INTERNATIONAL CONFERENCE ON PARALLEL PROCESSING

ICPP / 2021 / CHICAGO / USA

acm In-Cooperation

sighpc

AUGUST 9-12, 2021

Self-Stabilization with Selfish Agents

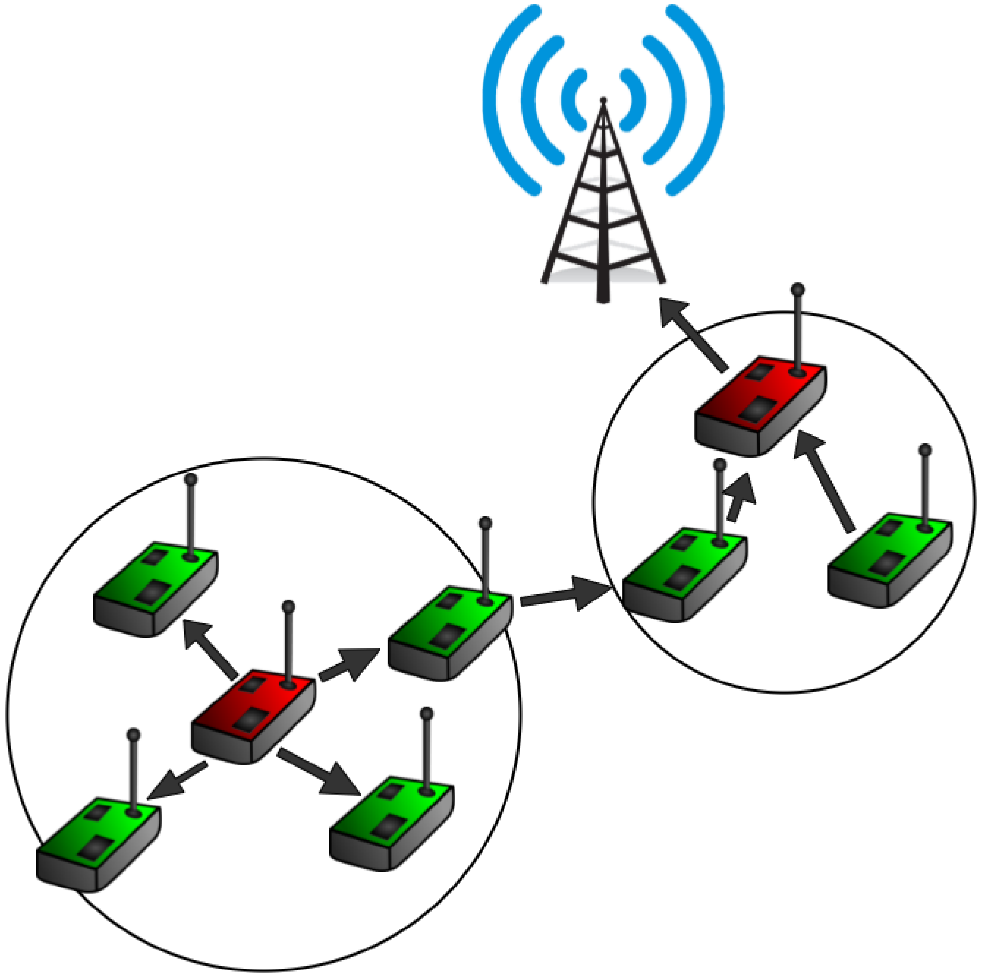
