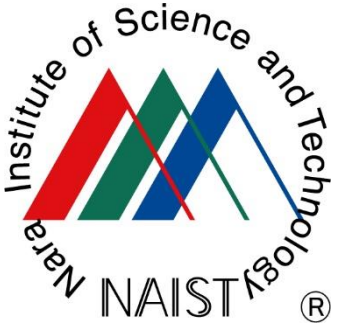
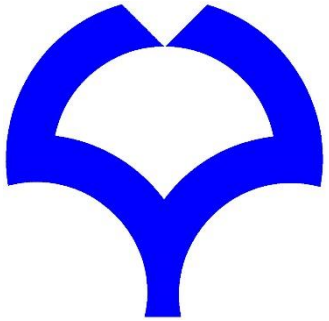
Self-stabilizing Rendezvous of Synchronous Mobile Agents in Graphs

Fukuhito Ooshita	Nara Institute of Science and Technology
Ajoy K. Datta	University of Nevada, Las Vegas
Toshimitsu Masuzawa	Osaka University



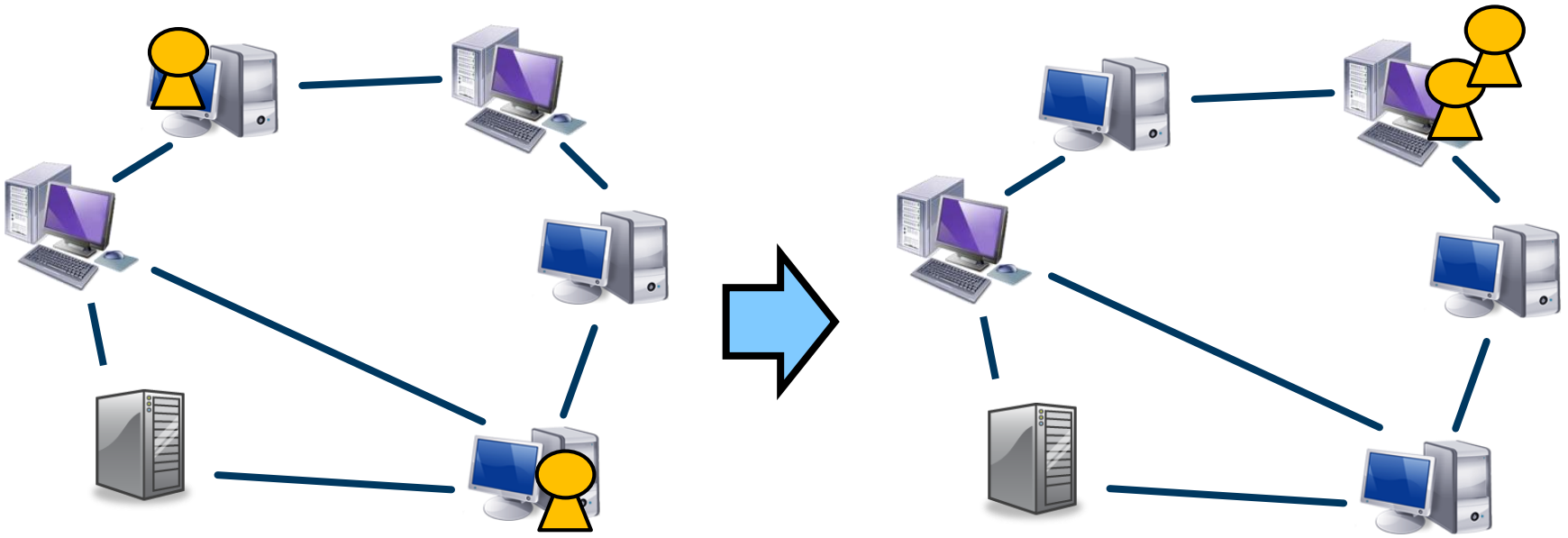


Rendezvous

- Two agents meet at a single node
  - Agents can exchange information at the meeting node

Initial configuration

Goal



Fault-tolerant Rendezvous

- Delay faults [1]
- Crash faults [2]
- Byzantine faults [3,4,5]

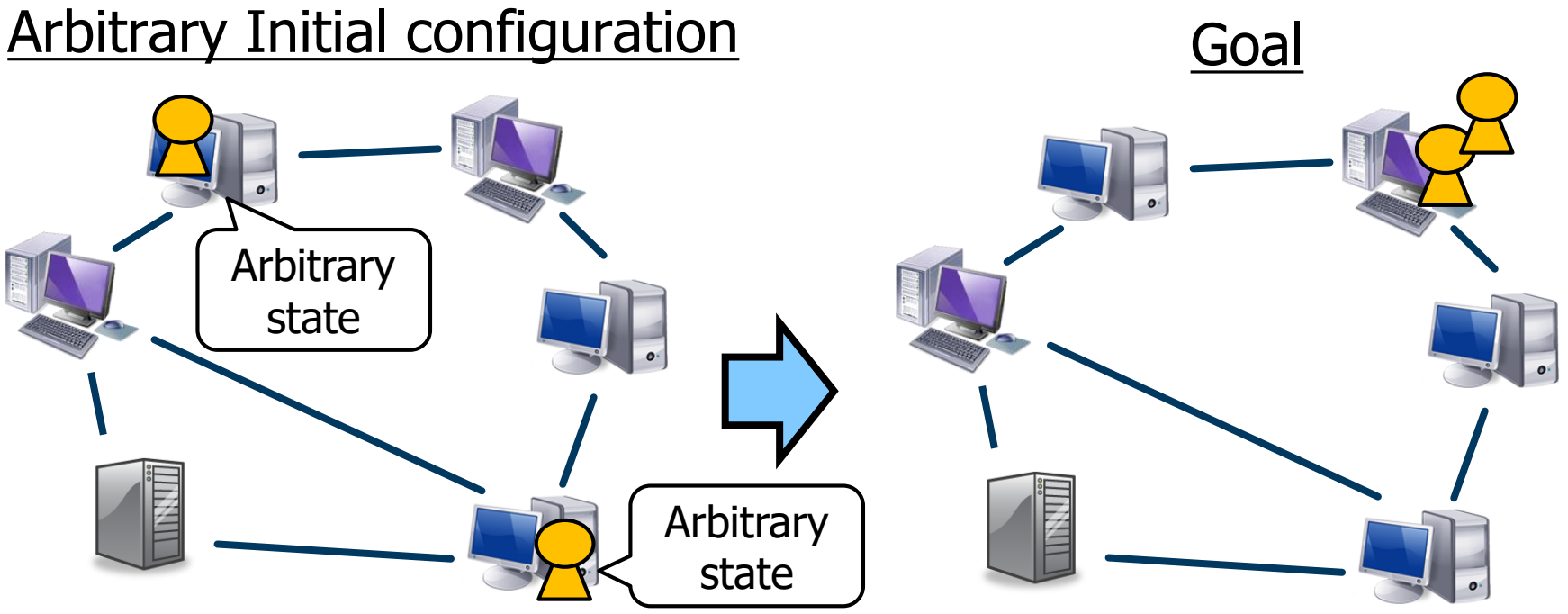
} Gathering (more than two agents)

