Randomized Byzantine Gathering in Rings

John Augustine, Arnhav Datar, **Nischith Shadagopan**

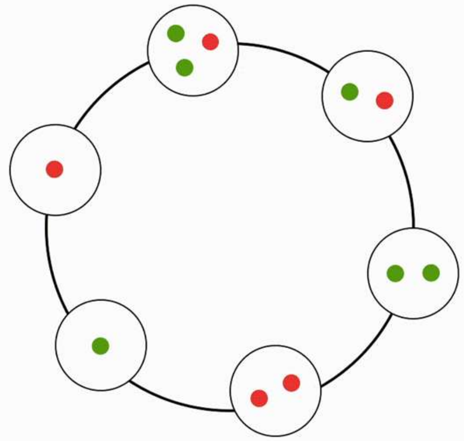
Indian Institute of Technology, Madras

Introduction

- What is swarm robotics?
- Why gathering?
- Why rings?

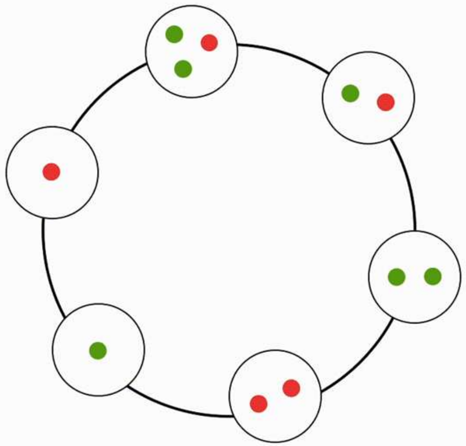
Problem Statement

- k: total number of agents
- n: number of nodes in the ring
- f: number of byzantine agents



Model Description

- Agents operate synchronously



Model Description

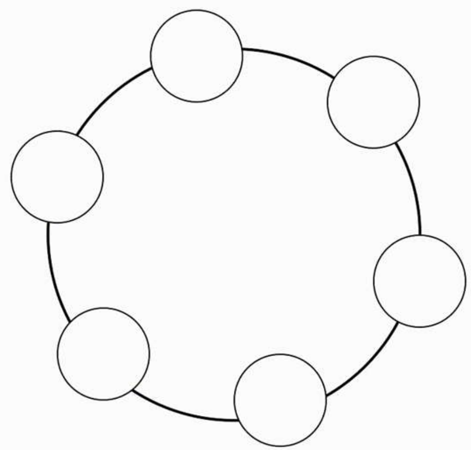
- Agents operate synchronously
- Nodes are identical

Model Description

- Agents operate synchronously
- Nodes are identical
- One round, one node

Model Description

- Agents operate synchronously
- Nodes are identical
- One round, one node
- Agents are anonymous



Model Description

- Agents operate synchronously
- Nodes are identical
- One round, one node
- Agents are anonymous
- Local broadcast communication
- Aware of n and an upper bound on f
- Ability to compute and store information

Agent Description

In each round, each agent can:

- Local broadcast
- Receive messages
- Transition state
- Move
- Update state

Adversary Description

- Controlled by a single adversary
- Computationally unbounded
- Can deviate from the protocol
- Can distinguish the nodes of the ring
- Also aware of starting distribution of agents
- Strongly adaptive

Standard Gathering Algorithm

- The algorithm runs in stages. Each stage is divided into phases.
- At the starting of each stage, sufficiently small groups become inactive
- In each phase, each group decides to be either a leader or a follower uniformly randomly
- The leader groups just move clockwise for n/2 rounds
- The follower groups move counterclockwise for n/2 rounds until they meet a sufficiently large leader group
  - On meeting such a leader group, they behave exactly like a leader group till the phase ends

Key Insights Into The Algorithm

- In this algorithm, two good agents never separate once they become part of the same group
- It can be shown that at the beginning of the j^{th} stage there is at least one group with 2^j number of good agents with high probability
- So eventually we get a group having a greater number of agents than the total number of Byzantine agents
- All good agents can then gather very easily
- The algorithm runs in O(n *log*n *log*k) rounds and can handle O(k / *log* k) number of Byzantine agents

What Can The Adversary Do?

- There is no use in imitating a follower group as leader groups are not responsive
- Suppose a Byzantine group imitates a leader group
  - A good follower group might encounter it and presume that it has combined with another group
  - This good follower group now behaves as a leader group for the rest of the phase
  - The Byzantine group cannot encounter other follower groups by moving in the counterclockwise direction, it must move clockwise
  - The newly formed leader group also moves clockwise. Therefore, any other follower group meets the newly formed good leader group no later than the Byzantine group

Visual Tracking Setting

- We allow the agents to visually track other agents.
- The algorithm runs in phases
- In each phase, each group decides to be either a leader or a follower uniformly randomly
- The leader groups just move clockwise for n/2 rounds
- The follower groups move counterclockwise for n/2 rounds
    - After this, the follower group chooses one leader group to combine with based on the size of the leader group and moves to the relevant location

Key Insights Into The Algorithm

- Again, two good agents never separate once they become part of the same group
- Since the motion of the good agents are predictable, their location at the end of each phase is predictable
- The algorithm runs in O(n *log*n) rounds and can handle any constant fraction of the total number of agents being Byzantine

What Can The Adversary Do?

- There is no use in imitating a follower group as leader groups are not responsive
- Suppose a Byzantine group imitates a leader group
  - If it tags along with a follower group and behaves like a leader group repeatedly, the follower group can recognize this due to visual tracking
  - If the Byzantine group does not tag along, then it can never catch up to the follower group again

Read The Paper For

- Proofs of correctness and time complexity analysis
- Adaptations of the algorithms for when the agents don't have a common orientation

Setting	Time Complexity	Max. Byz. Agents
Coalescing random walks [8]	$\tilde{\mathcal{O}}(n^2)$	$f < k$
With No Byzantine Agents	$\mathcal{O}(n)$ (on exp.)	$f = 0$
Standard setting	$\mathcal{O}(n \log n \log k)$	$f(35 + 34\log_2(f+1)) + 2 < k$
With visual tracking	$\mathcal{O}(n \log n)$	$f = \alpha k$, $0 \leq \alpha < 1$
Lower bound	$\Omega(n)$	$f = 0$

Table 1 Known results on randomized gathering protocols including those inferred from the literature on coalescing random walks. The $\tilde{\mathcal{O}}$ notation is used to hide polylog(n) factors.

Acknowledgements

- I would like to thank my advisor John Augustine and collaborator Arnhav Datar
- I would like to thank the Anonymous reviewers for their thoughtful suggestions
- I would like to thank Tarini Naik for helping with the animations for the slides