Amir Reza Ramtin, Don Towsley

University of Massachusetts Amherst

August 2021

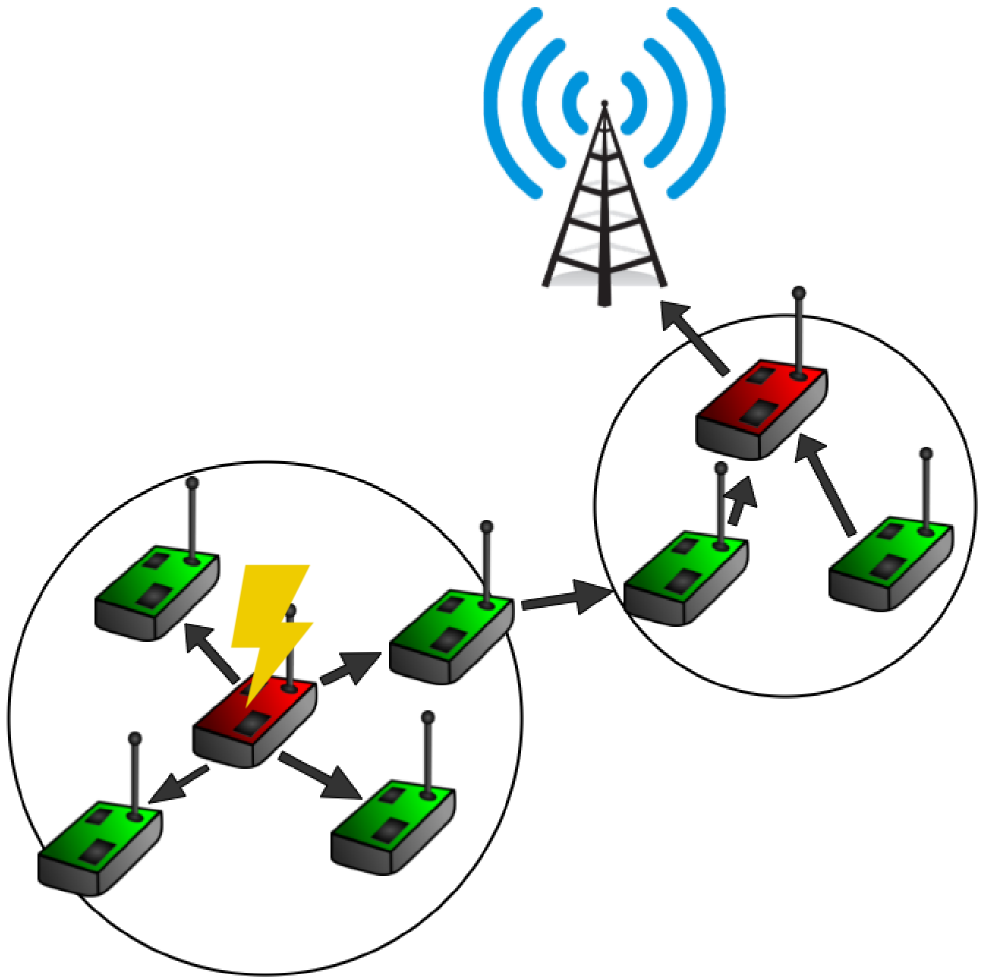
Challenge

- Wireless sensor networks
  - Fault prone
  - Dynamic topology
  - Energy efficiency