- Transient faults (This Work)
  - Systems become arbitrary and inconsistent

[1] Chalopin, J., Dieudonn´e, Y., Labourel, A., Pelc, A.: Rendezvous in networks in spite of delay faults. Distrib. Comput. **29**, 187–205 (2016)
[2] Pelc, A.: Deterministic gathering with crash faults. CoRR abs/1704.08880 (2017). http://arxiv.org/abs/1704.08880
[3] Bouchard, S., Dieudonn´e, Y., Ducourthial, B.: Byzantine gathering in networks. Distrib. Comput. 29(6), 435–457 (2016)
[4] Dieudonn´e, Y., Pelc, A., Peleg, D.: Gathering despite mischief. ACM Tran. Algorithms 11(1), 1:1–1:28 (2014)
[5] Tsuchida,M., Ooshita, F., Inoue, M.: Byzantine gathering in networks with authenticated whiteboards. WALCOM, pp. 106–118 (2017)

Self-Stabilizing Rendezvous

- Two agents meet at a single node from **an arbitrary initial configuration**



Our Contributions

Self-stabilizing rendezvous algorithms

- Two synchronous agents with different labels
- No whiteboard

- An SS algorithm for arbitrary graphs
  - Convergence time: Not bounded

→ Can we reduce the convergence time?

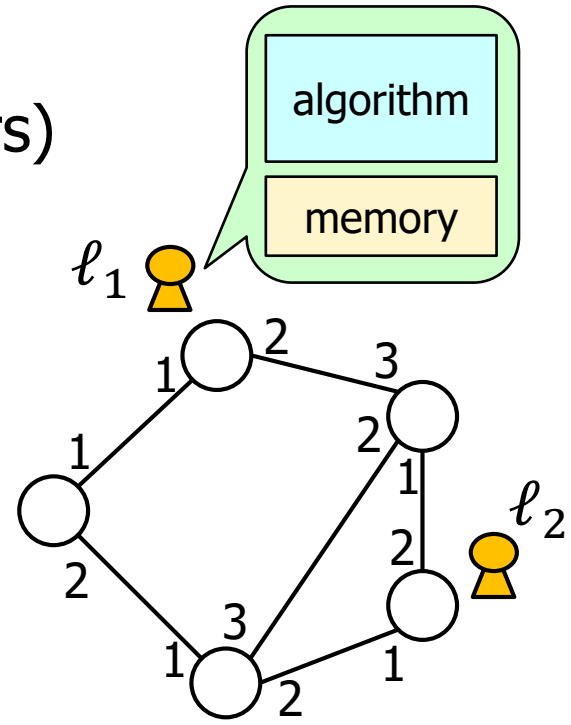
- SS algorithms for trees and rings
  - Convergence time: Polynomial

Outline

- Introduction
- Model & Problem
- SS Rendezvous for Arbitrary graphs
- SS Rendezvous for Trees
- SS Rendezvous for Rings
- Summary

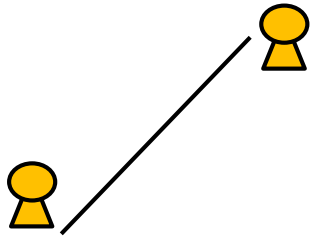
Model

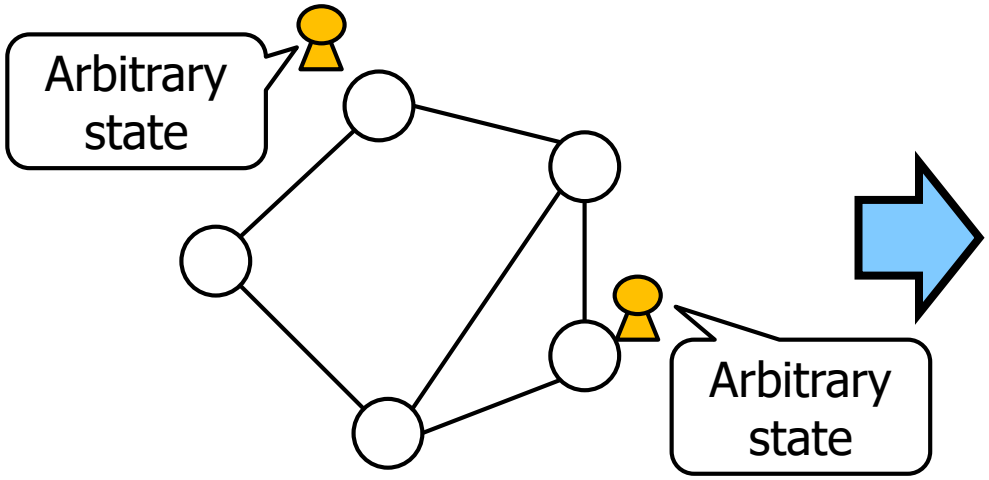
- Two (Mobile) Agents
  - have different labels (positive integers)
  - execute in synchronous rounds
  - start at different times
  - do not know #nodes
- Nodes
  - have port numbers to links
  - have no whiteboard
    - agents cannot leave any information

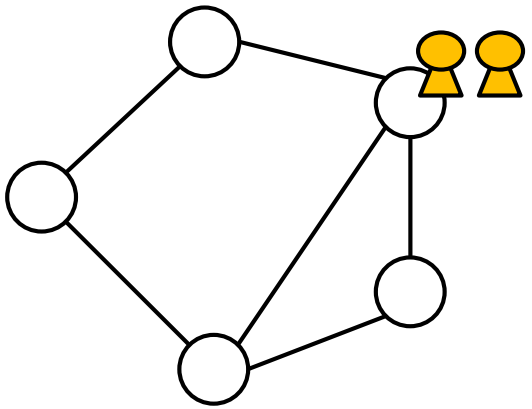


Self-Stabilizing Rendezvous

- Two agents meet at a single node from **an arbitrary initial configuration**
  - Arbitrary initial positions
  - Arbitrary initial states
- NOTE: Agents cannot meet inside a link





