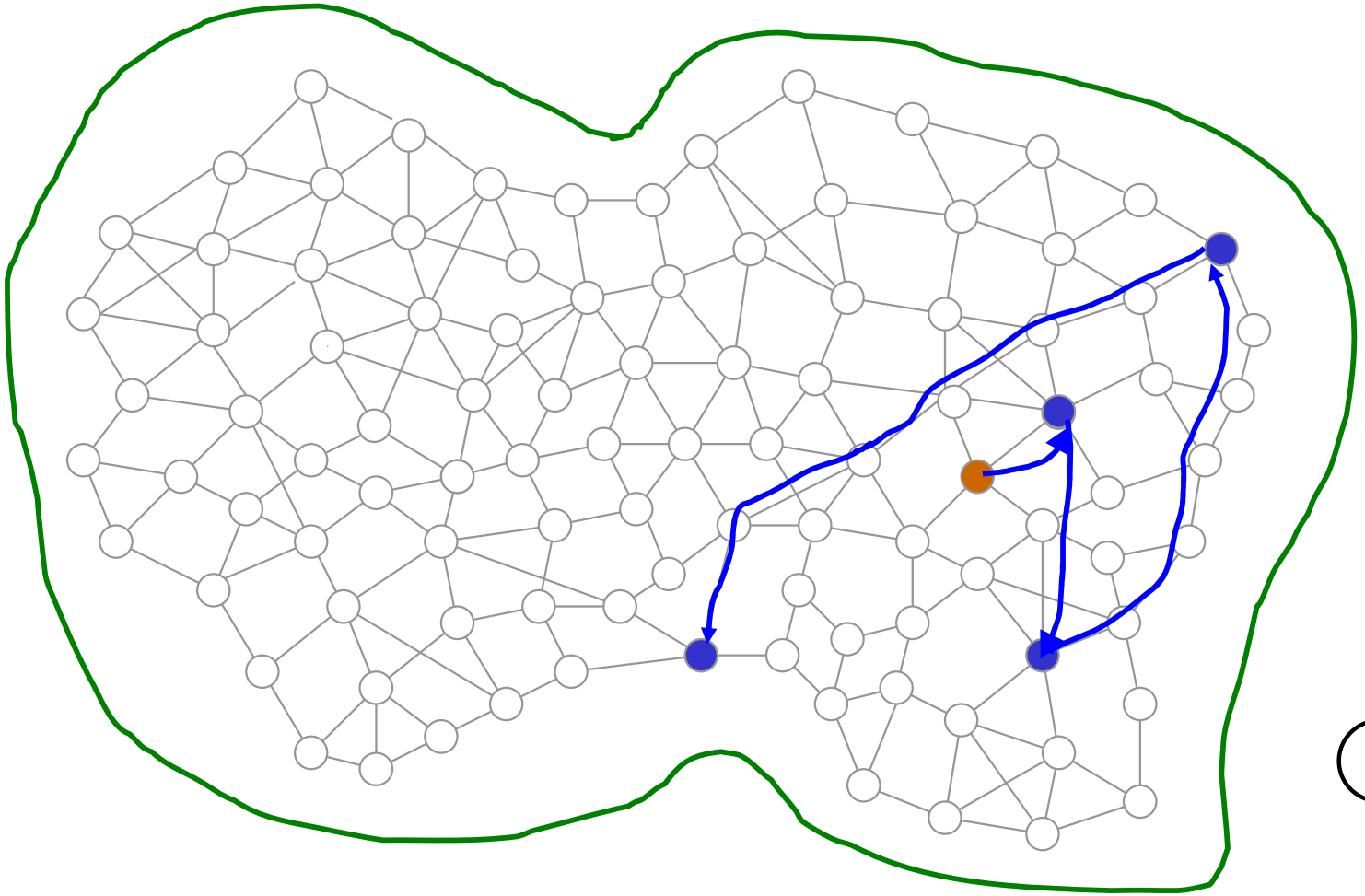
Continue up phase

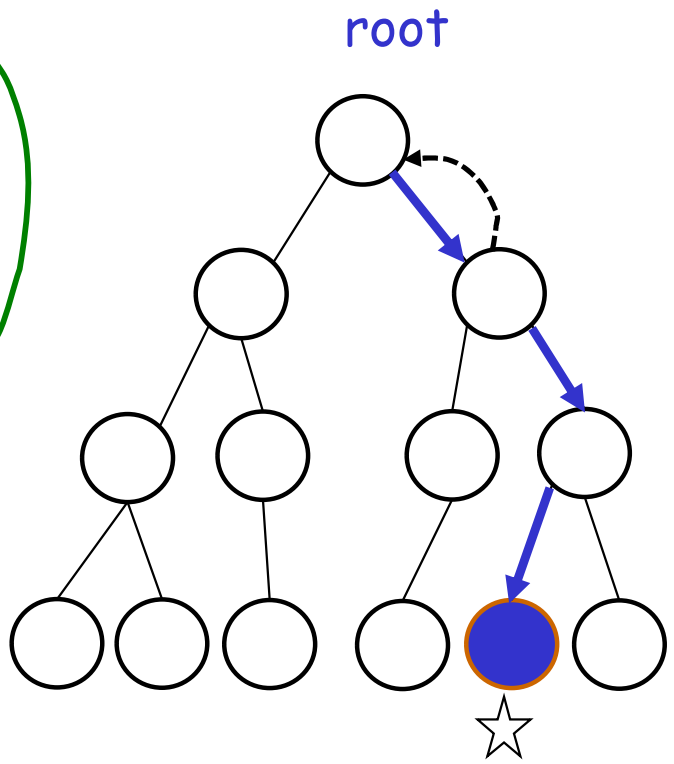


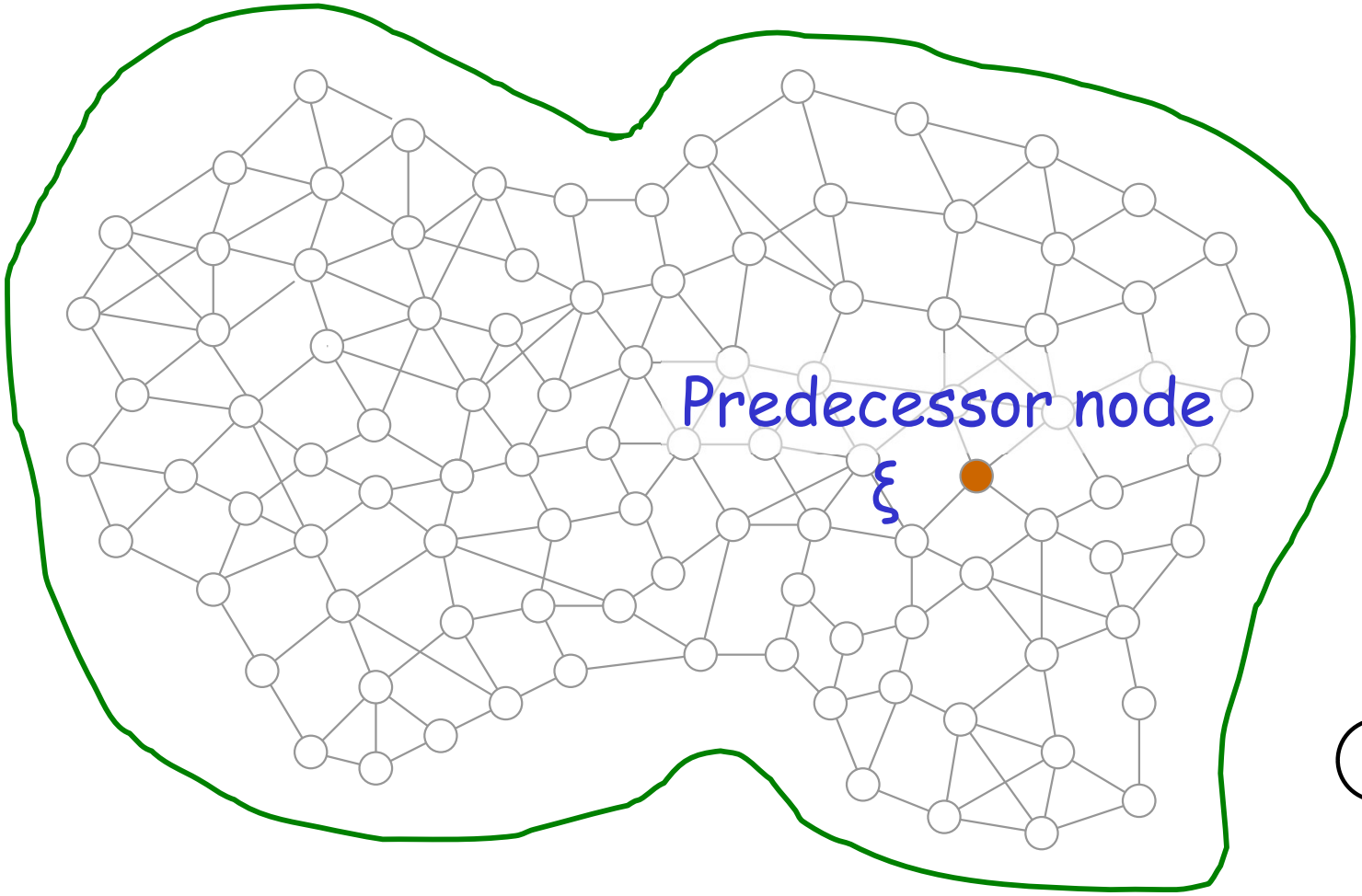


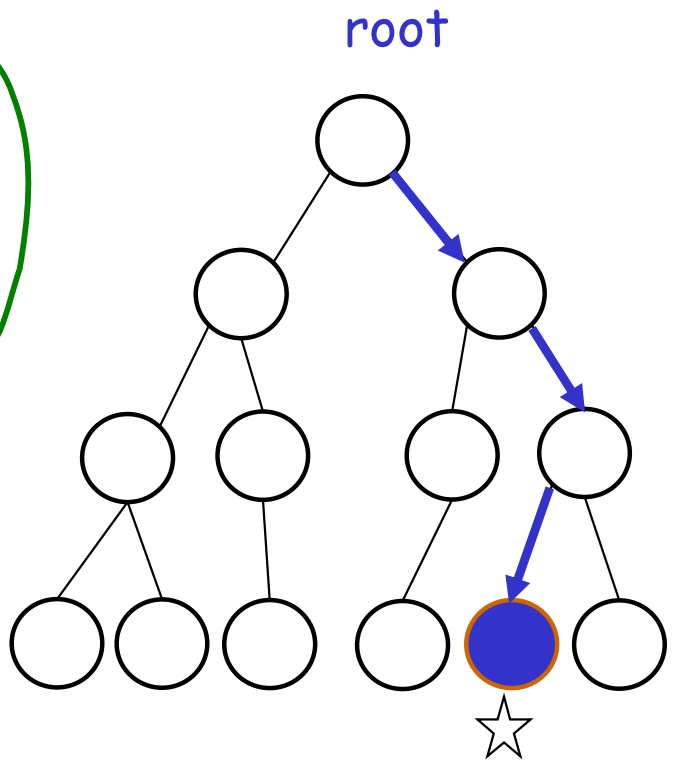
Set downward pointer while going up

Root node found, stop up phase



