Network Metric Approximation and Mobile Agent Coordination in Sensor Networks

Jie Gao

Stony Brook University

09/09/09

Finding street parking @ Manhattan is stressful

- Crowded
- Most streets are one-way.
- If you see a spot freed up behind, it is likely gone by the time you turn around.

Use sensors to detect empty parking spots

Event: where can I park?

Resource: available parking spot!

Challenges:

1. Need “broker service”
2. Avoid conflict → matching
3. Prefer small travel distances

Event: where can I park?

Resource: available parking spot!

Distributed resource management

- Goal: match **k events** with **k resources**, detected by a network of **n** sensors
  - Minimize the **total distance** of the matching
  - Use **low communication cost** among sensors for coordination.
  - Online & offline setting.
    - Offline: all events are present.
    - Online: events coming one by one.

Naïve approaches do not work well

- Flooding events
  - Flow algorithm to find optimal min-cost matching.
  - Communication cost is high: **$O(nk)$**
- Simple greedy matching
  - "Local exploration" to find the closest available resource
  - Match closest pair, remove it, and continue
  - Cost of matching is too high: approx. ratio **$O(k^{1.58})$**

Naïve approaches do not work well

- Flooding events
  - Flow algorithm to find optimal min-cost matching.
  - Communication cost is high: **O(nk)**
- Simple greedy matching **on a tree**
  - "Local exploration" to find the closest available resource
  - Match closest pair, remove it, and continue
  - Cost of matching is ~~too high: approx. ratio **$O(k^{1.58})$**~~

optimal

N. Bansal, N. Buchbinder, A. Gupta, J. S. Naor. *ESA'07.*

Our solution

- Extract a **tree** metric on the sensor nodes with O(logn) distortion
- Apply the simple greedy matching on the **tree** metric
- Two important facts
  - Greedy matching on a tree metric is optimal.
  - The tree structure allows easy probing based detection of nearby resources

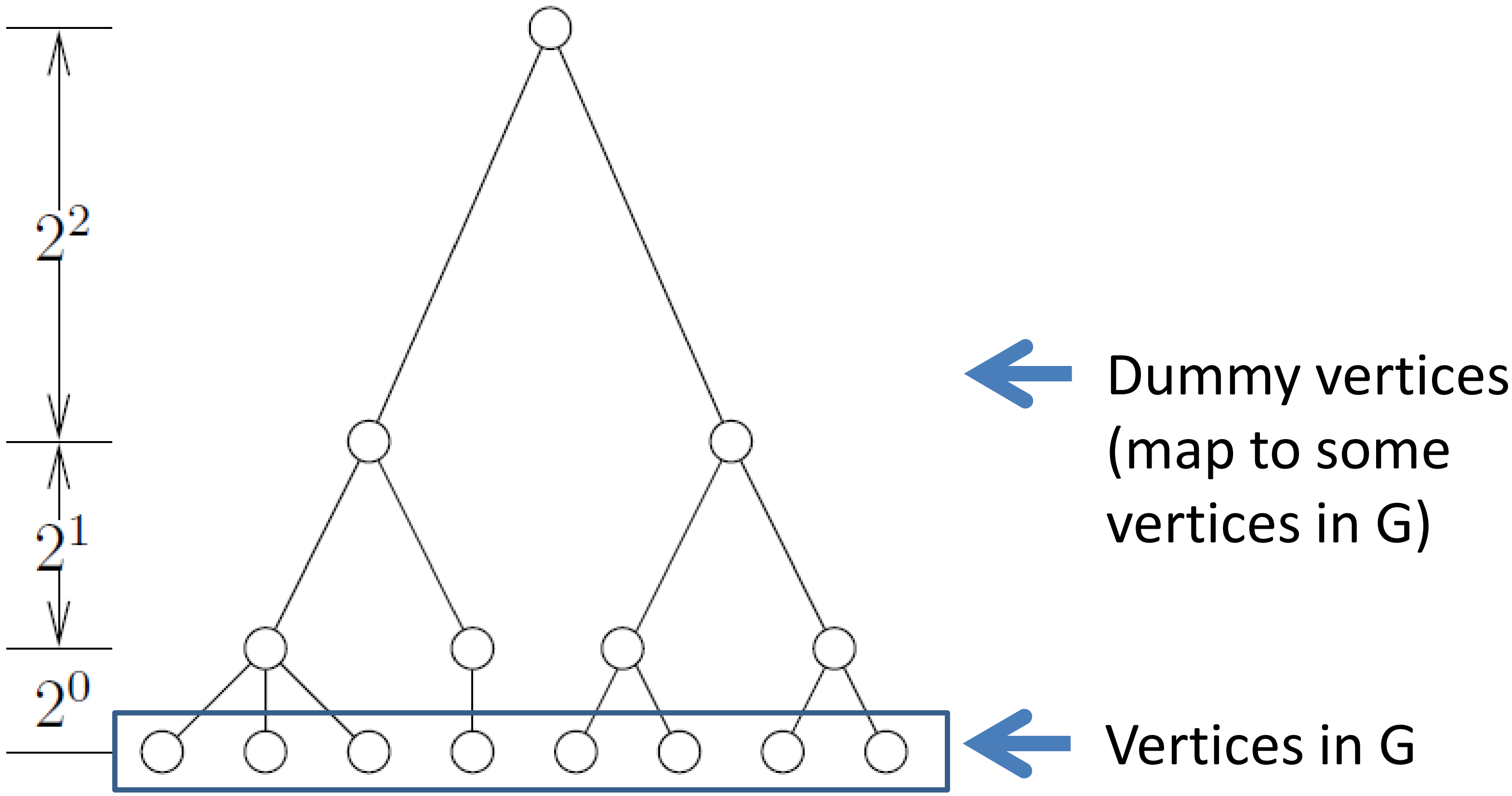
What is next?