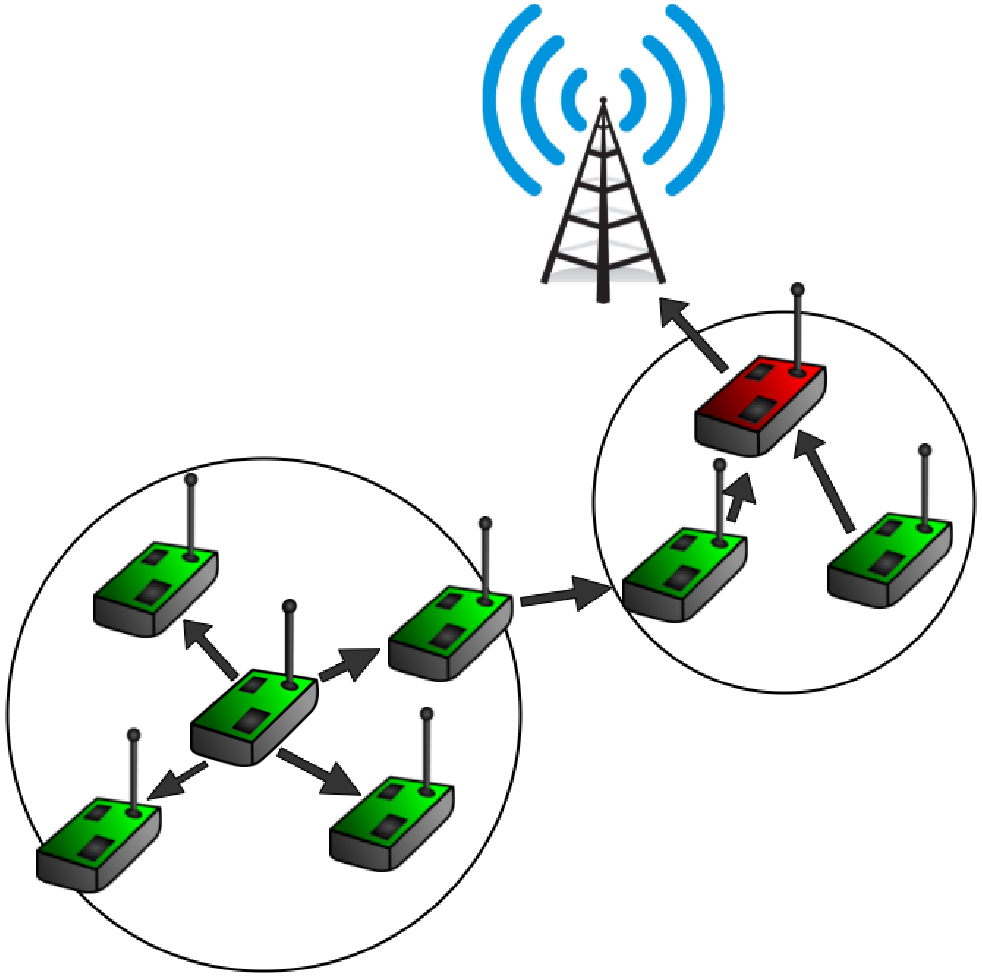
Challenge

- Wireless sensor networks
  - Fault prone
  - Dynamic topology
  - Energy efficiency



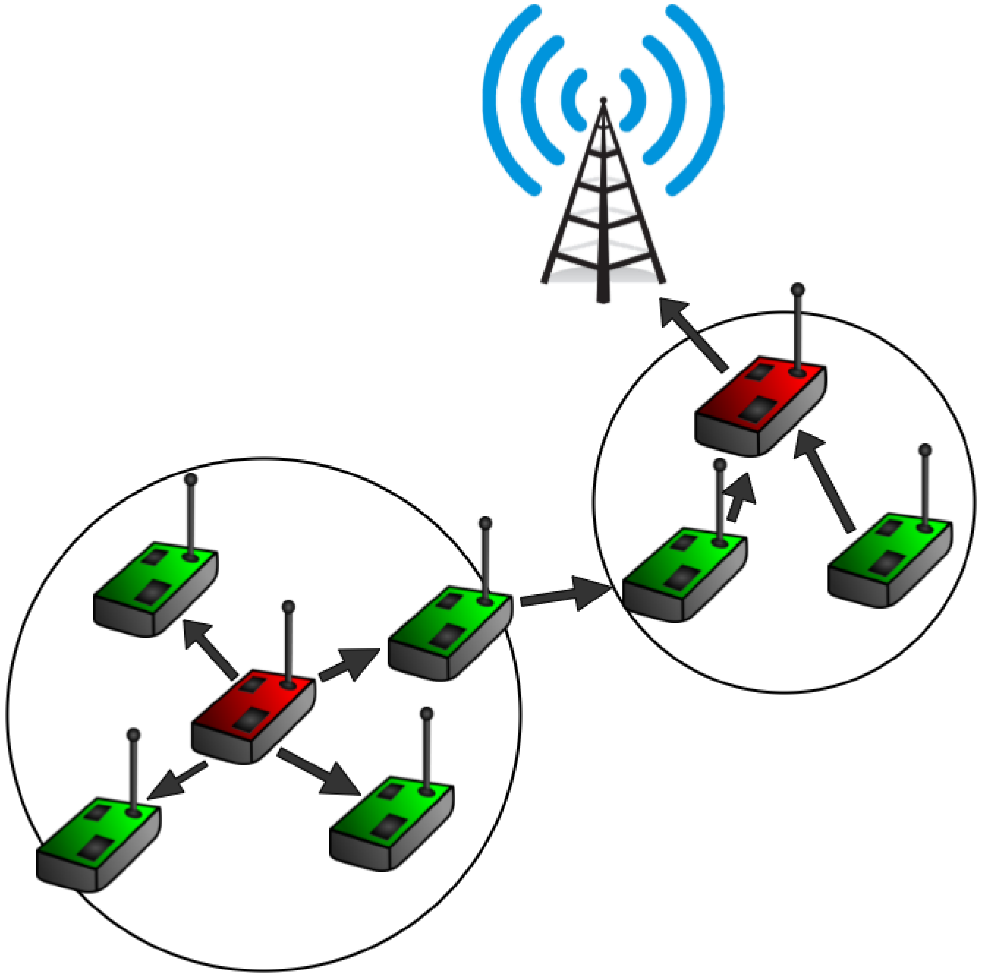
Challenge

- Wireless sensor networks
  - Fault prone
  - Dynamic topology
  - Energy efficiency