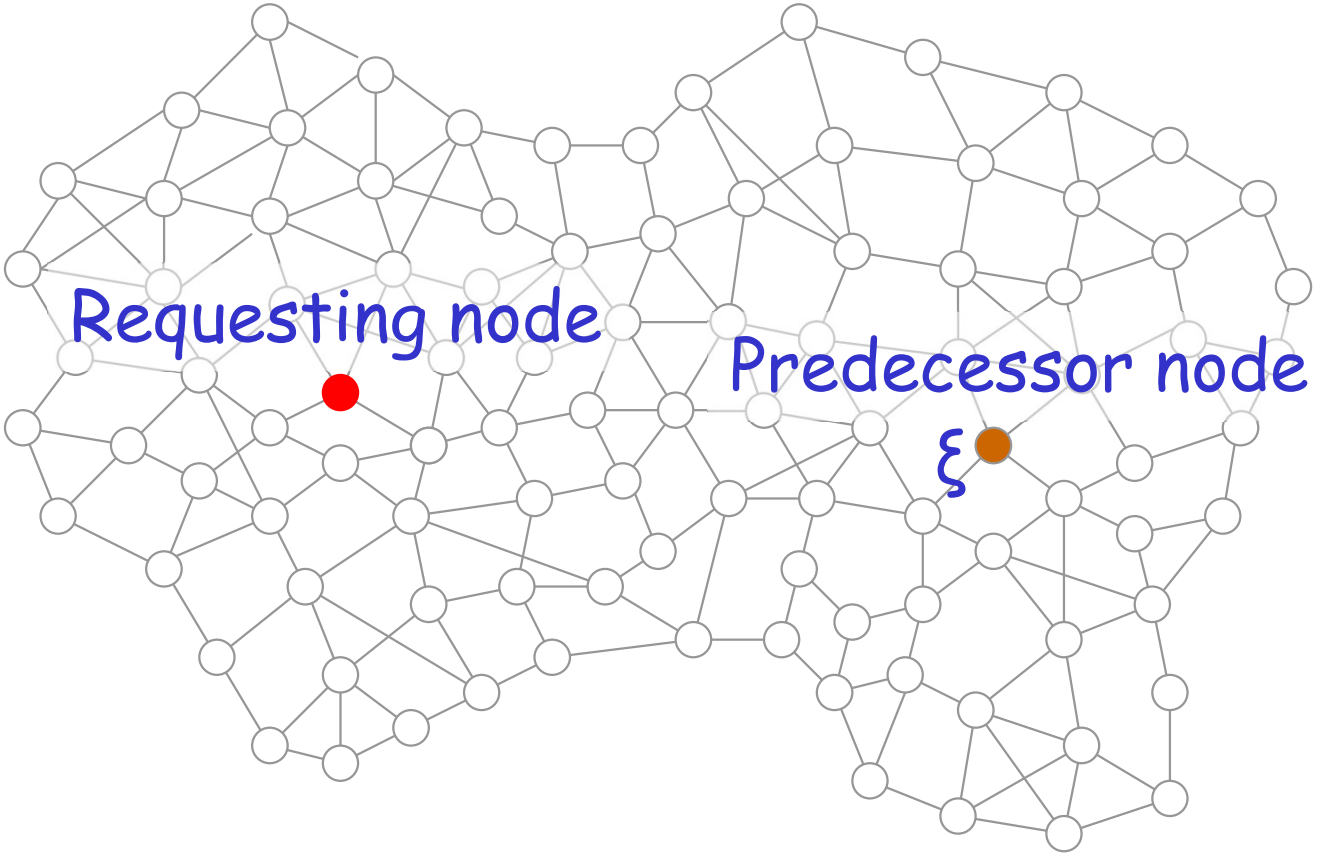


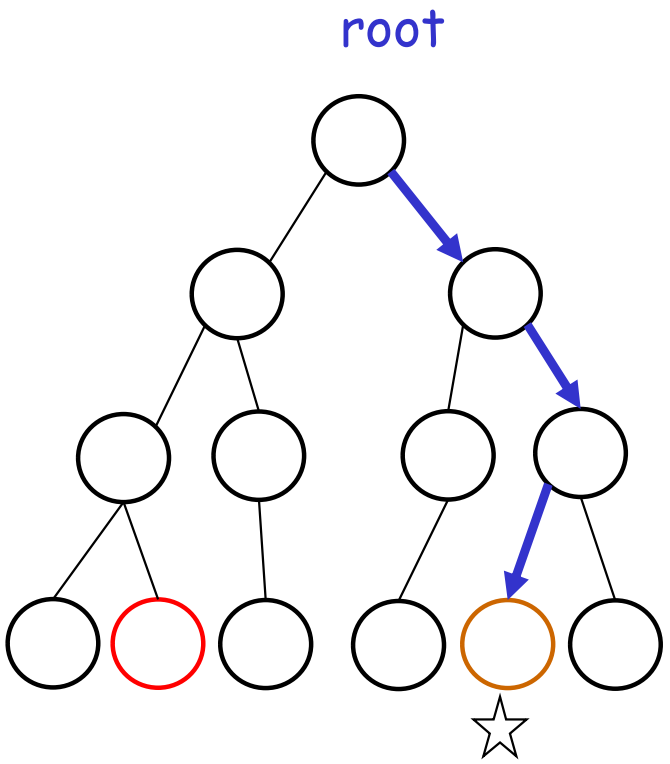




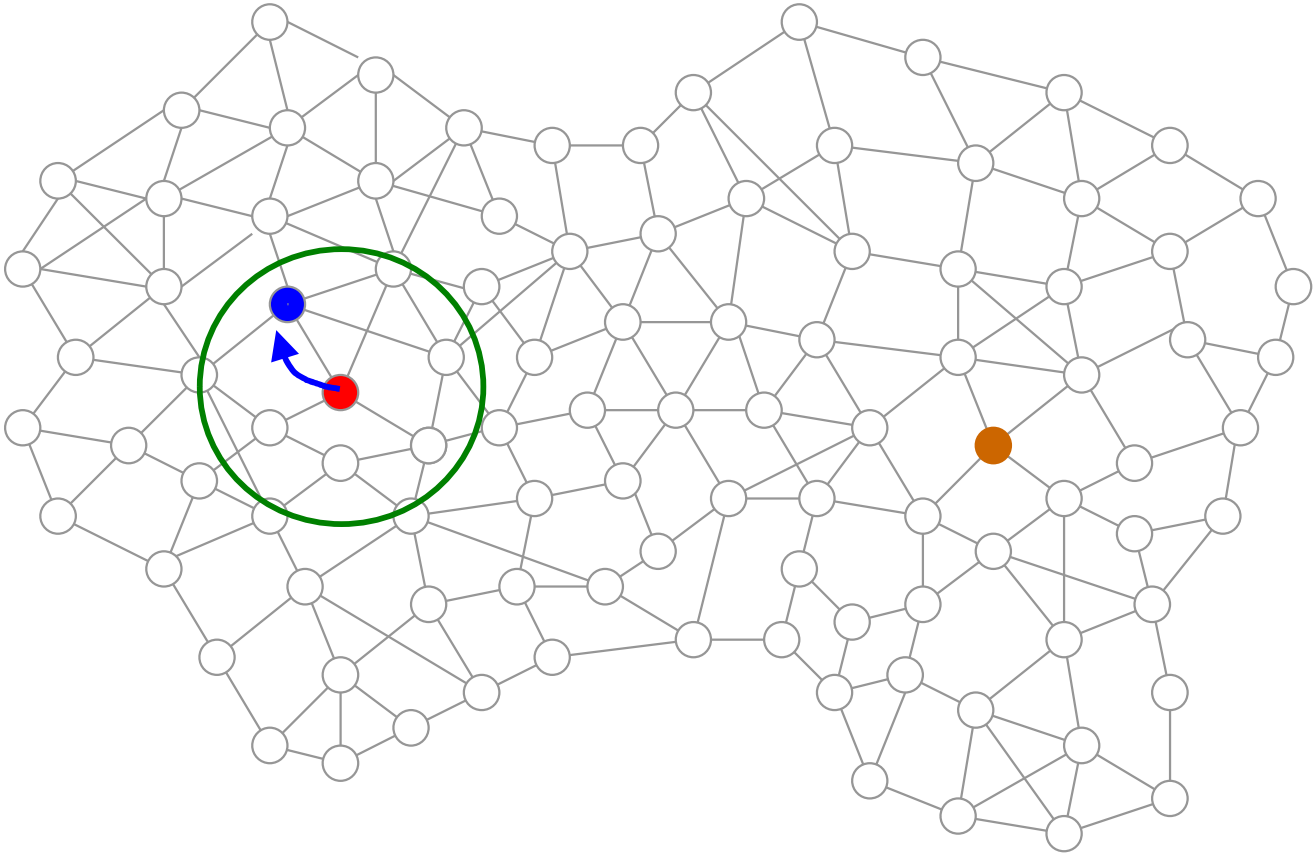
A successful **Publish** operation

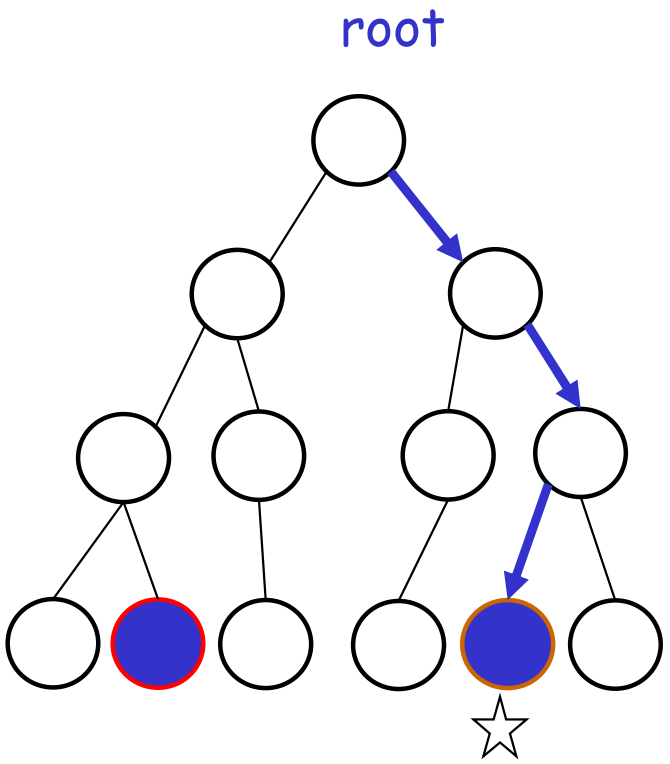
Move object (for Write)