- Extract a tree metric
- Implement greedy matching on the tree

Hierarchical well-separated tree (HST)

- α- HST: a weighted tree H extracted from G.
  – The weights from each node to its children are the same
  – All root-to-leaf paths have the same hop distance
  – The edge weights from the root to leaf decrease by a factor of α.

An example of 2-HST



Compute a HST

- [FRT03]: a top-down centralized algorithm:

  $$\mathbf{d_G(u, v) \le d_H(u, v) \le O(\log n)\, d_G(u, v)}$$

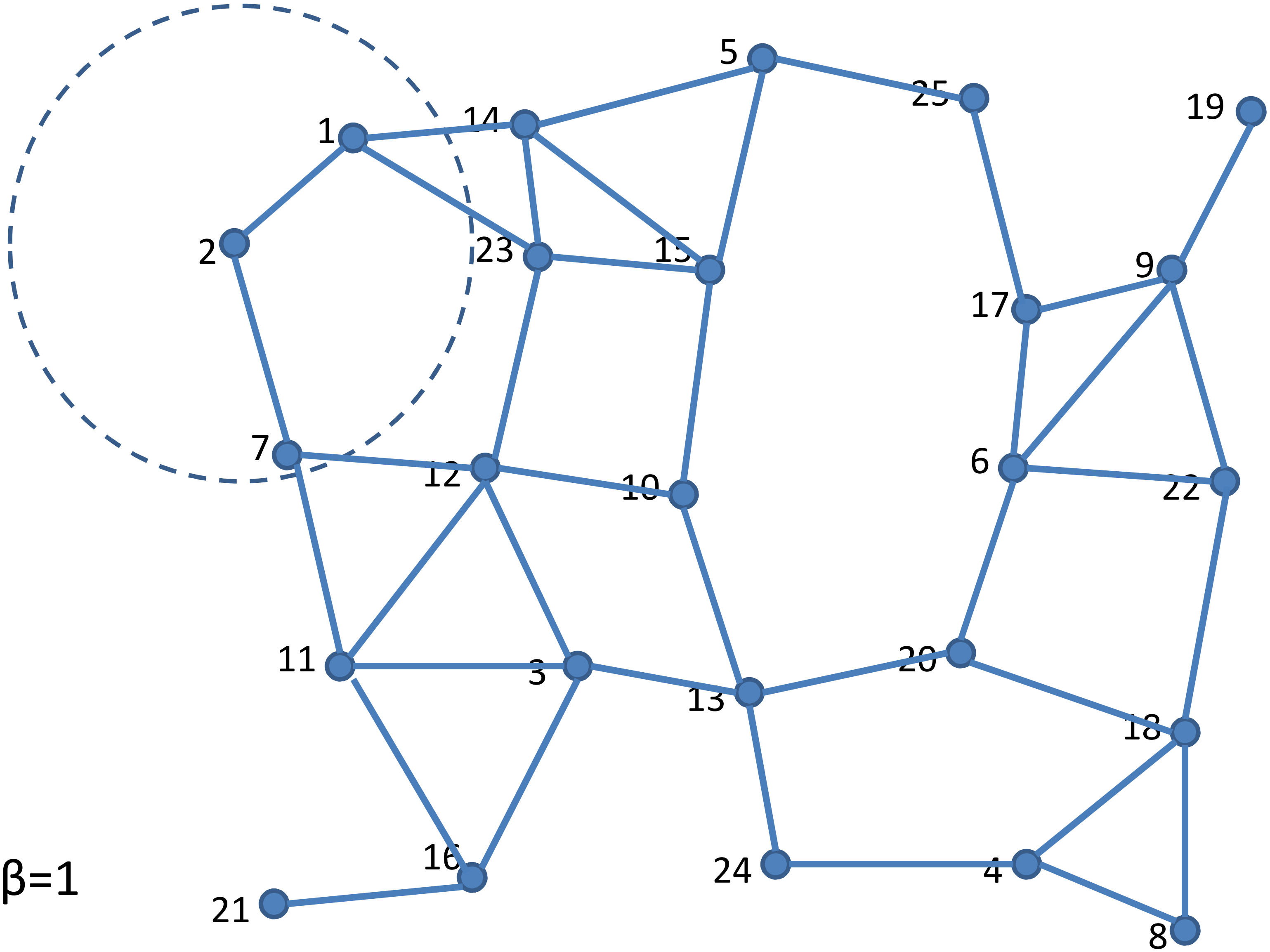
- We propose: a bottom-up distributed algorithm
  - Computes the same tree.
  - With a communication cost of **$O(n \log n)$**