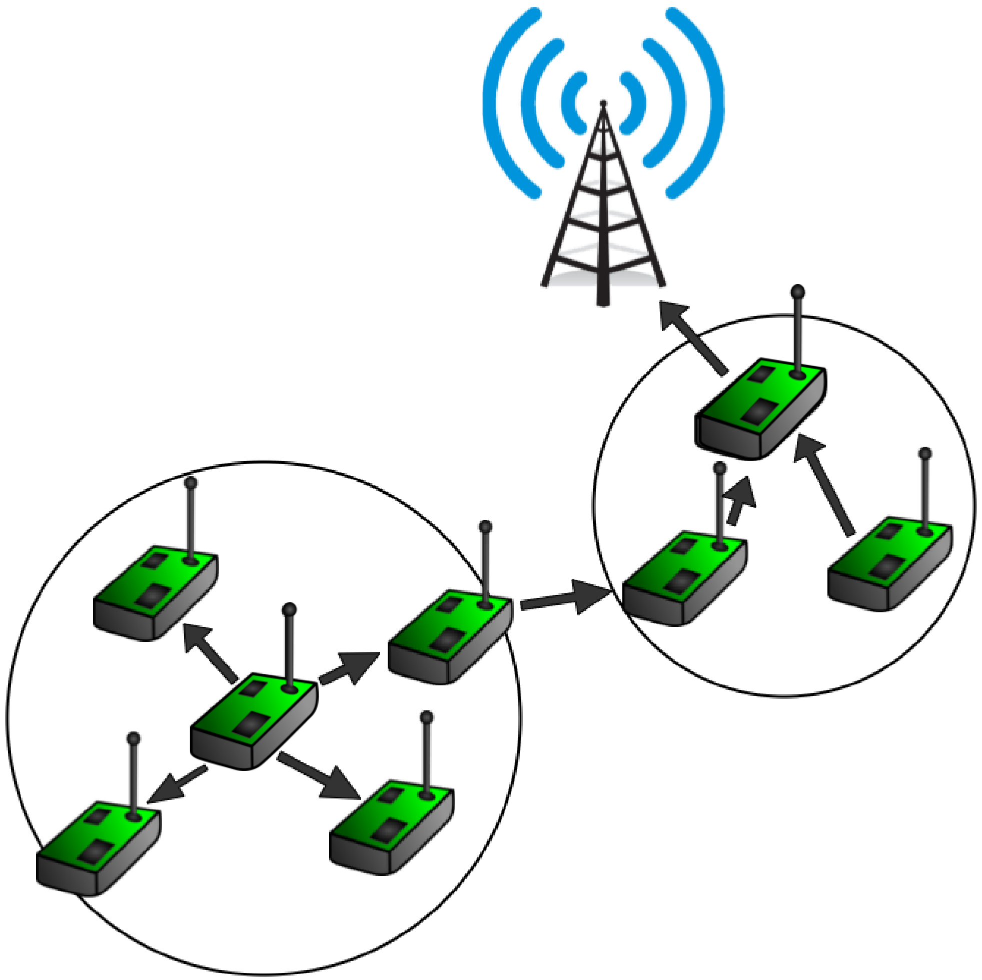
Challenge

- Wireless sensor networks
    - Fault prone
    - Dynamic topology
    - Energy efficiency



Challenge

- Wireless sensor networks
  - Fault prone
  - Dynamic topology
  - Energy efficiency
- How do we deal with selfish sensors if they prefer to not be cluster-heads?



Outline

- Self-stabilization
- Problem Definition
- Approaches
- Case study

Self-stabilization

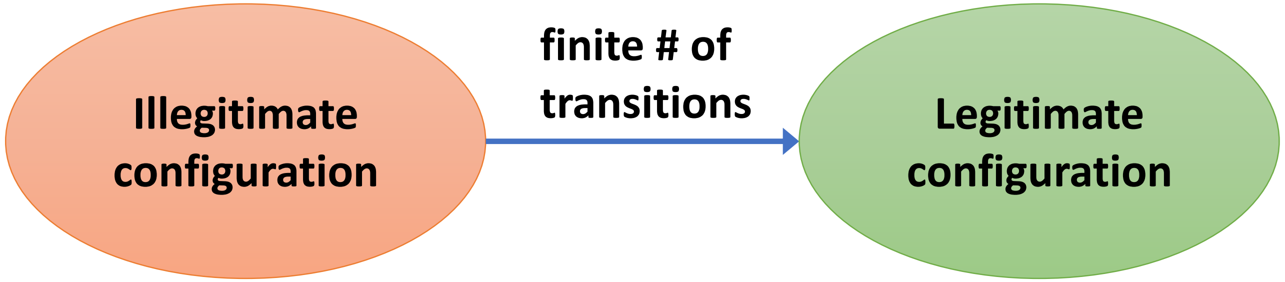
- A fault-Tolerance approach for distributed systems
- Definition
  - Self-stabilizing rules: $\langle guard \rangle \rightarrow \langle action \rangle$