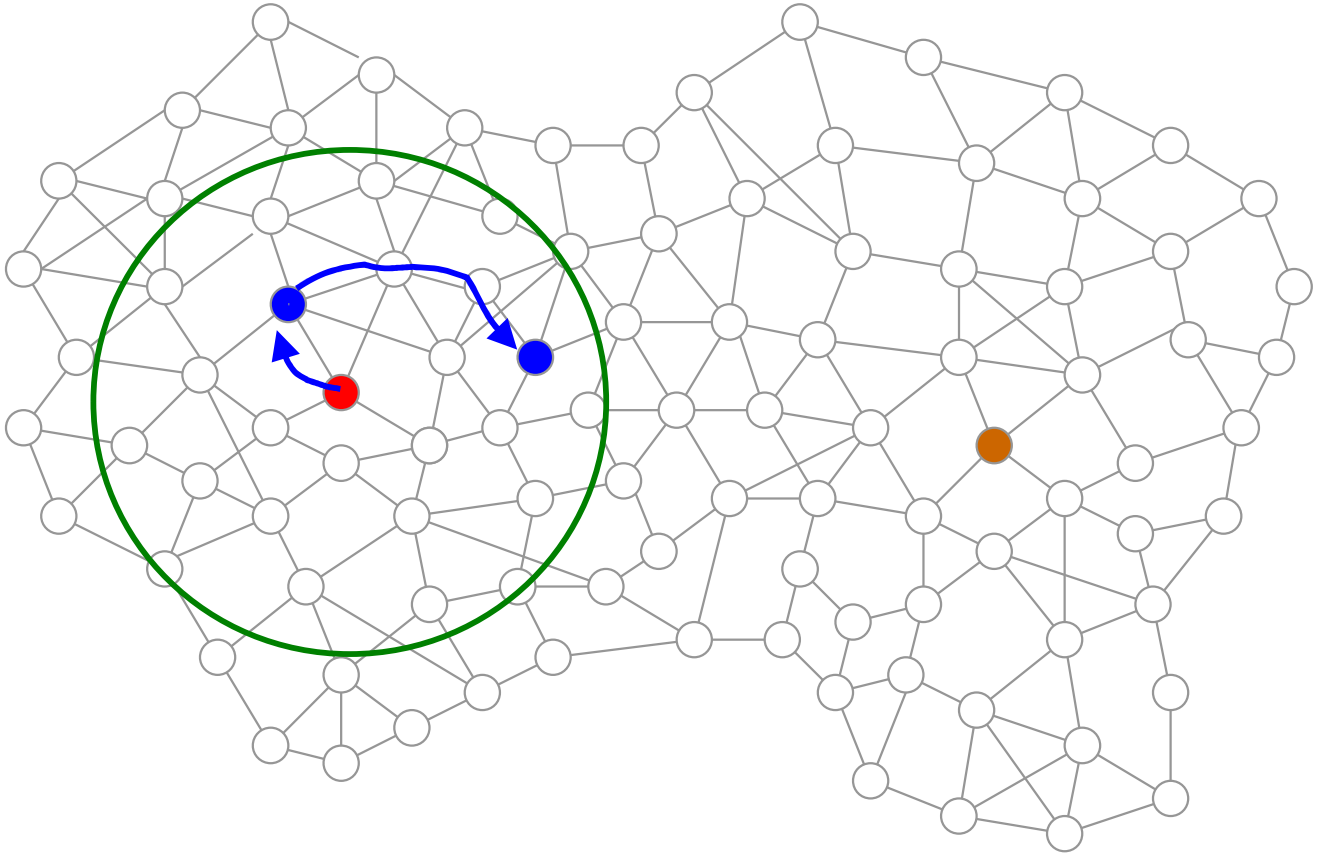


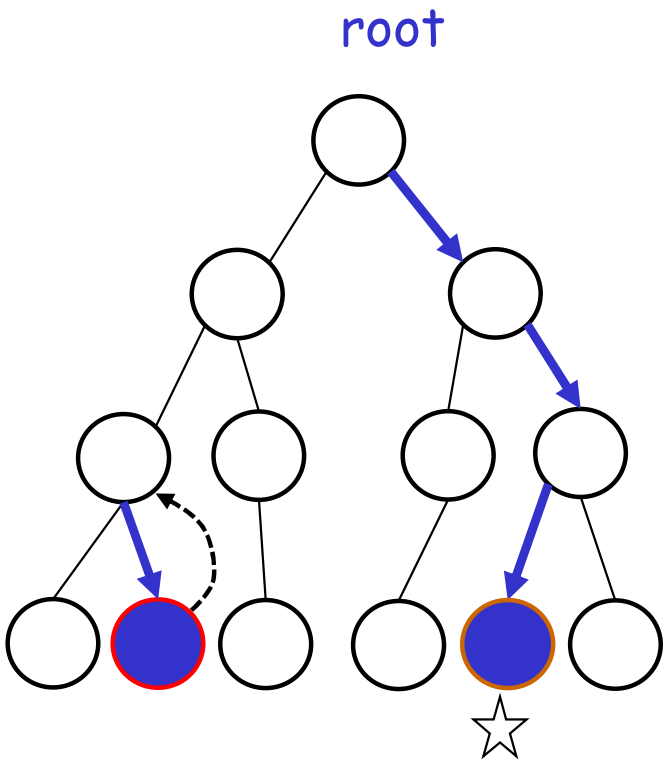
Send request to leader node of the cluster upward





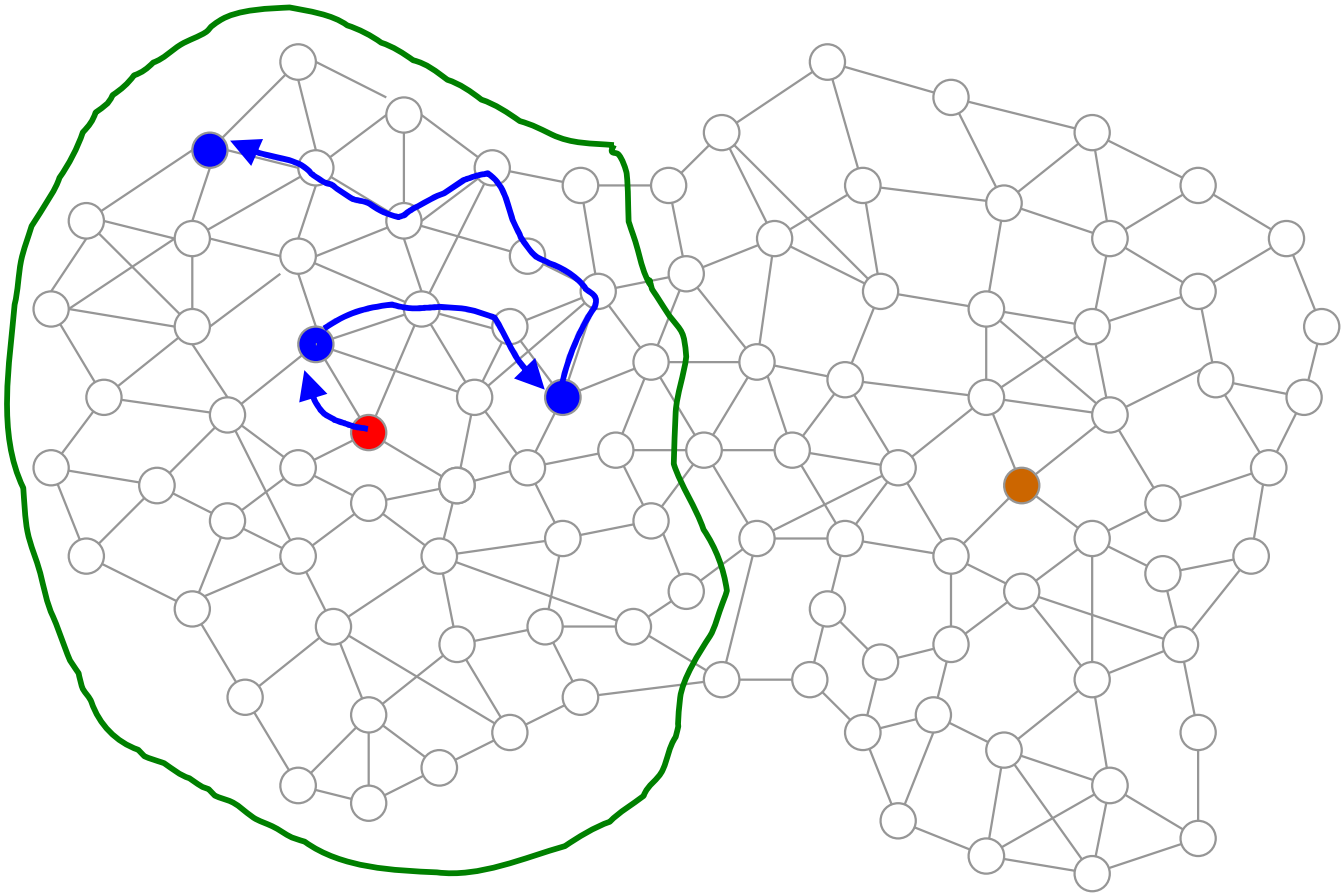
Continue up phase until downward pointer found