J. Fakcharoenphol, S. Rao, K. Talwar, STOC 2003.

Distributed Algorithm for 2-HST

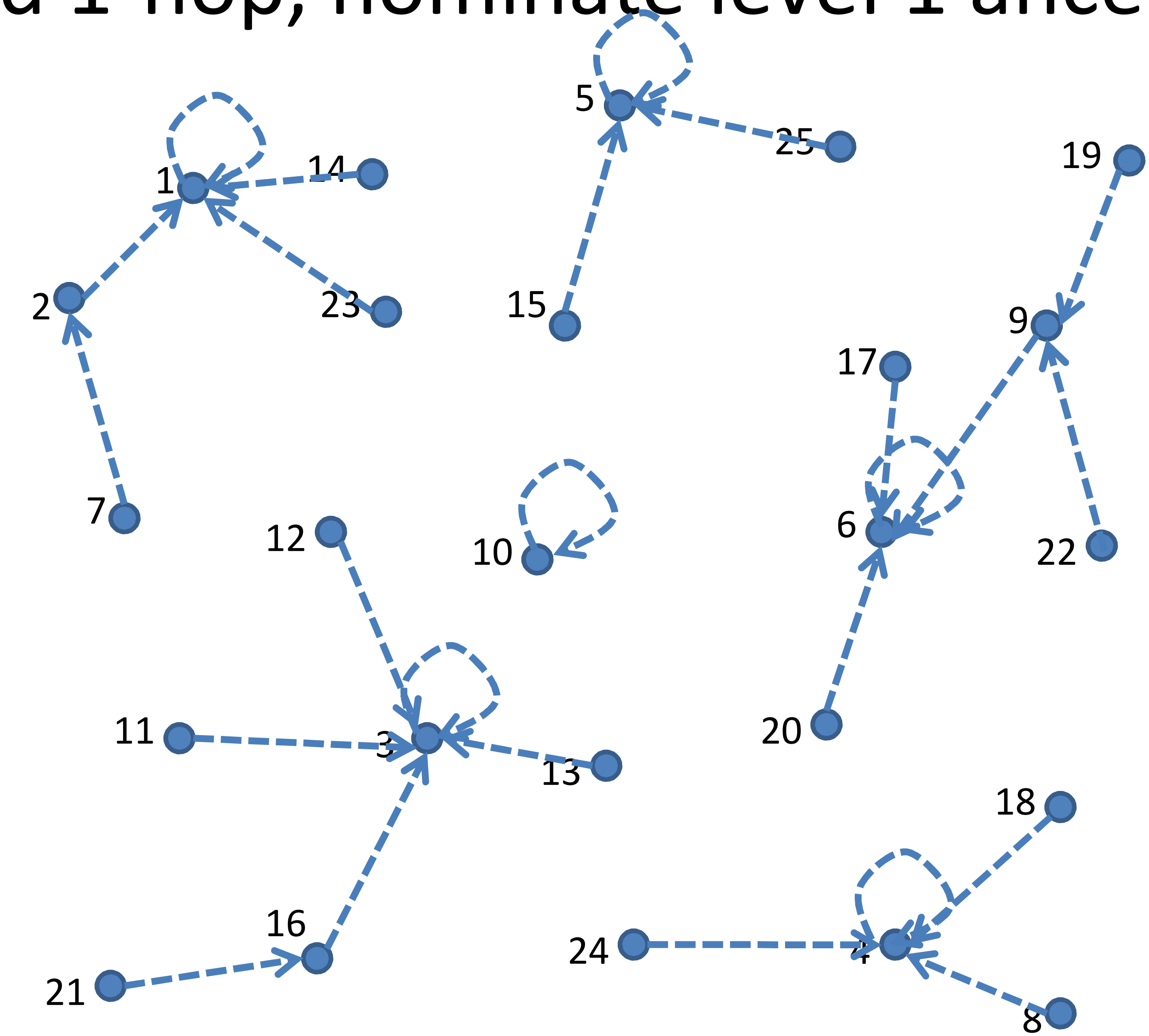
- Assume a random permutation on the nodes and a parameter β randomly chosen in [.5, 1].
- In each round i,
  - Each candidate nodes flood up to distance 2^i β
  - Each node ***u*** keep the lowest rank node ***v*** reached.
  - ***u*** nominate ***v*** as its i^{th}-level ancestor on H.
  - Only nominated nodes remain in the next round.
- Until only one candidate is left.