Self-stabilization

- A fault-Tolerance approach for distributed systems
- Definition
  - Self-stabilizing rules: $\langle guard \rangle \rightarrow \langle action \rangle$
  - Properties

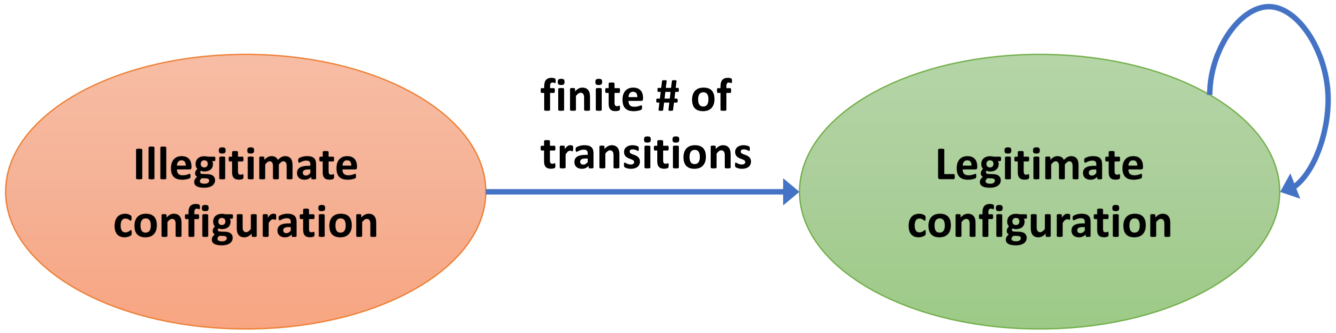
Self-stabilization

- A fault-Tolerance approach for distributed systems
- Definition
  - Self-stabilizing rules: $\langle guard \rangle \rightarrow \langle action \rangle$
  - Properties
    - **Convergence**



Self-stabilization

- A fault-Tolerance approach for distributed systems
- Definition
  - Self-stabilizing rules: $\langle guard \rangle \rightarrow \langle action \rangle$
  - Properties
    - **Convergence**
    - **Closure**



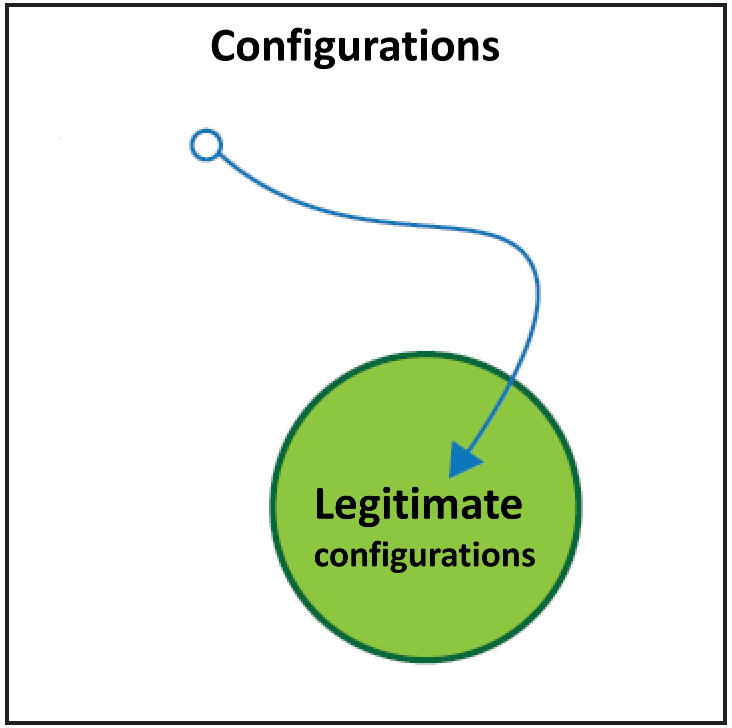
Problem Definition

- Selfish agents
  - Private goals
- Problem: how to deal with deviations from the desired behavior of a self-stabilizing algorithm?
- Related Works
  - “Selfish Stabilization”
  - “Nash equilibria in stabilizing systems”