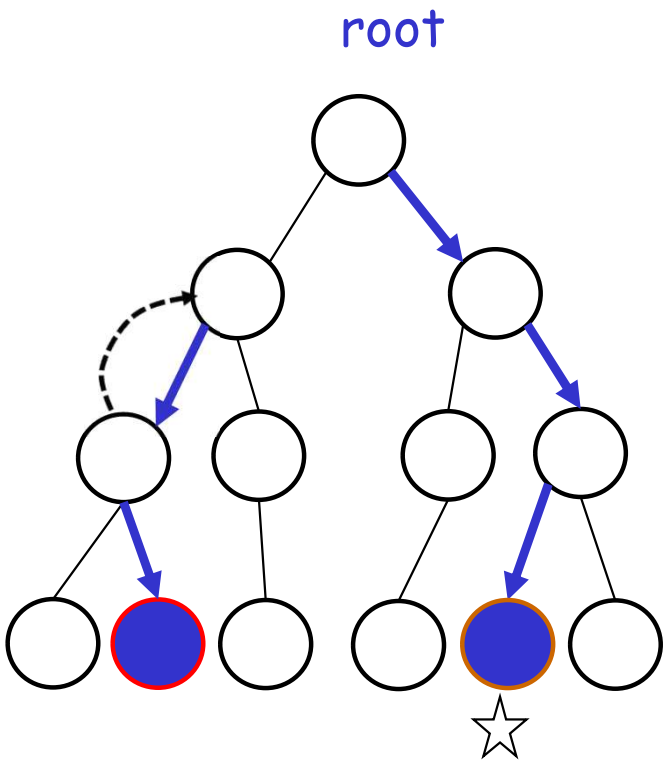




Set downward pointer while going up

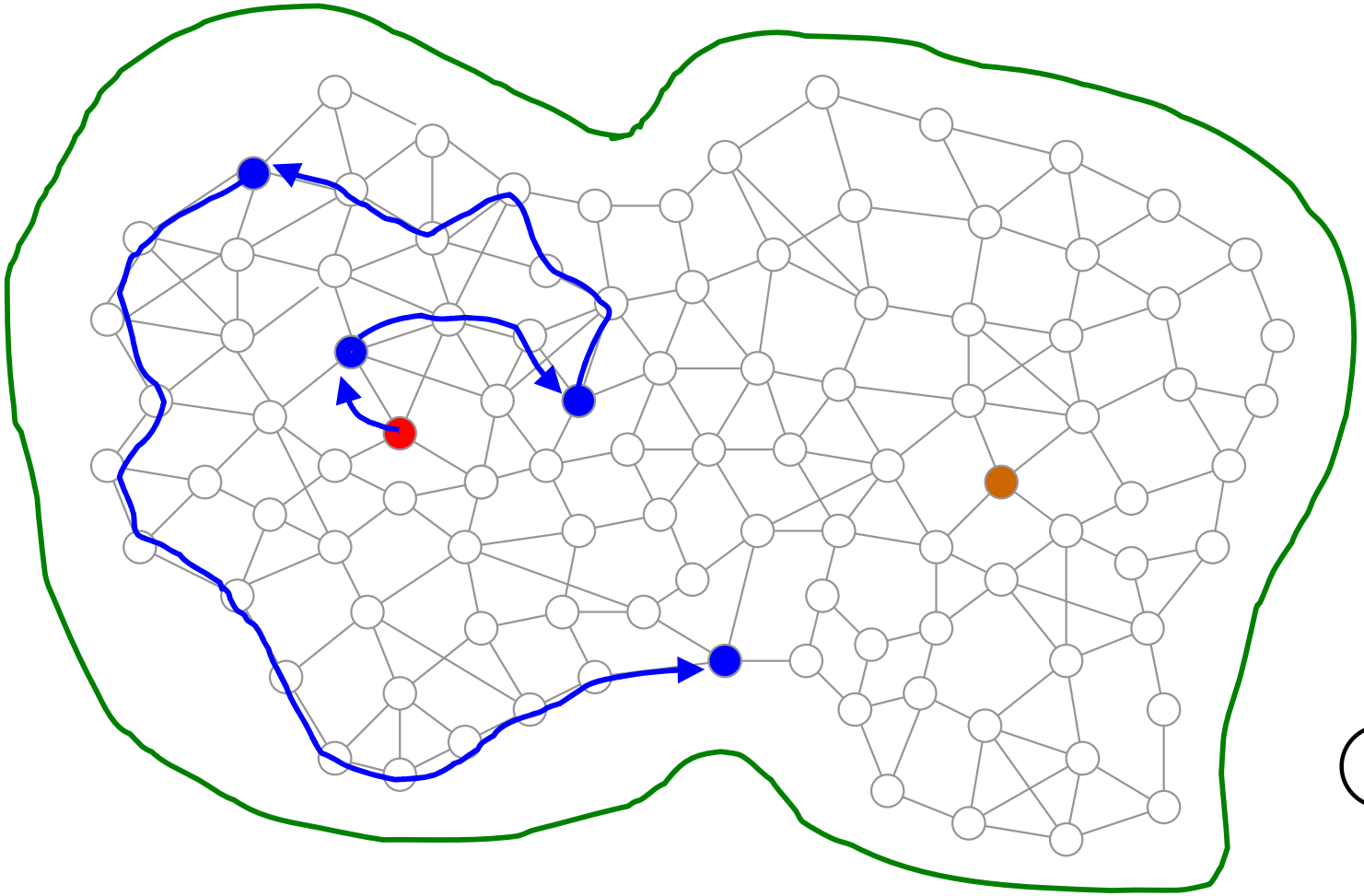
Continue up phase

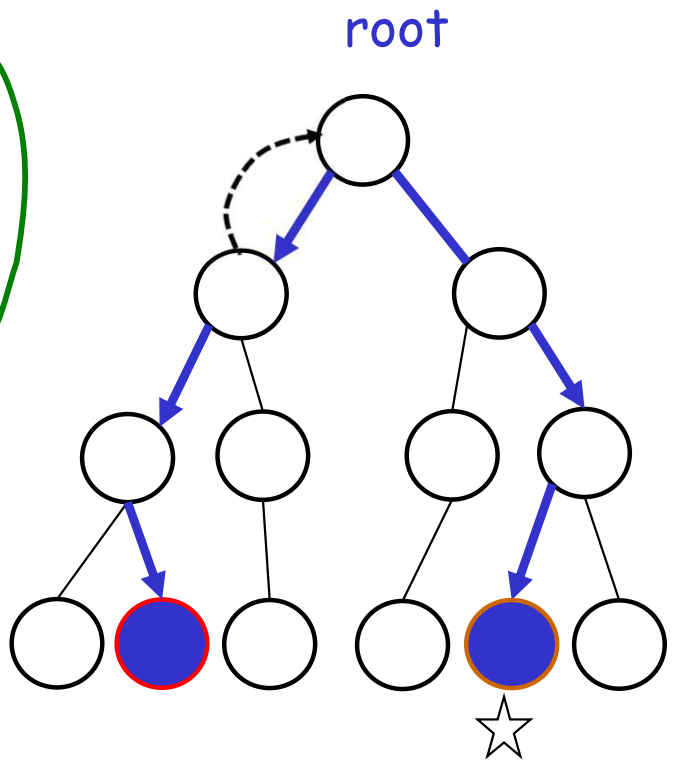




Set downward pointer while going up