An example of 2-HST

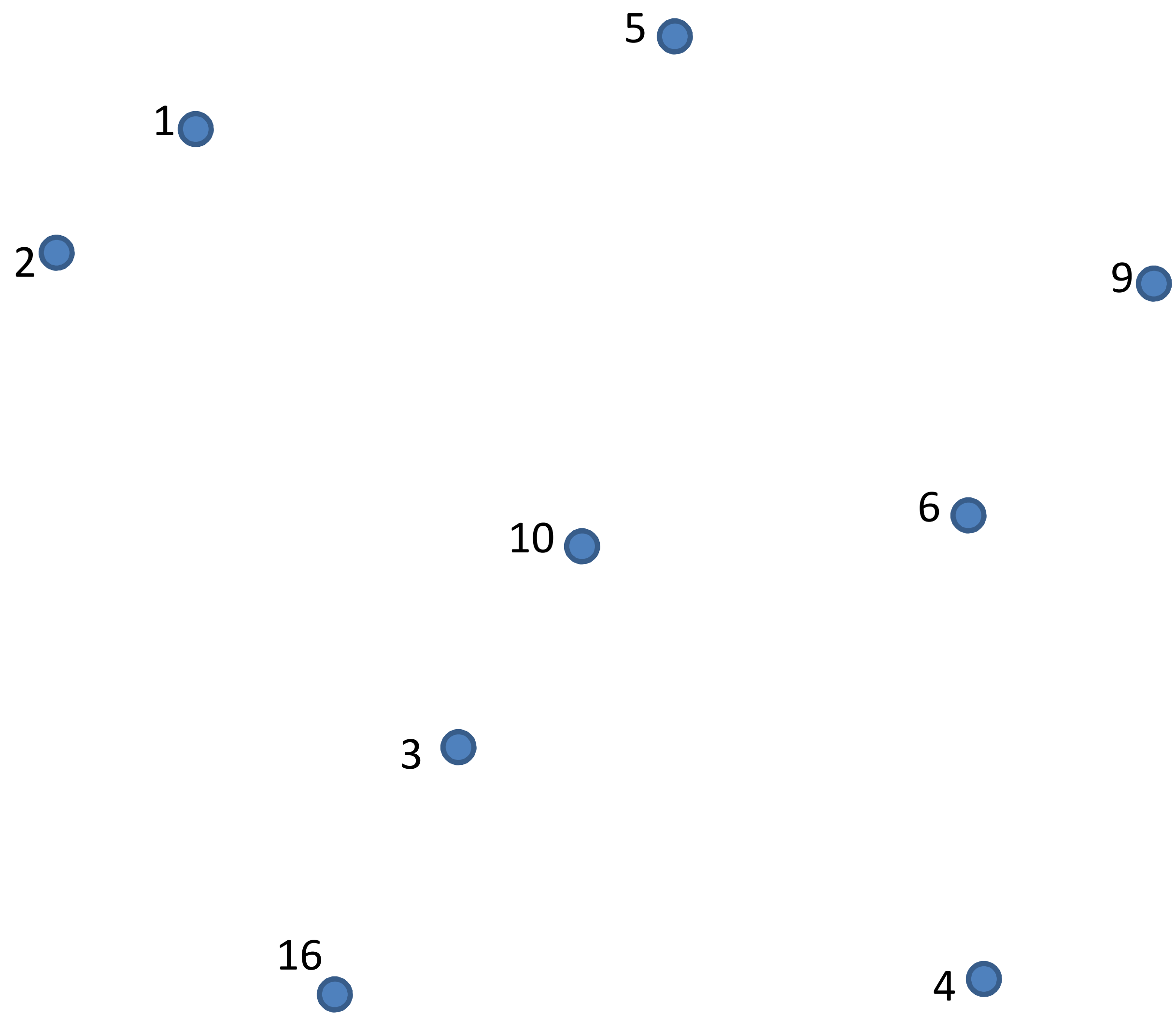


β=1

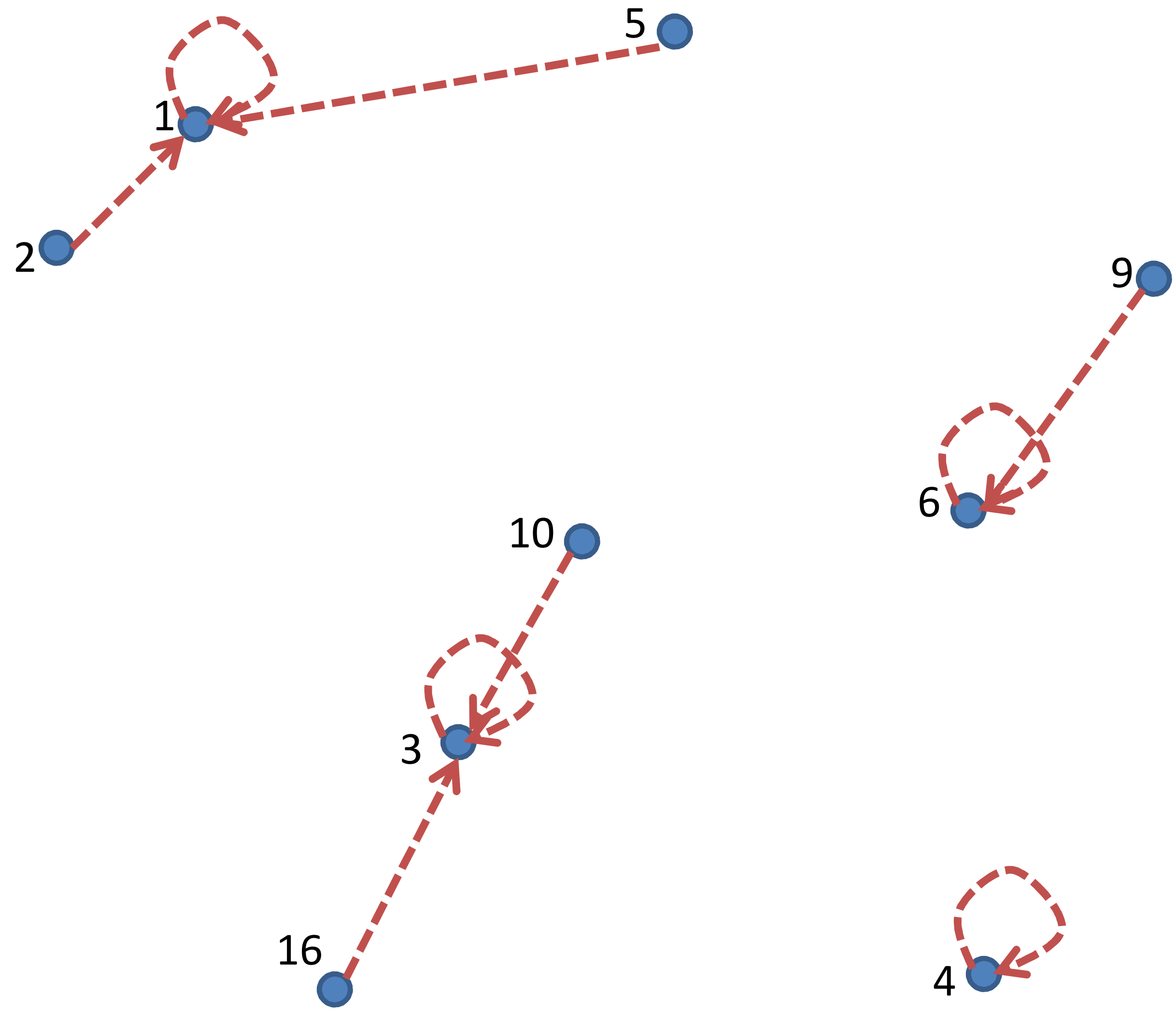
Flood 1-hop, nominate level 1 ancestor



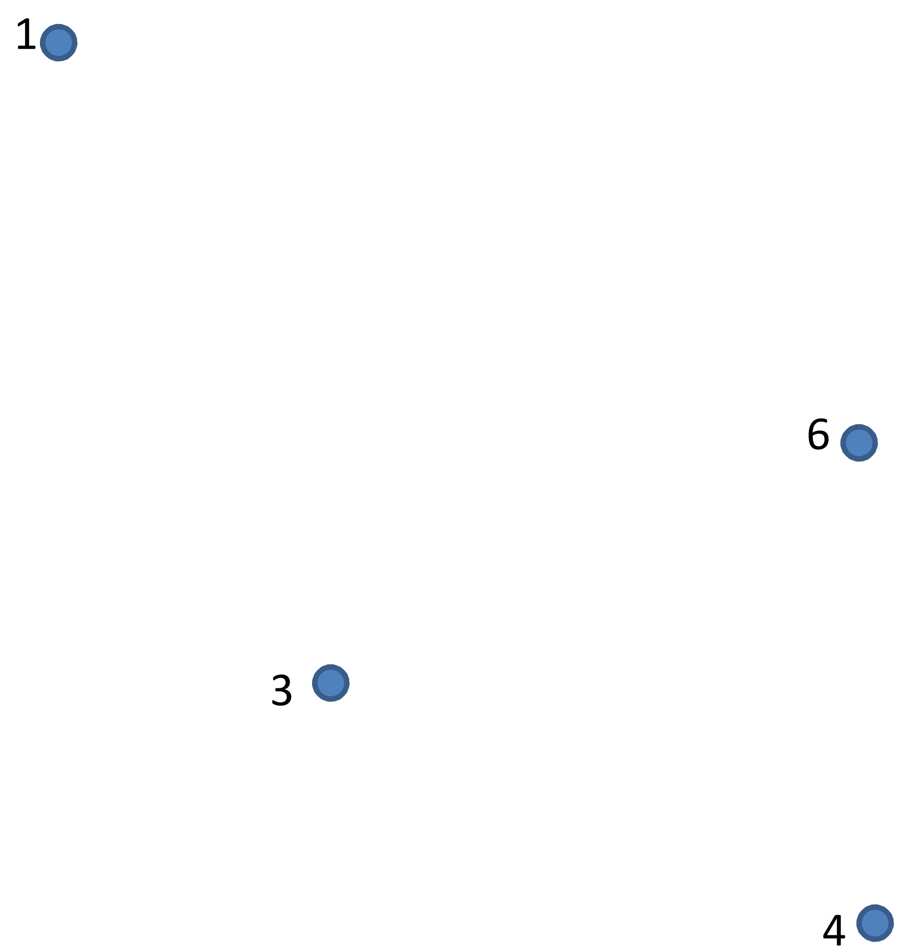
Nominated candidates remain



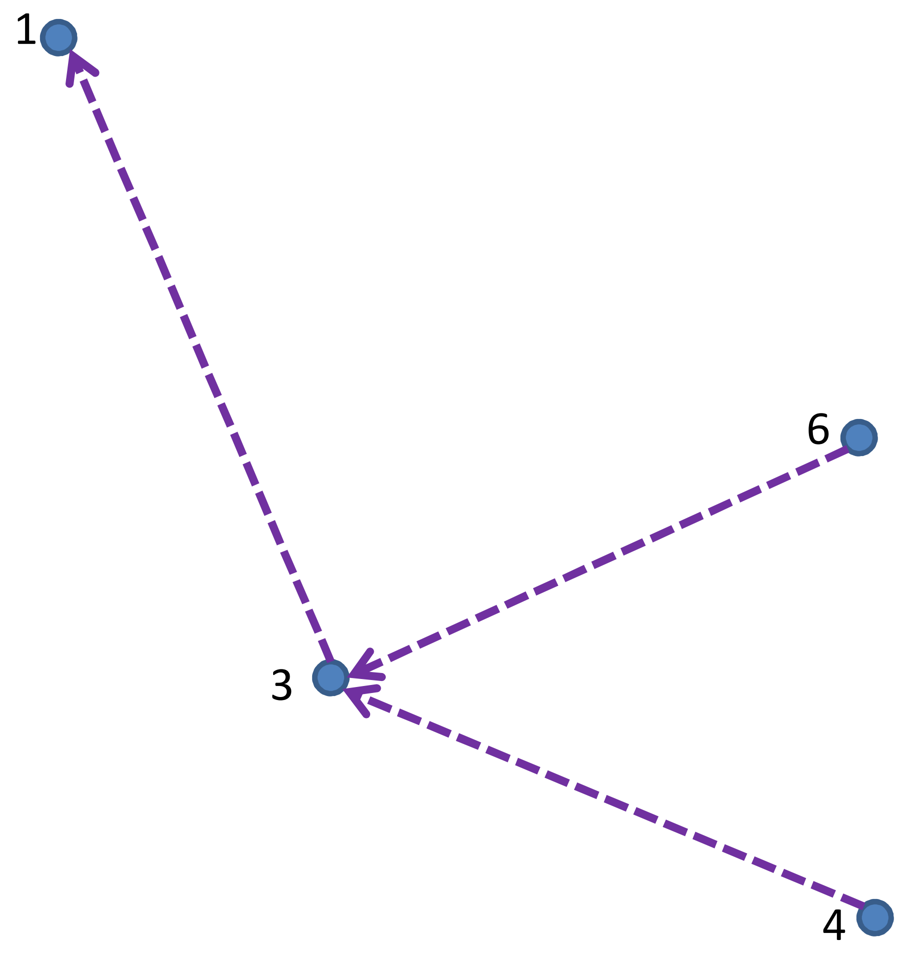
Flood 2-hops, nominate level 2 ancestor