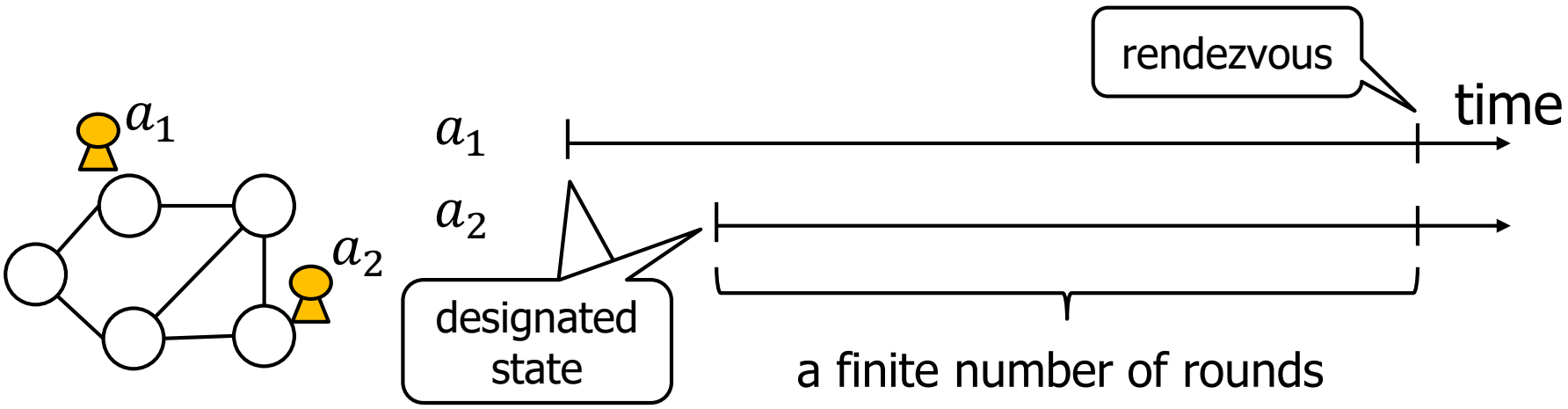


Outline

- Introduction
- Model & Problem
- SS Rendezvous for Arbitrary graphs
- SS Rendezvous for Trees
- SS Rendezvous for Rings
- Summary

Strategy

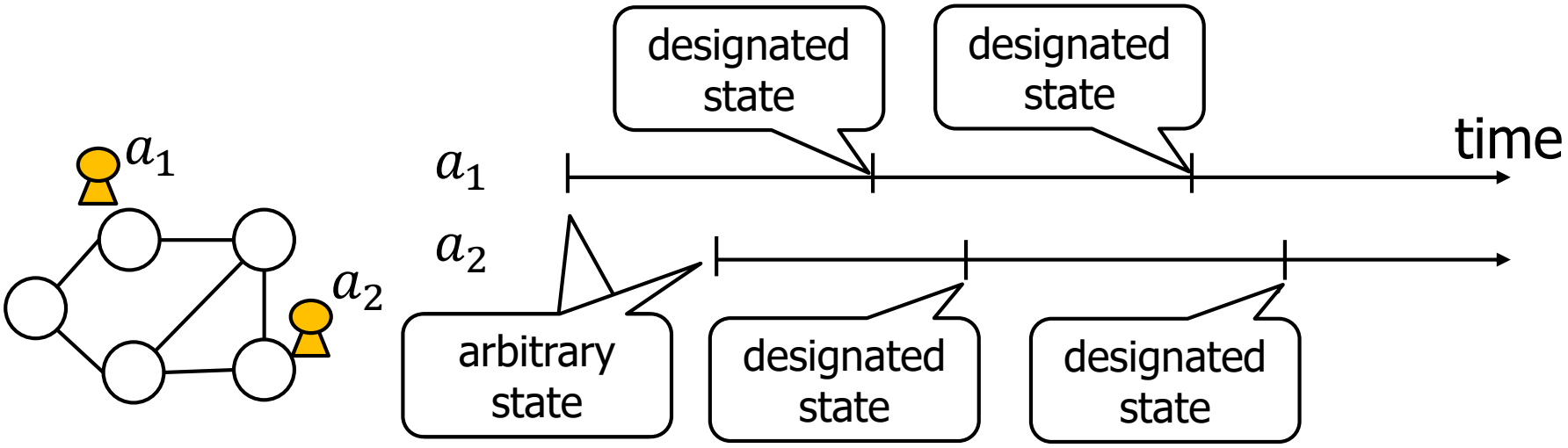
- Transform a non-SS algorithm to a SS one
- Non-SS algorithm (ex. [6])
  - Start from **designated** states and arbitrary positions possibly at different times
  - Achieve a rendezvous in a finite number of rounds



[6] Ta-Shma, A., Zwick, U.: Deterministic rendezvous, treasure hunts, and strongly universal exploration sequences. ACM Trans. Algorithms 10(3), 12:1–12:15 (2014)

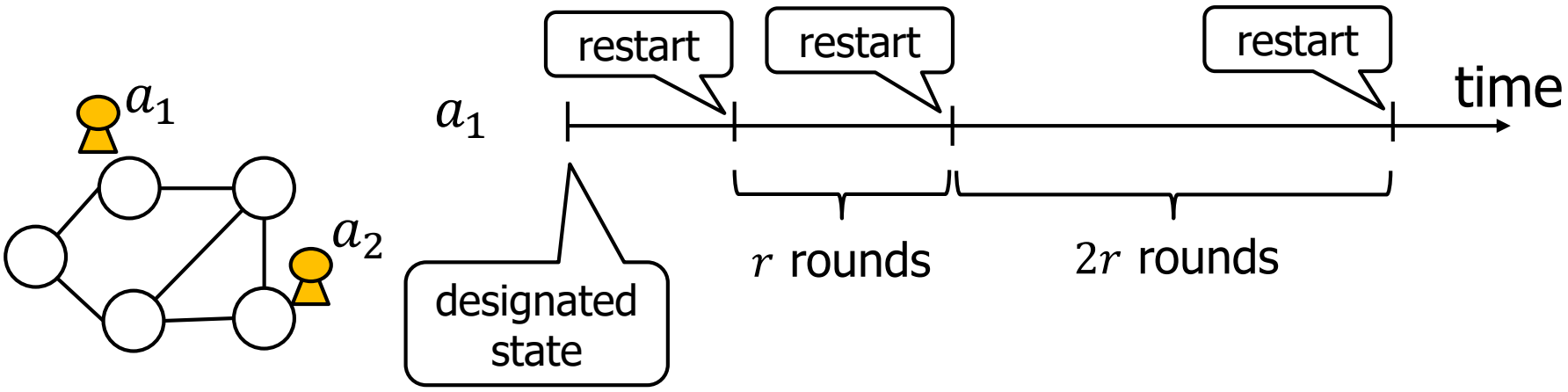
How to Transform (1/2)

- Difference
  - Non-SS: start from **designated** initial states
  - SS: start from **arbitrary** initial states
- Restart the non-SS algorithm regularly