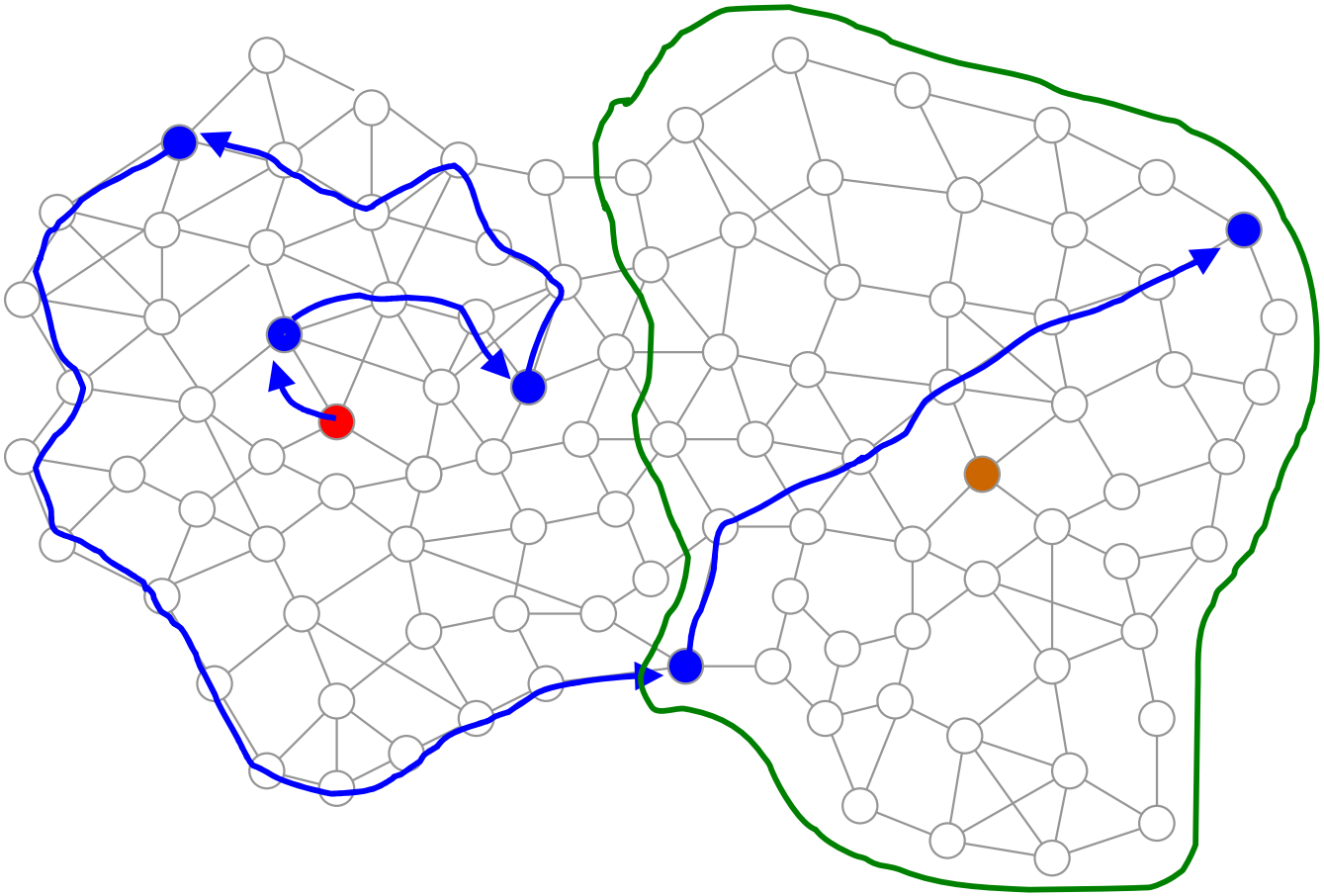
Continue up phase

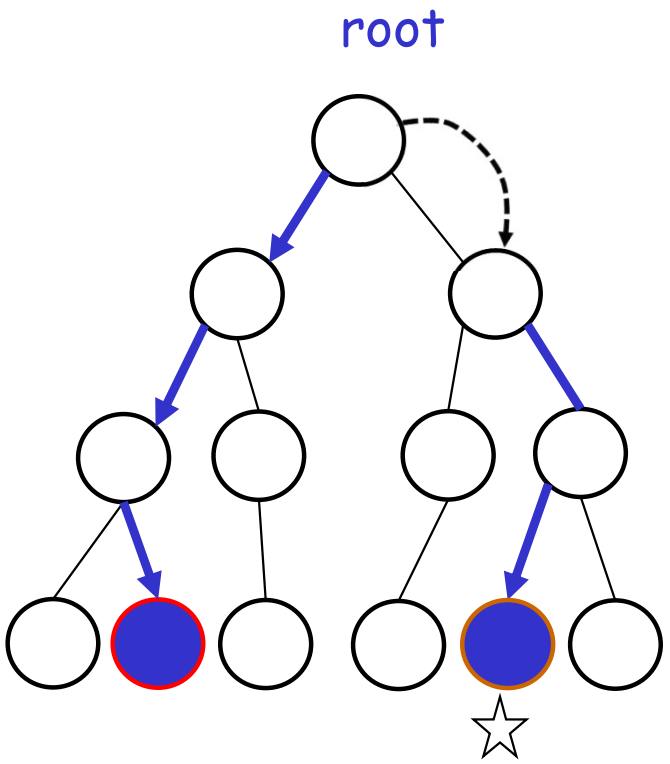




Set downward pointer while going up

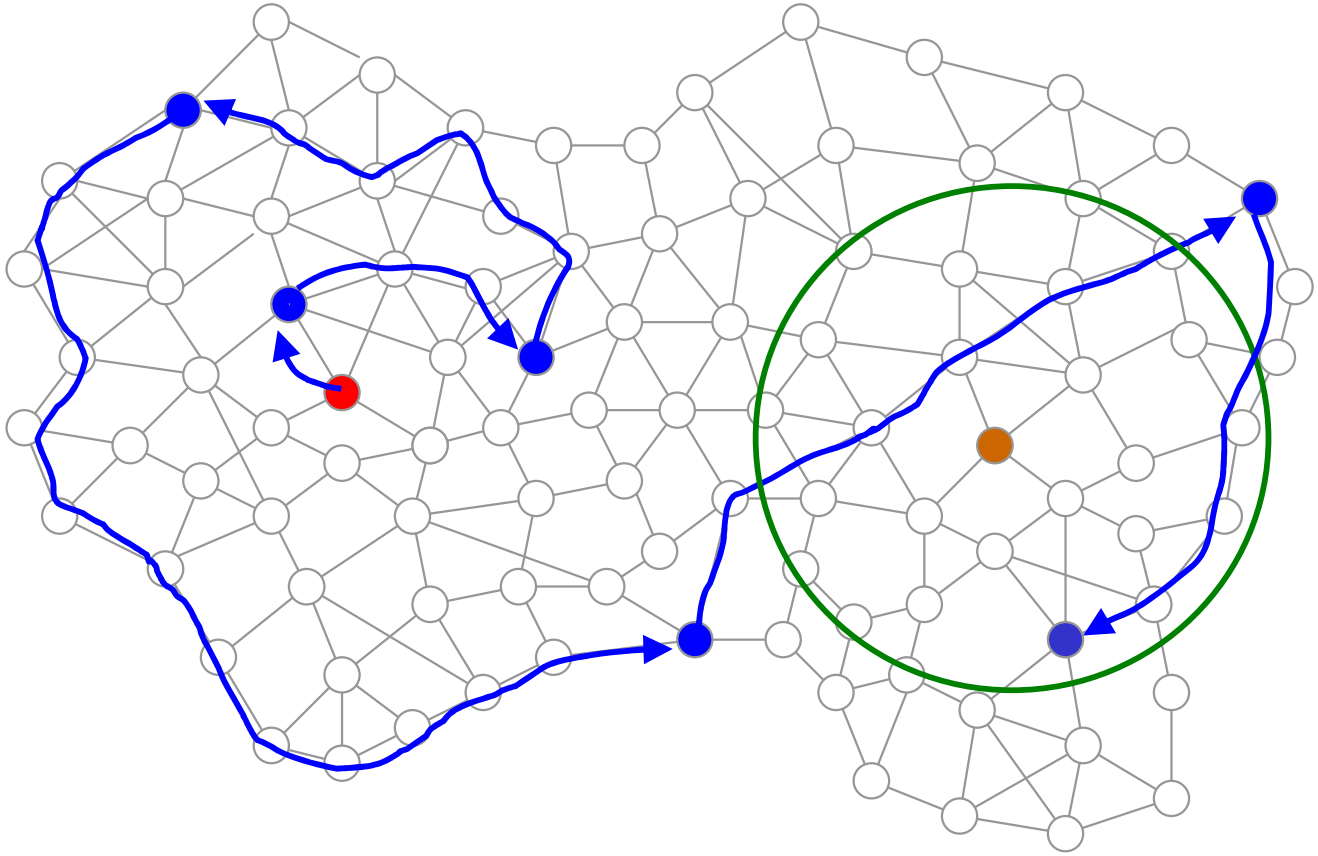
Downward pointer found, start down phase