Remaining nodes

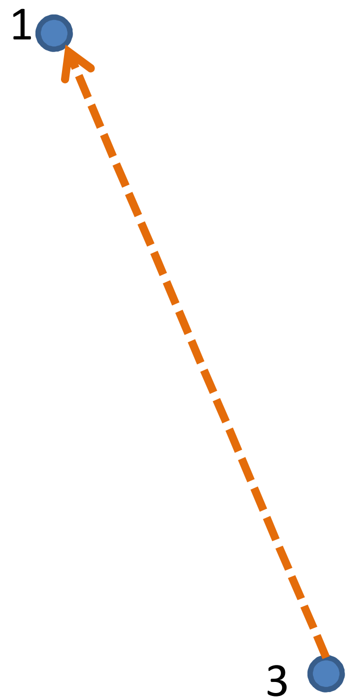


Flood 4-hops, nominate level 3 ancestor

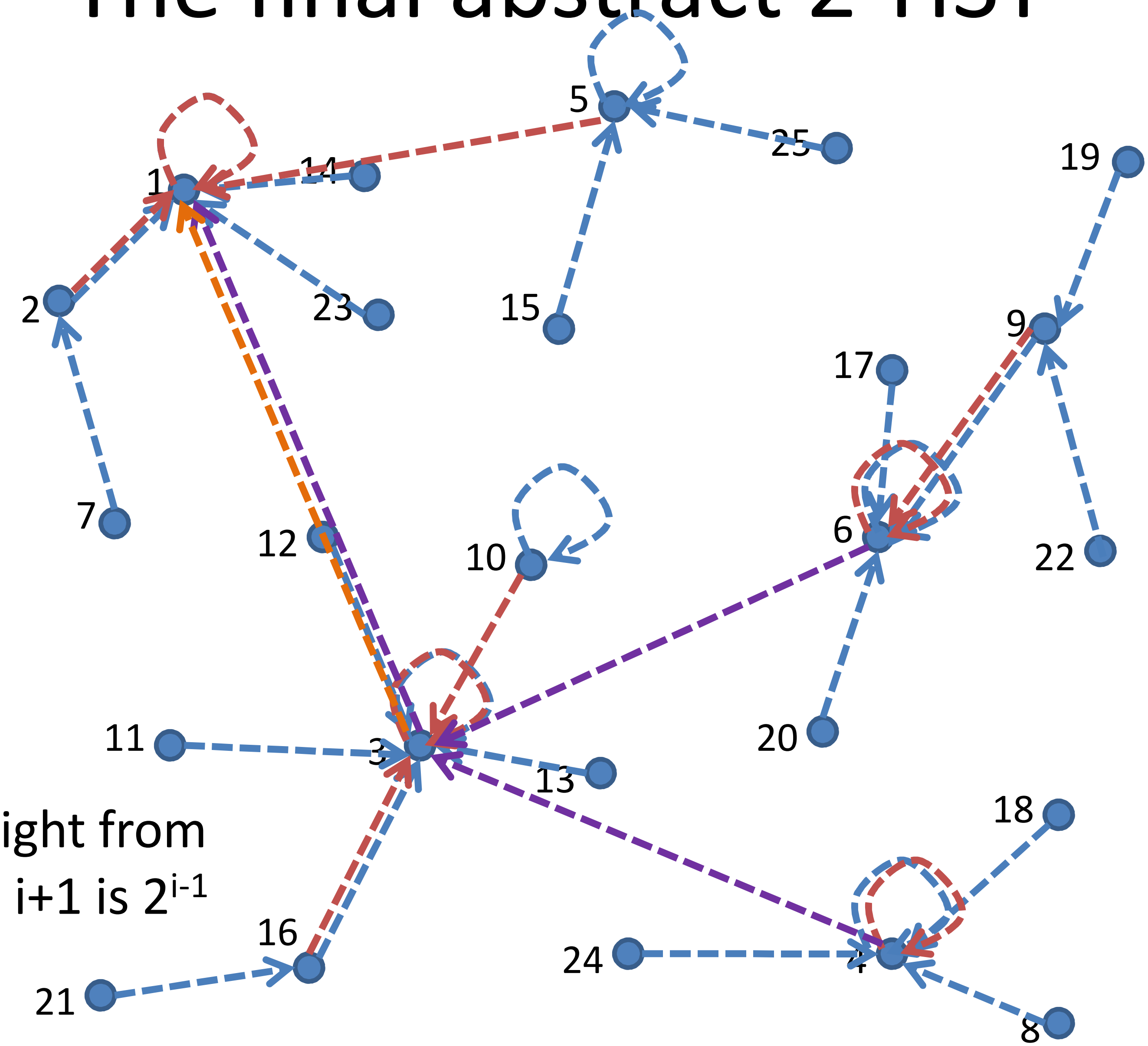


Remaining nodes

Flood 8-hops, nominate level 4 ancestor



The final abstract 2-HST



Edge weight from level i to i+1 is 2^{i-1}

Properties and analysis

- **O(log n)** distortion:

$$d(u,v) \le d_H(u,v) \le \log n \cdot d(u,v)$$

- The weight of the HST ≤ **O(log k) ·|MST|**
- **We show:** Communication cost is **O(nlogn)**
  - At round i,
  - communicate cost per candidate doubles.
  - # candidates decreases by half.
  - There are total O(logn) rounds.