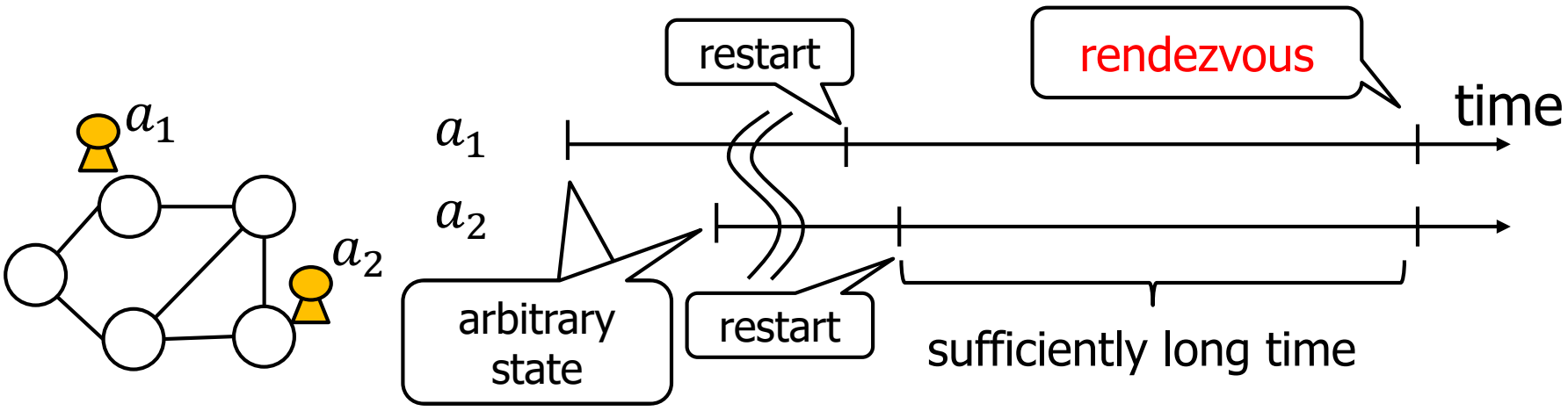
How to Transform (2/2)

- Requirement
  - Two agents execute the algorithm for a sufficiently long time without restarting
- Double the duration to the next restart



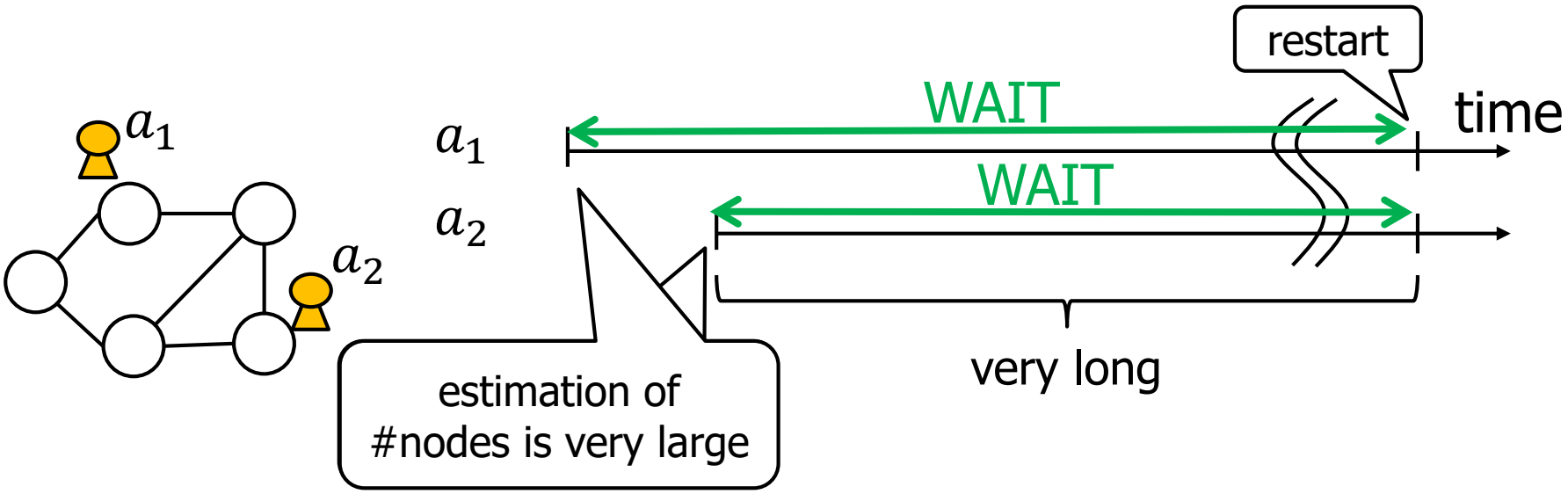
How to Transform (2/2)

- Requirement
  - Two agents execute the algorithm for a sufficiently long time without restarting
- Double the duration to the next restart



Time Complexity

- The convergence time is not bounded
  - The duration to the next restart may be very long
  - Agents may execute the non-SS algorithm with wrong estimation of #nodes
    - Wait for a period proportional to the estimation
      → Large estimation implies a long waiting period



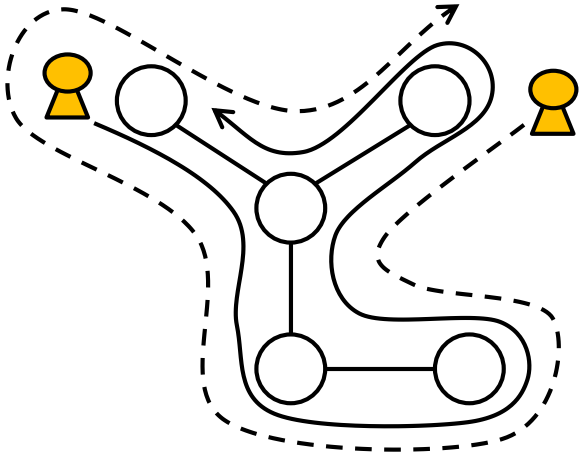
Outline

- Introduction
- Model & Problem
- SS Rendezvous for Arbitrary graphs
- SS Rendezvous for Trees
- SS Rendezvous for Rings
- Summary

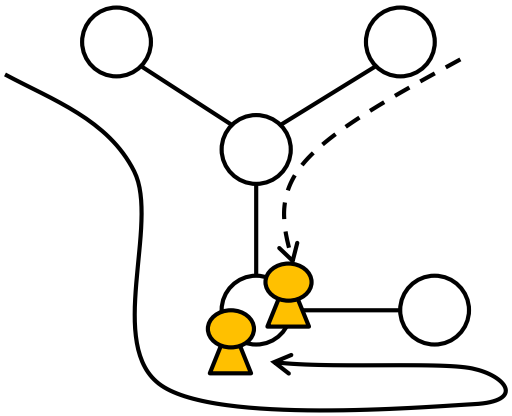
Strategy

- Make a period such that
  two agents traverse a tree in opposite directions
- Two agents can meet during this period

traverse in opposite directions



rendezvous

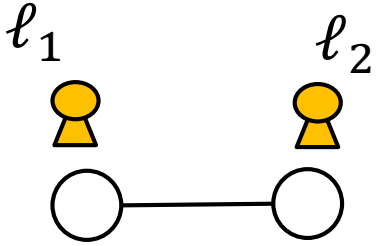


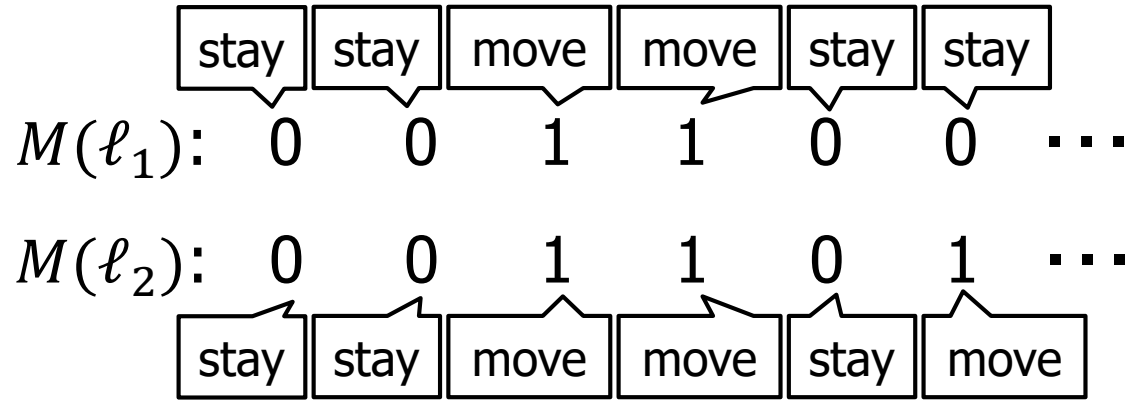
Make the Period (1/2)