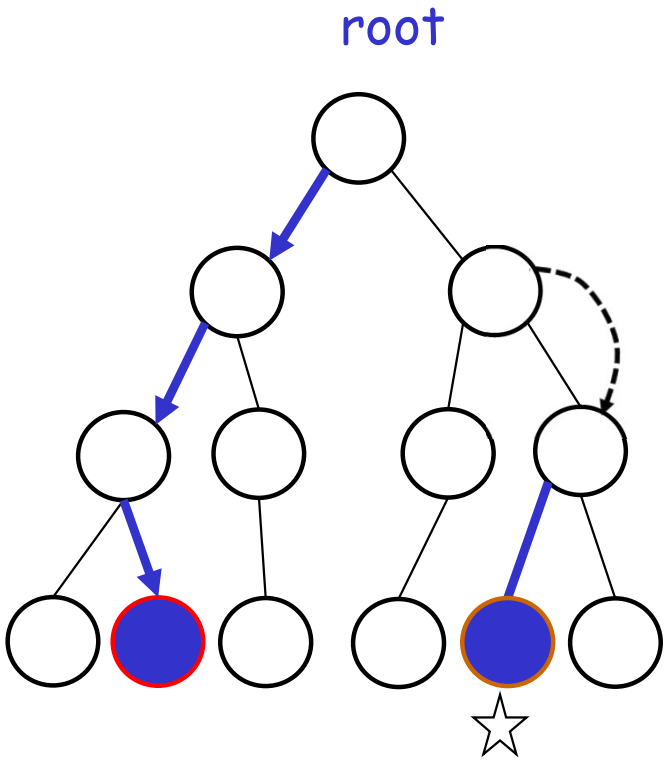




Discards downward pointer while going down

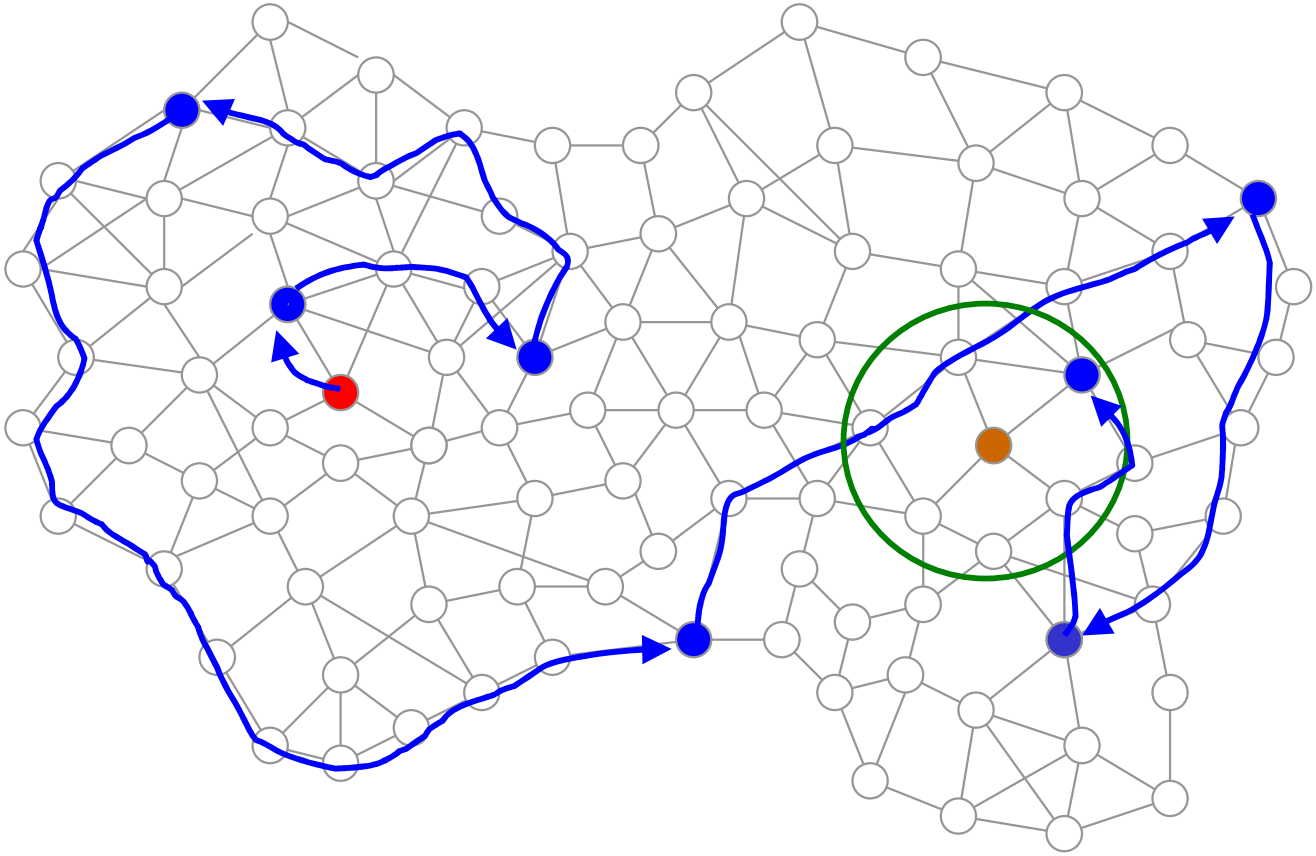
Continue down phase

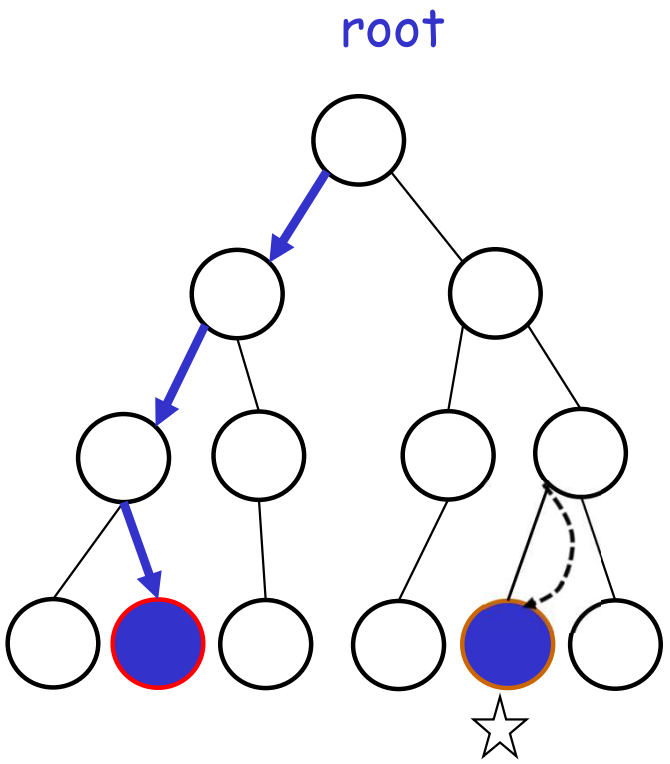




Discards downward pointer while going down

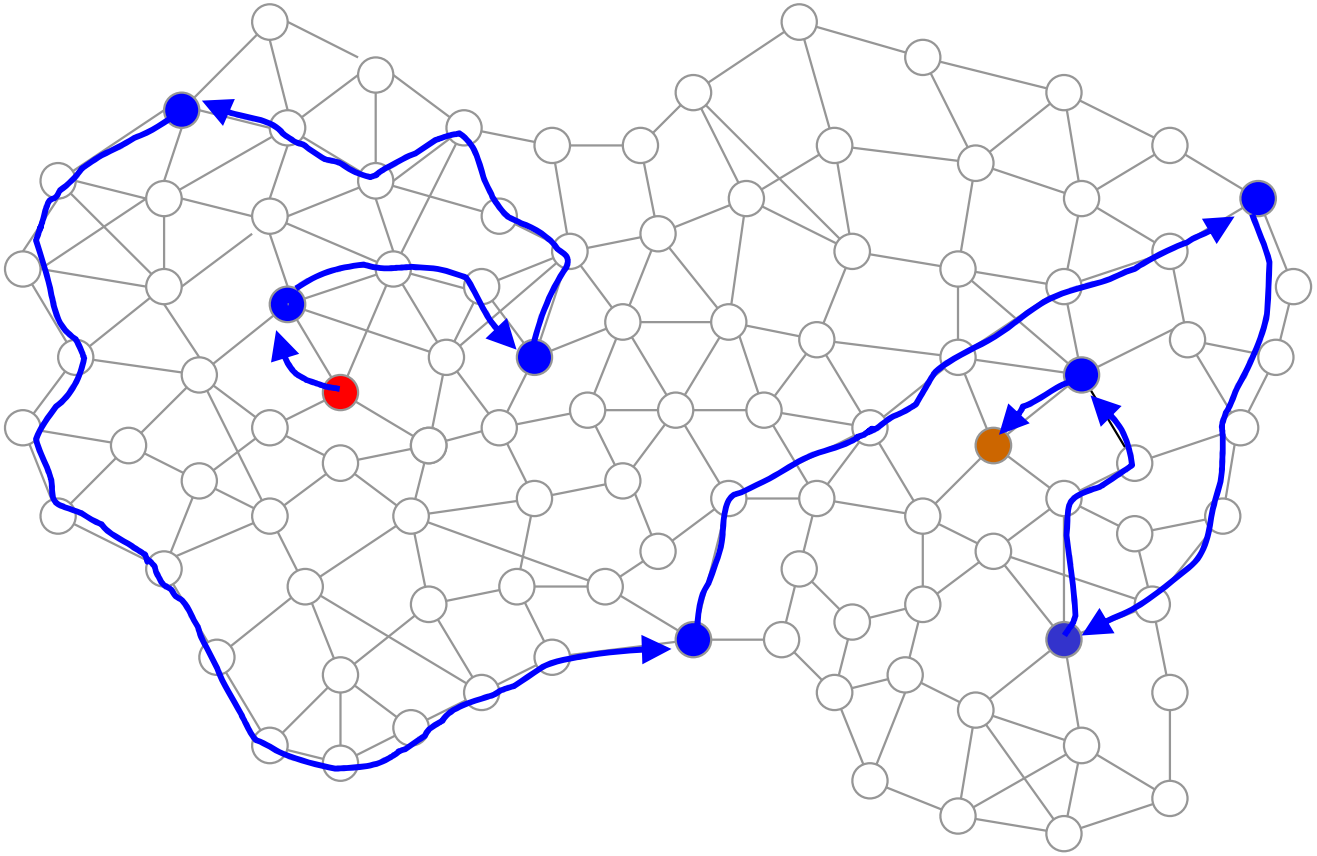
Continue down phase

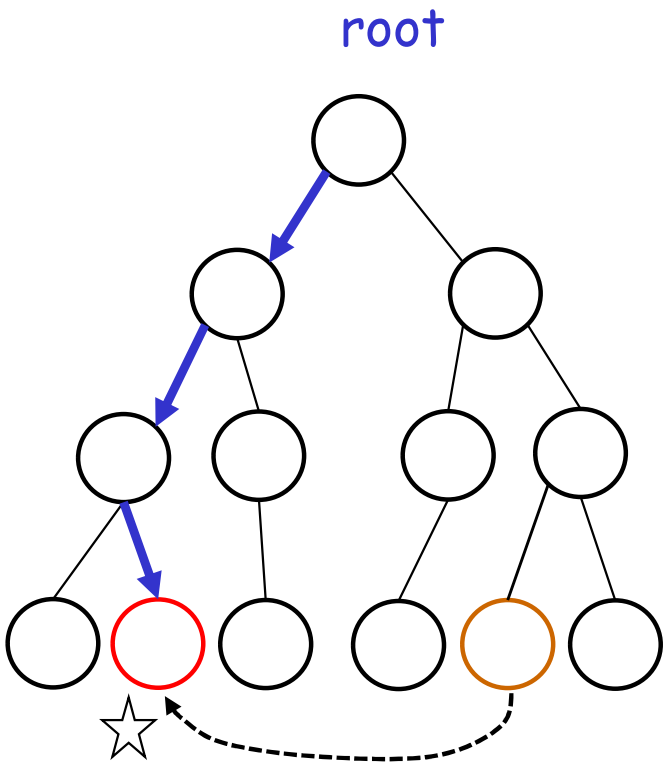




Discards downward pointer while going down

Predecessor reached, object is found and can be moved





Lookup object (for Read) is similar

Landscape of Existing Research

- **Online Model** – Move/Lookup (write/read) requests come over time
  - Requesting nodes not known a priori
  - Request arrival times not known a priori