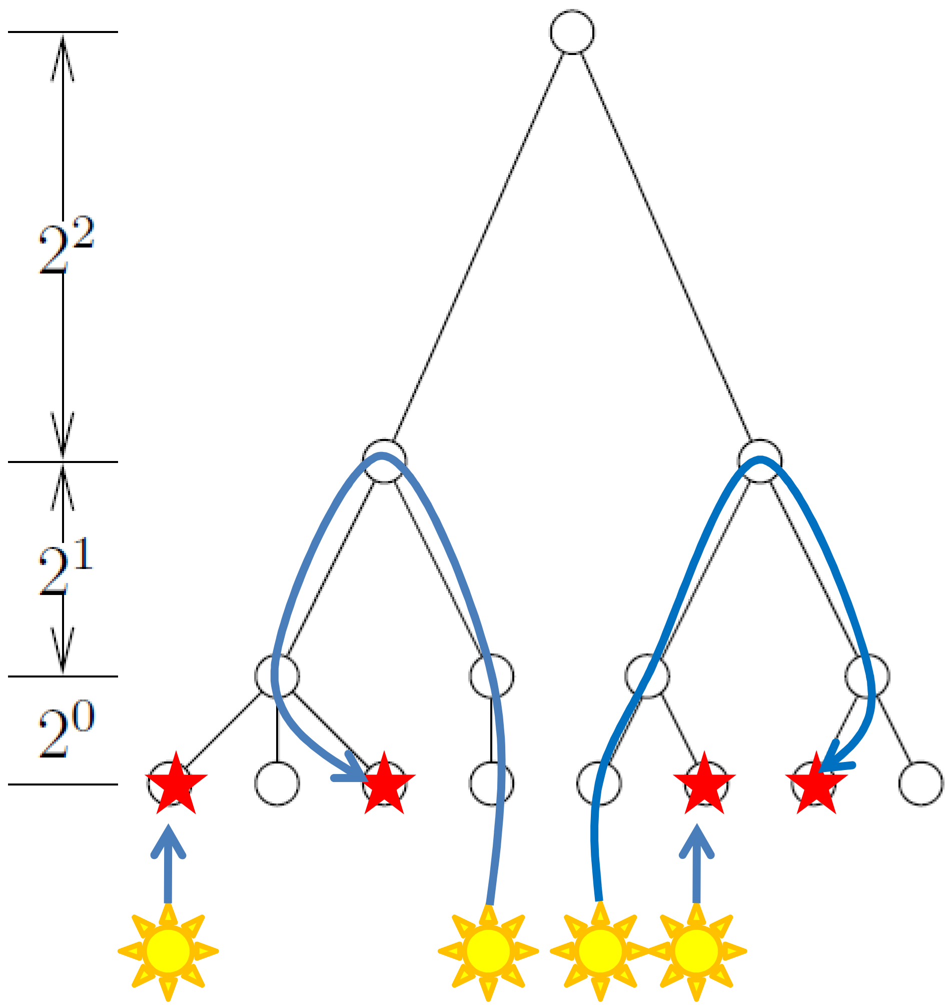
Assumption: the graph metric G has constant doubling dimension (i.e., packing argument holds).

Matching on HST

- k resources are given in advance
- Offline setting
  - k requests show up simultaneously
- Online setting
  - k requests come in one by one
  - Each request must be matched to some resource, and can't be changed later.

Offline setting

- Each resource sends a message to the root
- Each internal node keeps a counter on the # resources in the subtree.
- Each event sends a message to the root and is matched to the closest unmatched resource.