- Apply algorithm Extend-Labels [7]
  - Two agents achieve a rendezvous in a 2-node graph in a self-stabilizing manner
  - Each agent moves based on extended-label $M(\ell_i)$
    - Bit 1 → move
    - Bit 0 → stay

  → One agent moves and the other stays in some round

ℓ_i: original label



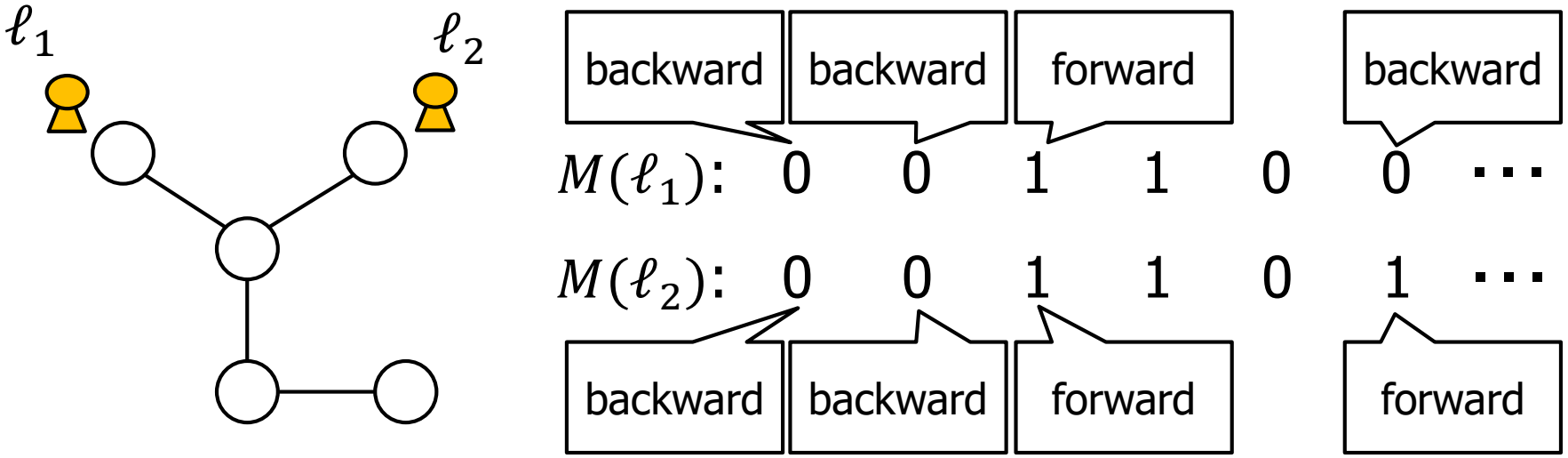


[7] Dessmark, A., Fraigniaud, P., Kowalski, D.R., Pelc, A.: Deterministic rendezvous in graphs. Algorithmica 46, 69–96 (2006)

Make the Period (2/2)

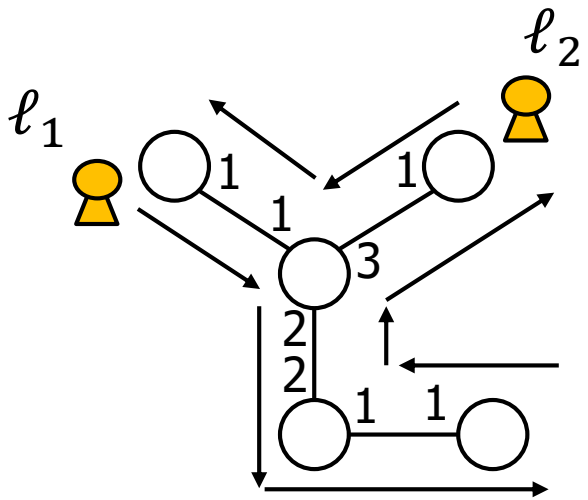
- Change the behavior of Extend-Label
  - Bit 1 → traverse a tree in the forward direction
  - Bit 0 → traverse a tree in the backward direction

In some period,
two agents traverse a tree in the opposite directions



How to Traverse a Tree

- Basic walks
  - arrive from port p → leave from port $p+1$
  - $2(n-1)$ basic walks realize a forward traversal
- Reverse walks
  - arrive from port p → leave from port $p-1$
  - $2(n-1)$ reverse walks realize a backward traversal