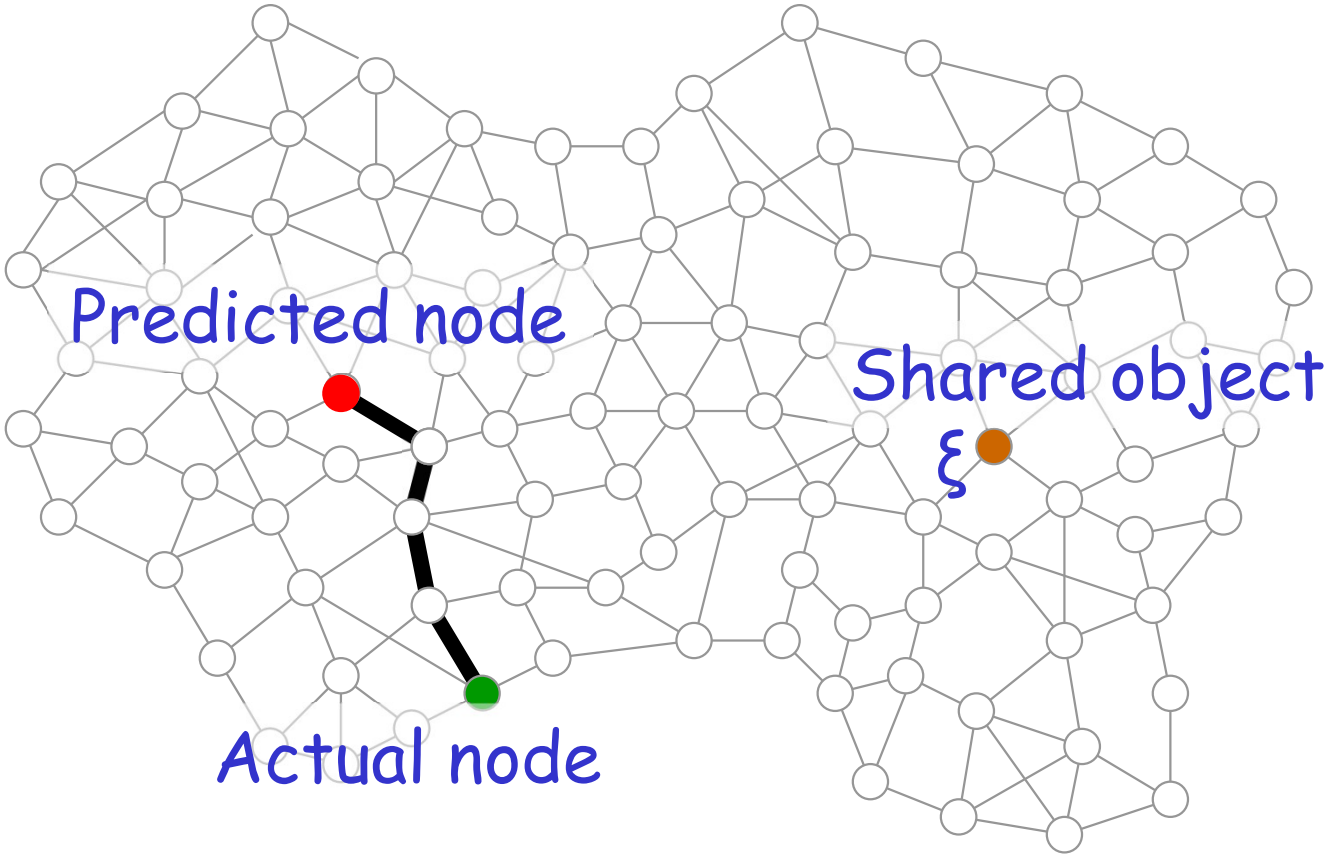
New Line of Research

- **Online Model** – Move/Lookup (write/read) requests come over time
  - Requesting nodes not known a priori
  - Request arrival times not known a priori

- **Prediction Model** – Can the protocols be optimized (w.r.t. stretch) if
  - Some limited knowledge on input is provided as predictions

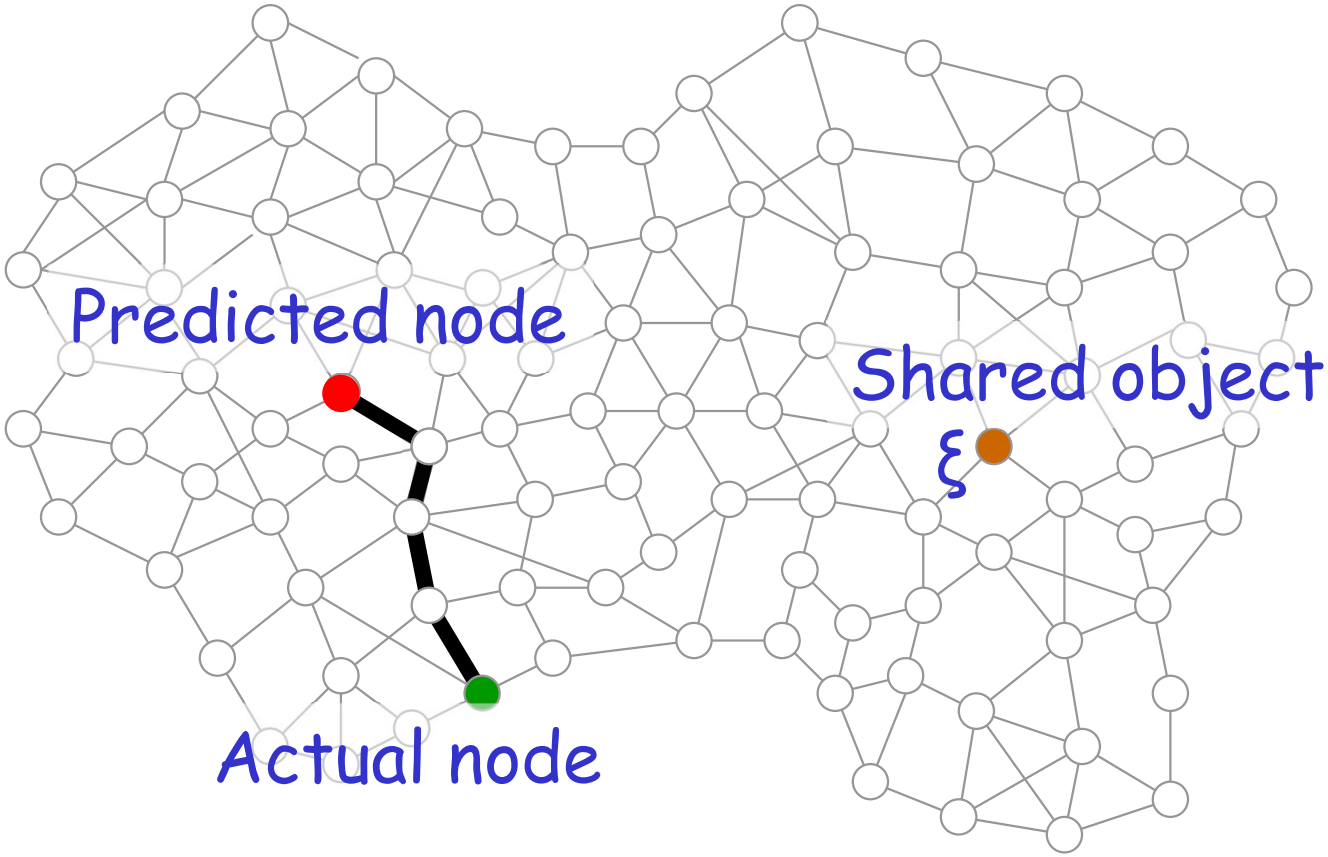
Location Prediction Model

- Requesting nodes are ***predicted*** a priori
- Request arrival times not known a priori
- **Predictions** can be erroneous (not known both error and its amount)



Location Prediction Model

- **Prediction error:** shortest distance between predicted node and actual node
- **Maximum Prediction Error:** D, the diameter



Protocol Design in the Prediction Model

- **Consistency**
  - Stretch when no error (predictions are perfect)
- **Smoothness**
  - Stretch when predictions have error
- **Robustness**
  - Stretch when predictions have maximum error

Protocol Design in the Prediction Model

- **Consistency**
  - Stretch when no error (predictions are perfect)
  - **Goal: much better than online (no prediction) model**
- **Smoothness**
  - Stretch when predictions have error
  - **Goal: gradually degrading with increasing error**
- **Robustness**
  - Stretch when predictions have maximum error (D in our case)
  - **Goal: no worse than online (no prediction) model**

New Results

Two Protocols in the Location Prediction Model

- **Tree-based**
  - $S_{steiner}$ -consistent, no error
  - ($S_{steiner}$+Error)-smooth for error "Error"
  - S_{ST} -robust, maximum error

- **Cluster-based**
  - $O(\log^2 n \log D')$-consistent, D' diameter of predicted nodes + initial owner node
  - $O(\log^2 n \log(D' + \text{Error}))$, D' diameter of predicted nodes + initial owner node
  - $O(\log^2 n \log D)$-robust, maximum error

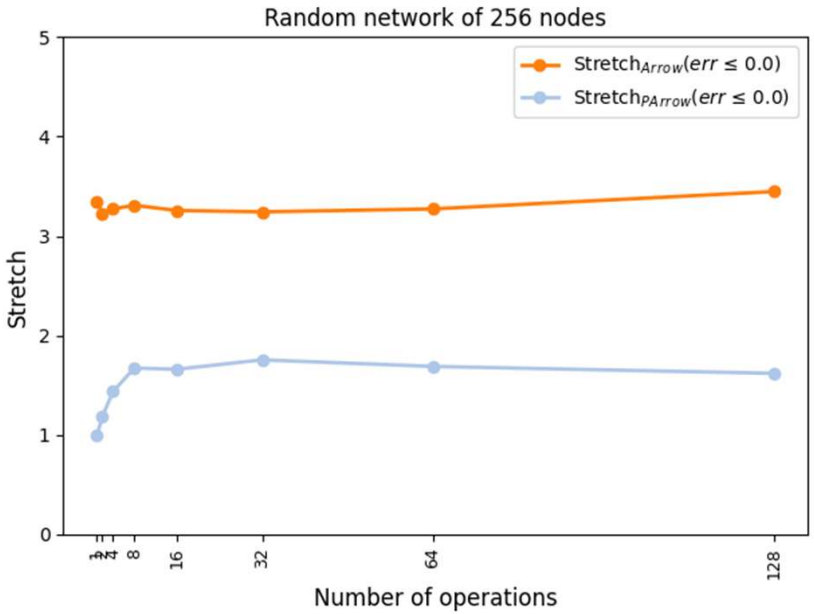
Ideas

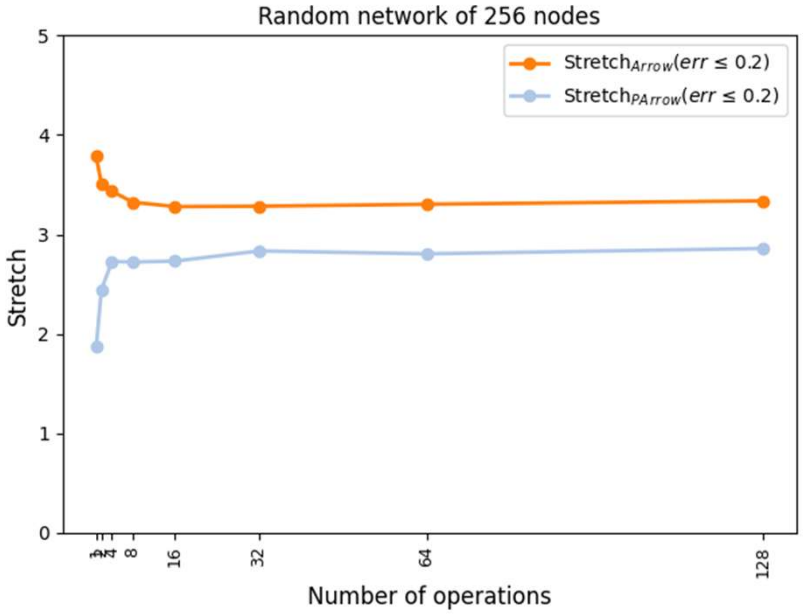
- **Tree-based (called Parrow)**
  - Construct Steiner Tree of predicted nodes
    - Use existing ideas, many protocols
  - Augment Steiner Tree with nodes not already in the tree
    - Consider Steiner Tree as a super node
    - Construct a MST of super node and nodes not in Steiner Tree
  - Prove the claimed bounds