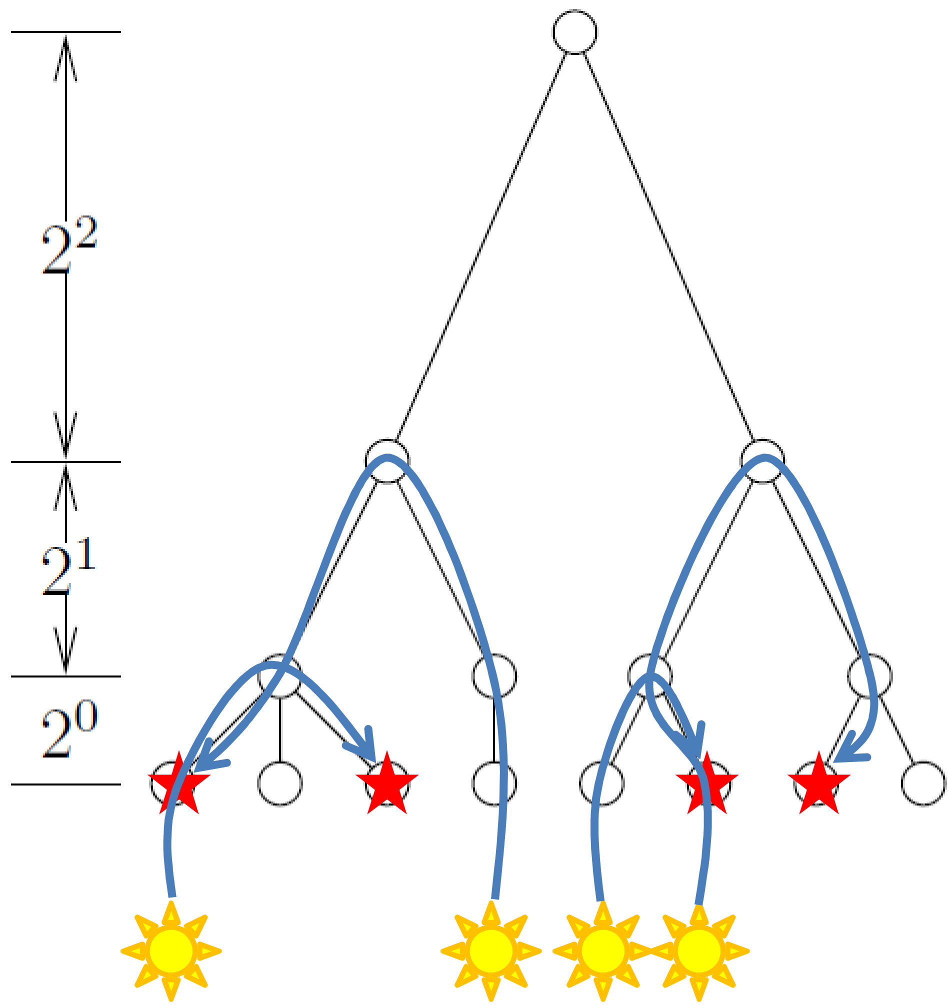


Offline setting

- Greedy matching is optimal on H and has approximation ratio **O(log k)** on G.
- The expected communication cost of a matching with length L is **O(L log k)**.

Online matching

- An event is matched to the **closest unmatched resource** on the HST.
- Ties are broken randomly.

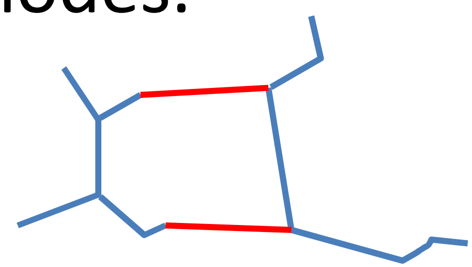


Online setting

- The expected communication cost of a matching with total length L is **$O(L^{\gamma})$**,
  where γ is the doubling dimension .
- Approximation ratio **$O(\log^3 k)$**.

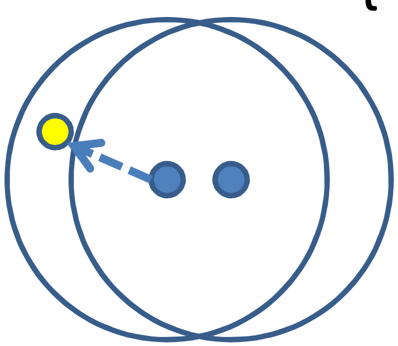
Conclusion with the 2nd application

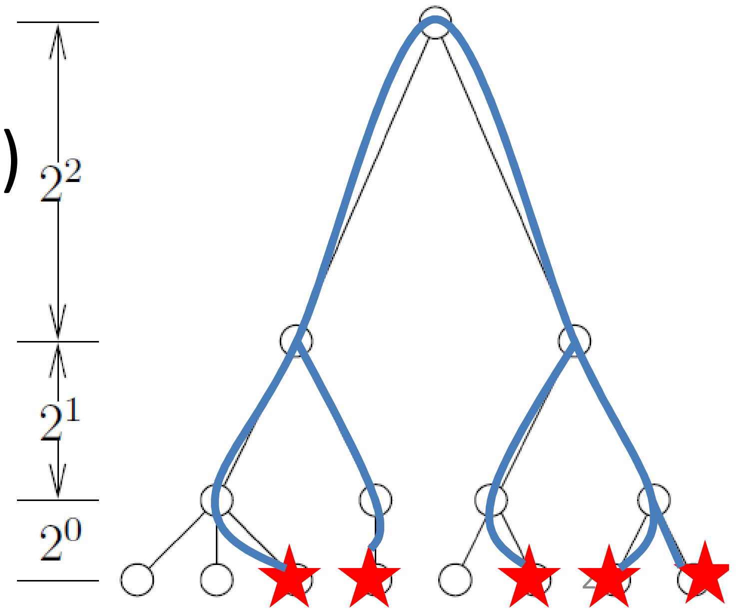
- Tree metric approximation is useful as many problems are simpler on trees.
- Maintain the **minimum Steiner tree** of **k mobile agents** in a sensor network of **n** nodes.
- Two challenges:
  - Where are the other agents?
    - Use location service. Extra cost.
  - Maintaining MST in the centralized setting is not easy.
    - Local movements may cause global changes.



Our solution

- Maintain the MStT on the HST **H**.
  - O(logn) approximation
  - No need of location service.
- Expected update cost is O(log n) per 1 hop move.
  - dist to LCA(u, v) at level i= $O(2^i)$
  - Prob{LCA at level i}=$O(1/2^i)$.





Papers

- Distributed resource management and matching in sensor networks, joint with Leonidas Guibas, Nikola Milosavljevic, Dengpan Zhou, IPSN'09.
- Maintaining Approximate Minimum Steiner Tree and k-center for Mobile Agents in a Sensor Network, joint with Dengpan Zhou, submitted, 2009.

Questions?