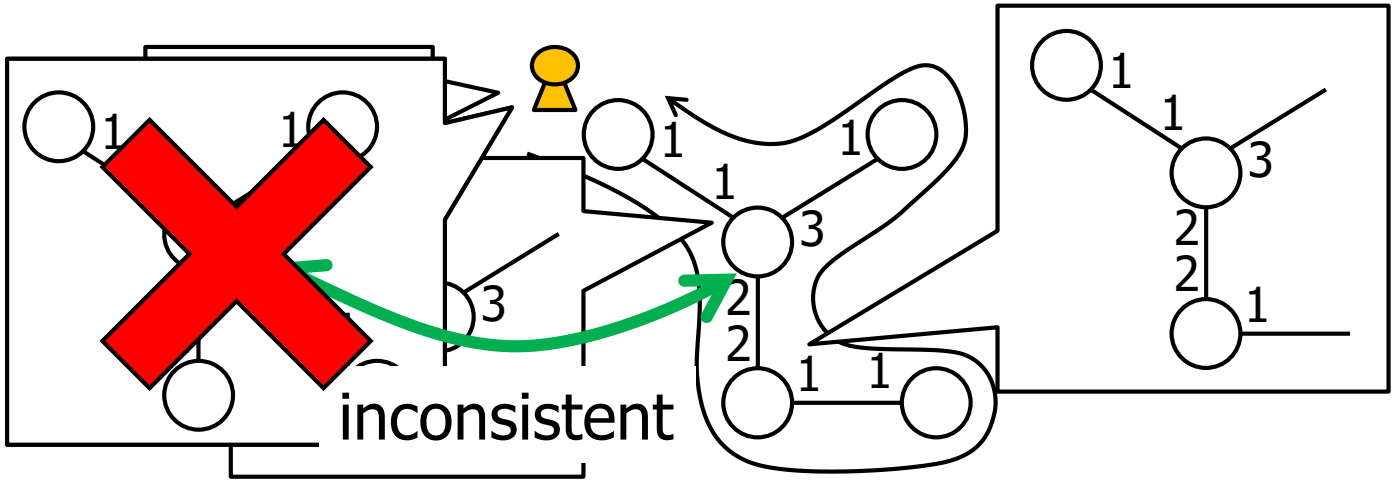


n: #nodes

Agents require #nodes

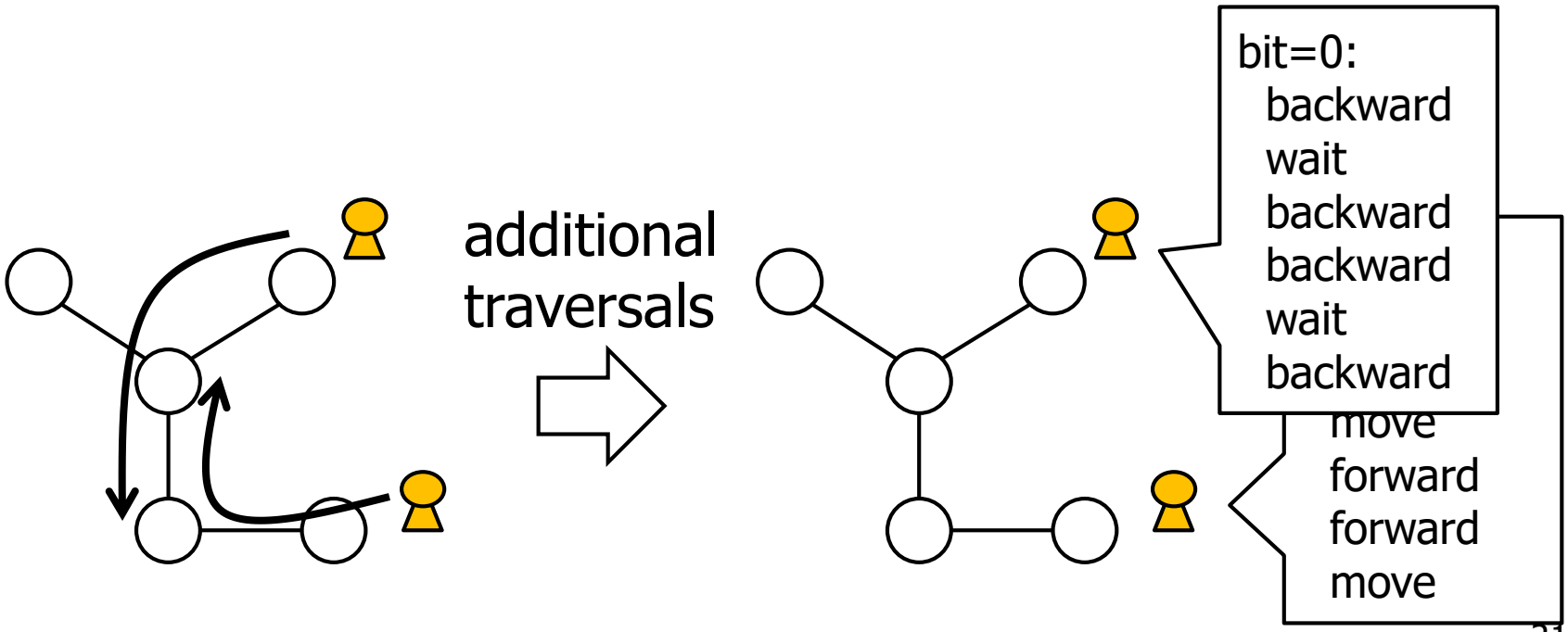
How to Obtain n (#Nodes)

- Record topology while moving
  - Compute #nodes from the recorded topology
- Check consistency of the recorded topology while moving
  - Find inconsistency → Discard the recorded topology



Additional traversals

- Issue
  - Two agents may pass through the same link in the opposite directions
- Solution: Agents make additional traversals
  - Insert waits and change visit times of nodes

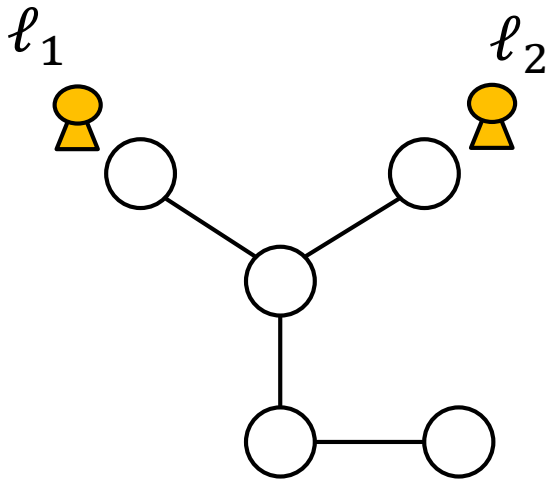


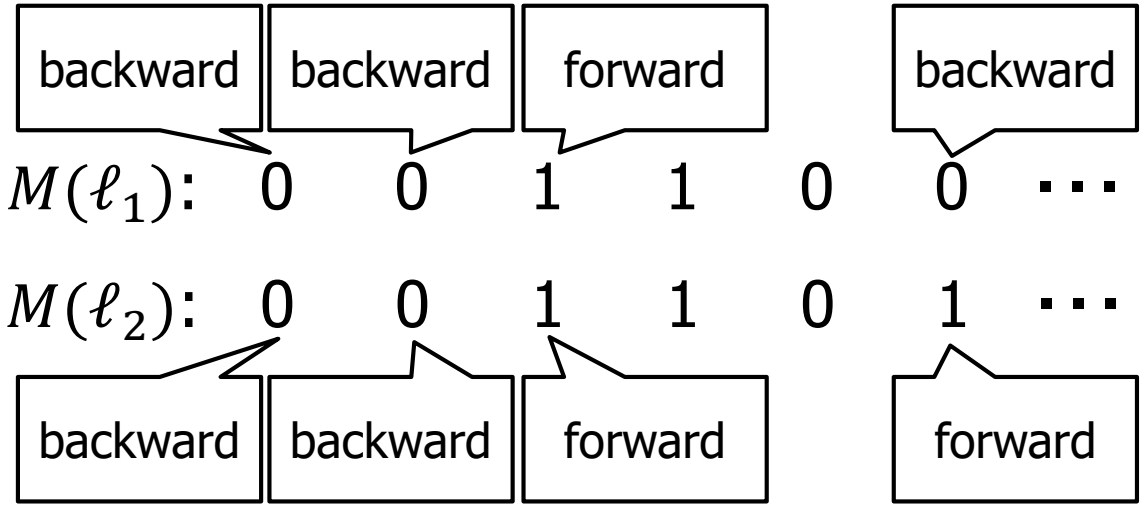
Time Complexity

n: #nodes

- Obtain correct #nodes in $O(n)$ rounds
- $O(\min\{|\ell_1|, |\ell_2|\})$-th bits are different
  - Each bit implies $O(n)$ rounds