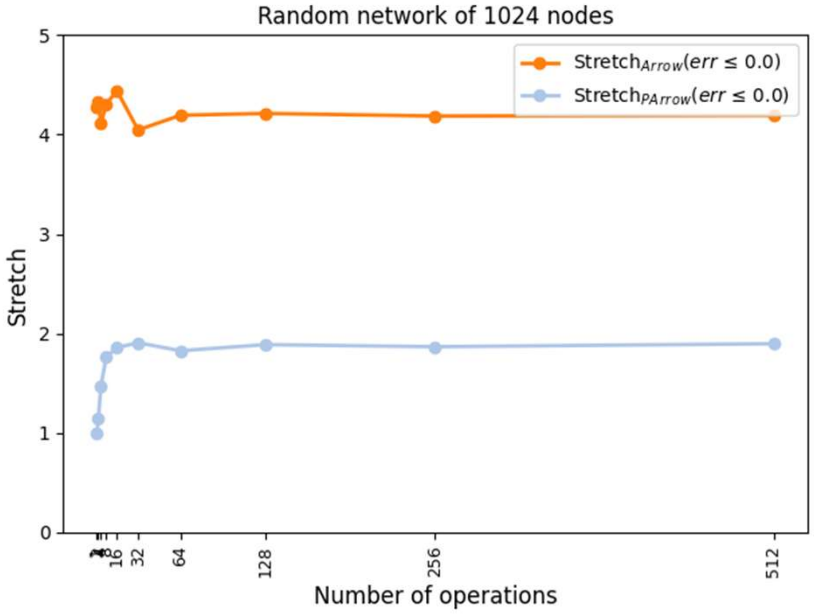
Ideas

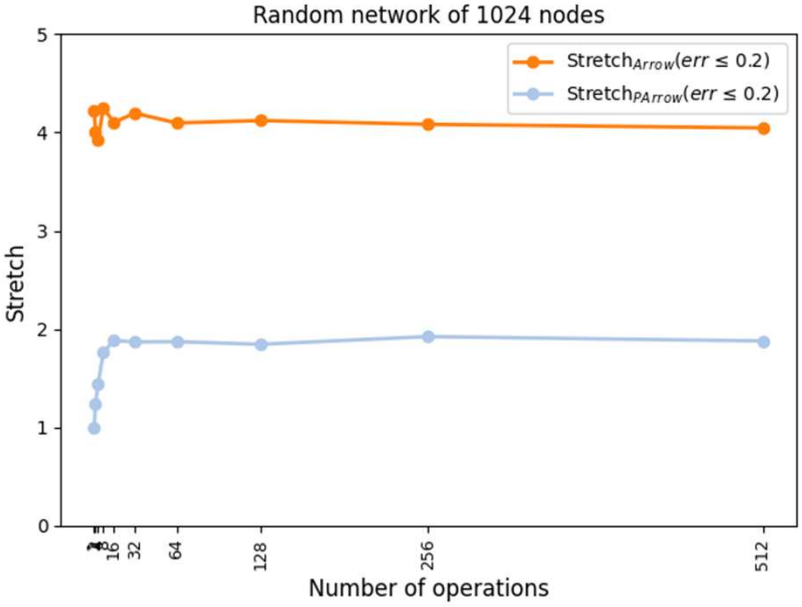
- **Cluster-based (called PSpiral)**
  - Use the cluster hierarchy of existing protocols
  - Show that no major change is required,
    - except the object needs to be published up to the root cluster to account for predictor error
  - Prove the claimed bounds

Experimental Results (Tree-Based)

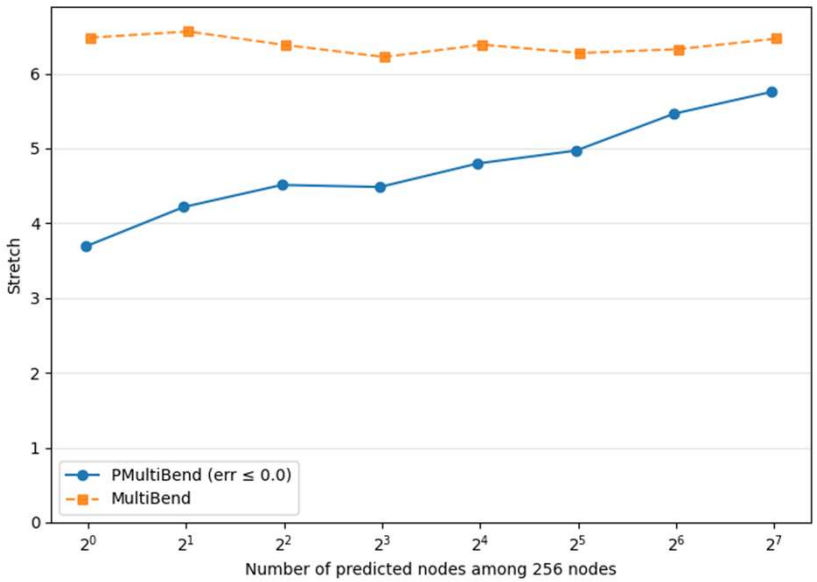


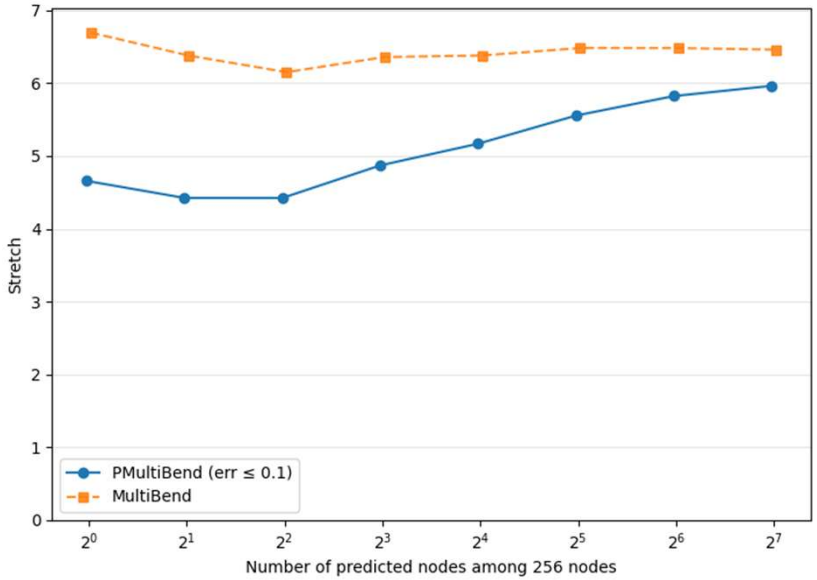


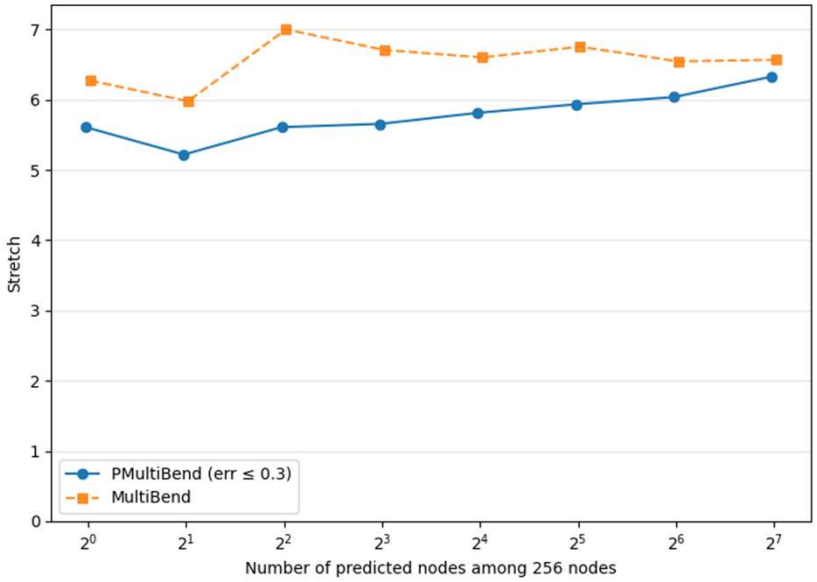


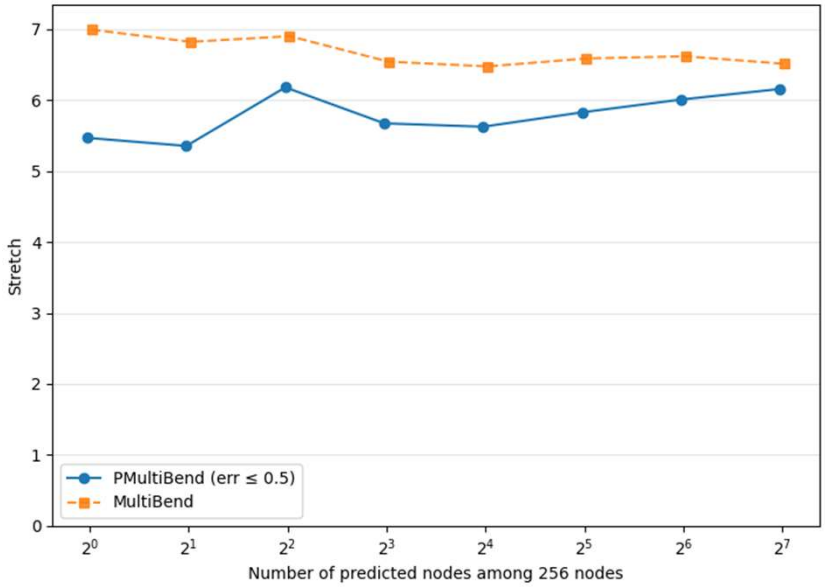


Experimental Results (Cluster-Based)









Summary

Two protocols for accessing a shared object in the network in the location prediction model:

- Tree-based
- Cluster-based

Future Directions

- Online Location Prediction Model – predicted locations released over time
- Other prediction models (Time?)
- Detailed Experiments

Thank you!!!