Achieve a rendezvous in $O(n \cdot \min\{|\ell_1|, |\ell_2|\})$ rounds





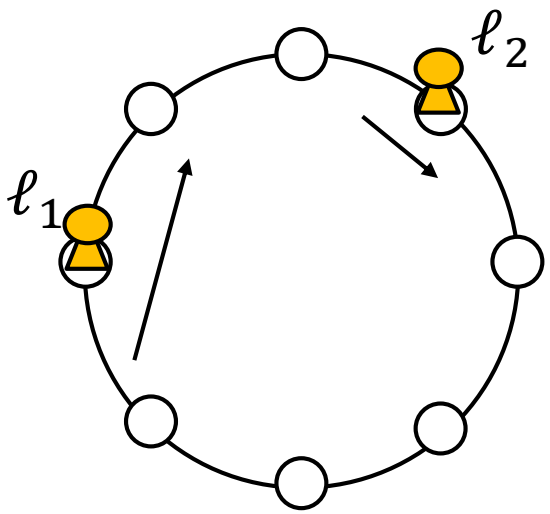
Outline

- Introduction
- Model & Problem
- SS Rendezvous for Arbitrary graphs
- SS Rendezvous for Trees
- SS Rendezvous for Rings
- Summary

Strategy

- Agents move in different speeds
- Behavior of agent with label ℓ_i
  - Repeat the following
    - Wait for ℓ_i rounds and then move forward once

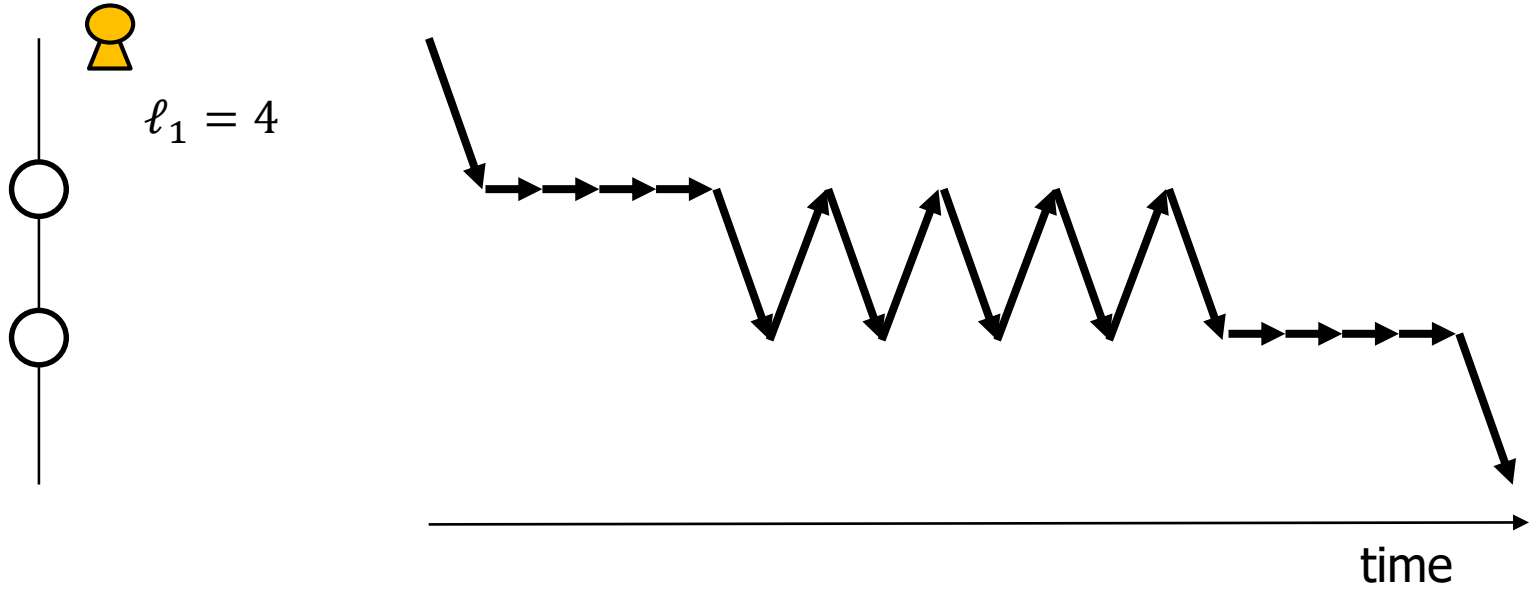
If two agents move in the same direction, they can achieve a rendezvous



If two agents move in the opposite directions, they can pass each other

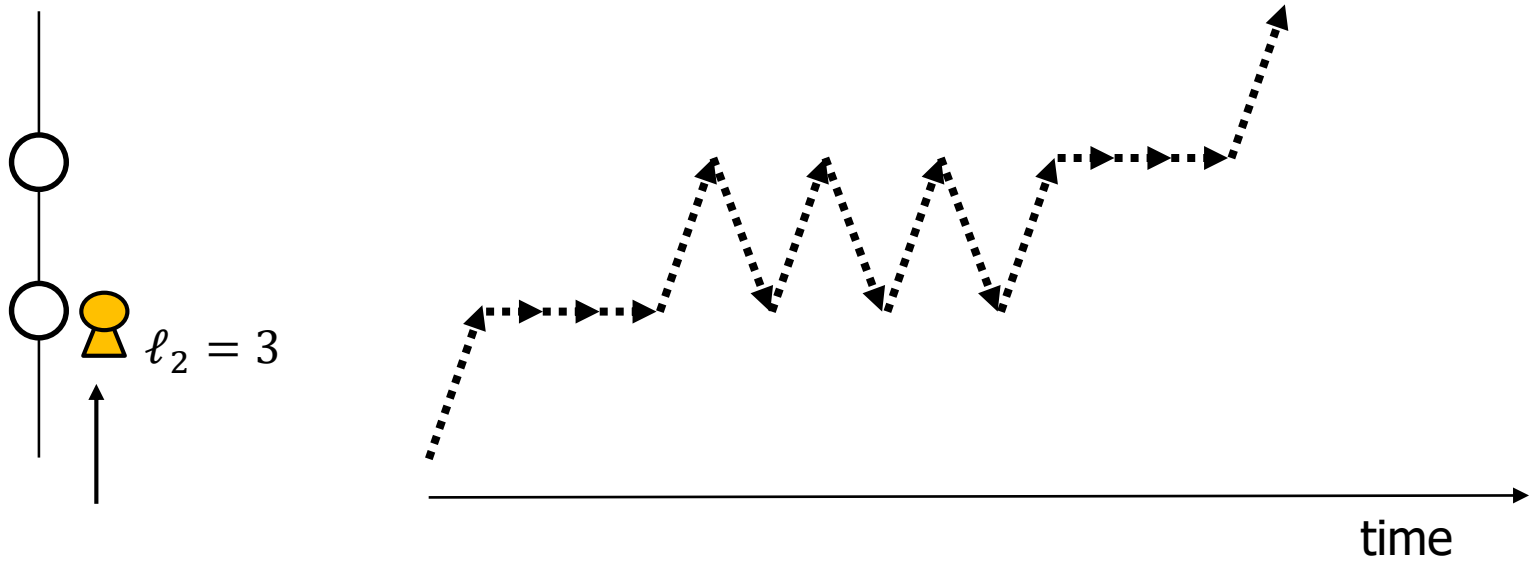
How to Prevent Passing

- Before moving forward,
  agent executes ℓ_i sweeping operations
  - One sweeping operation
    = one forward move and one backward move



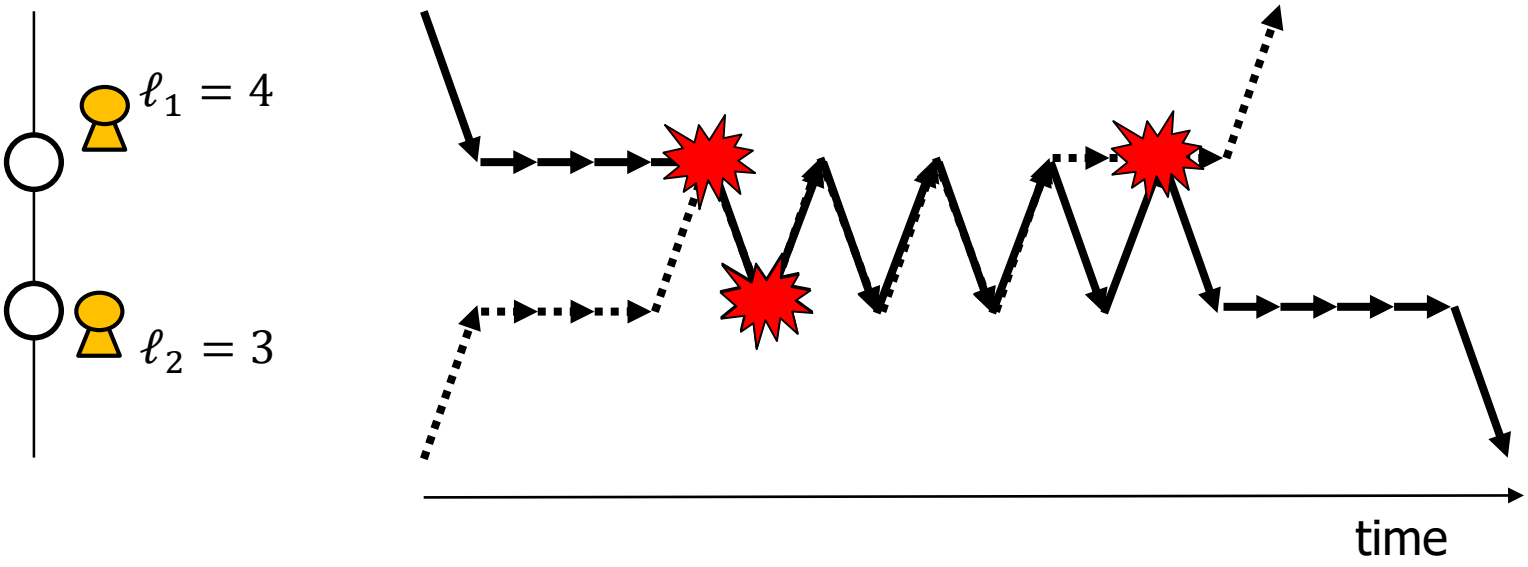
How to Prevent Passing

- Before moving forward,
  agent executes ℓ_i sweeping operations
  - One sweeping operation
    = one forward move and one backward move



How to Prevent Passing

- Before moving forward,
  agent executes ℓ_i sweeping operations
  - One sweeping operation
    = one forward move and one backward move

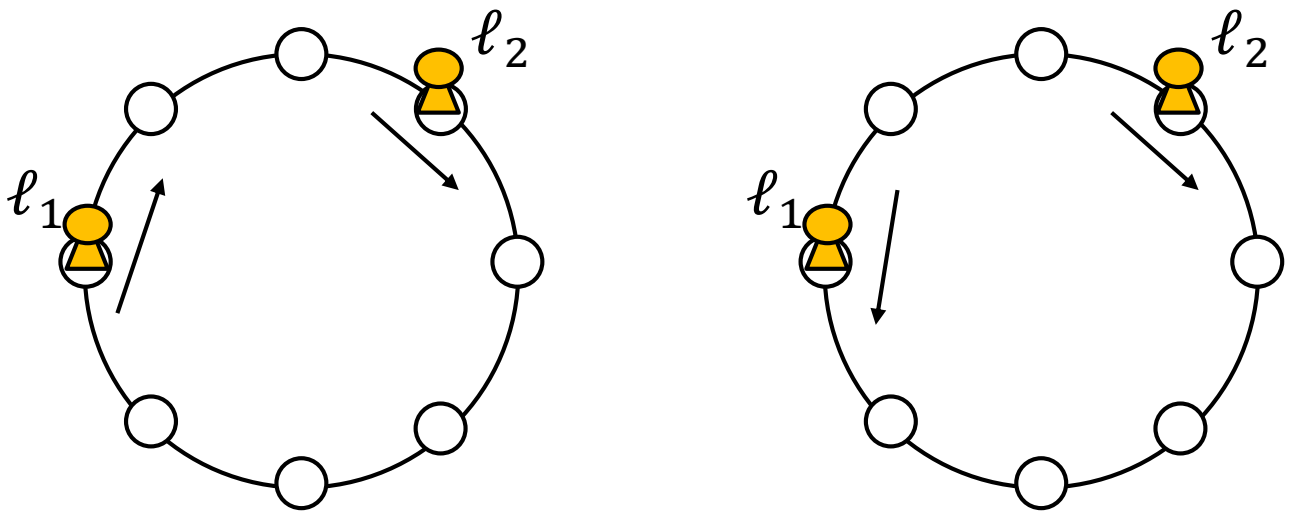


Time Complexity

ℓ_1, ℓ_2: labels
n: #nodes

- It takes more time
  when two agents move in the same direction
  - Agents move once in $O(\ell_1)$ and $O(\ell_2)$ rounds
  - The distance between two agents decreases
    in $O(\ell_1 \ell_2)$ rounds

Achieve a rendezvous in $O(n\ell_1\ell_2)$ rounds



Summary

Self-stabilizing rendezvous algorithms

- Two synchronous agents with different labels ℓ_1 and ℓ_2
- No whiteboard

◆ Arbitrary graphs

- Transformer from a non-SS algorithm to a SS one
- Convergence time: Not bounded

◆ Trees

- Convergence time: $O(n \cdot \min\{|\ell_1|, |\ell_2|\})$

◆ Rings

- Convergence time: $O(n\ell_1\ell_2)$

n: #nodes

End