Deviation Types

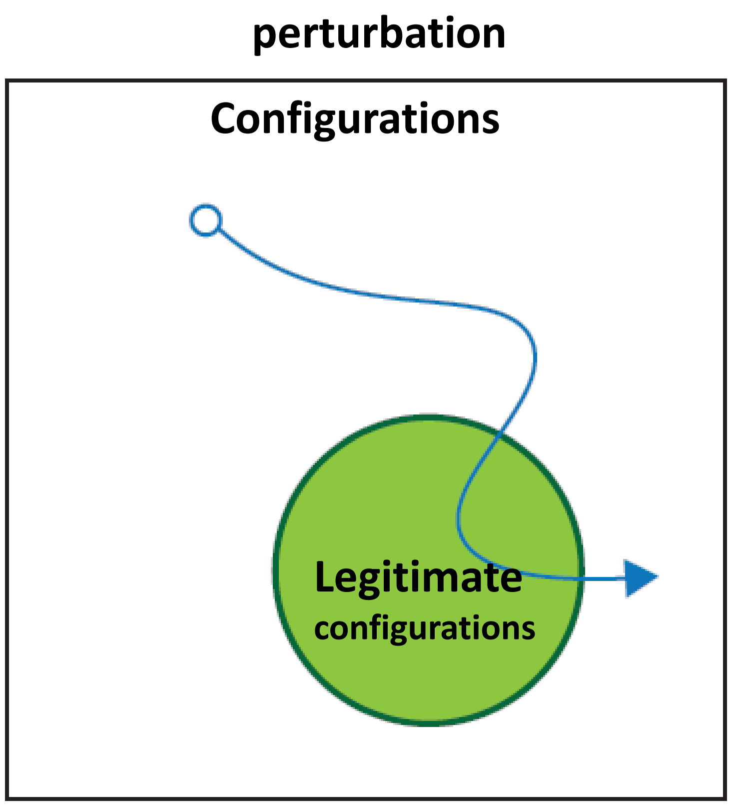
- No deviation

no deviation



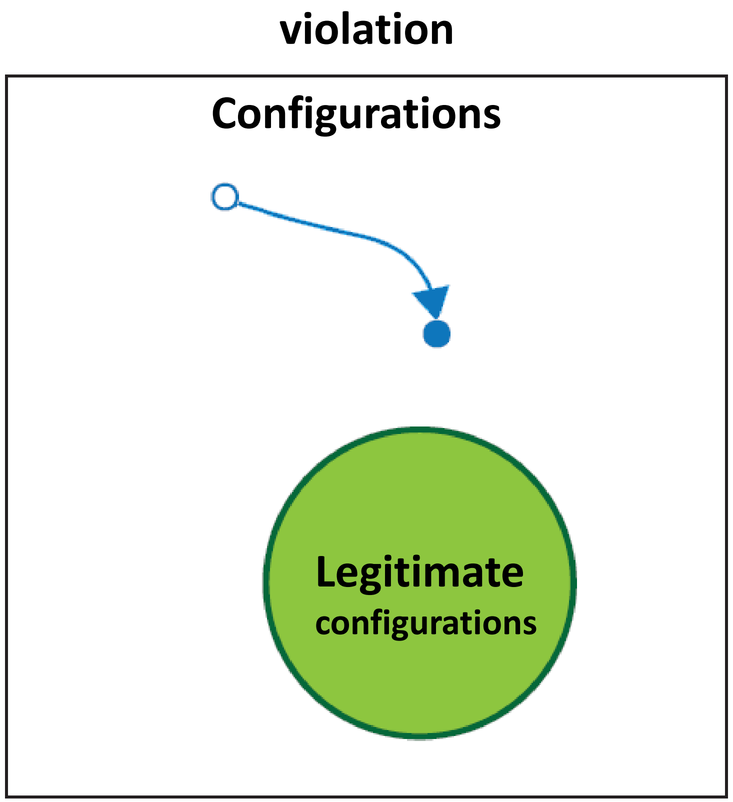
Deviation Types

- No deviation
- Perturbation



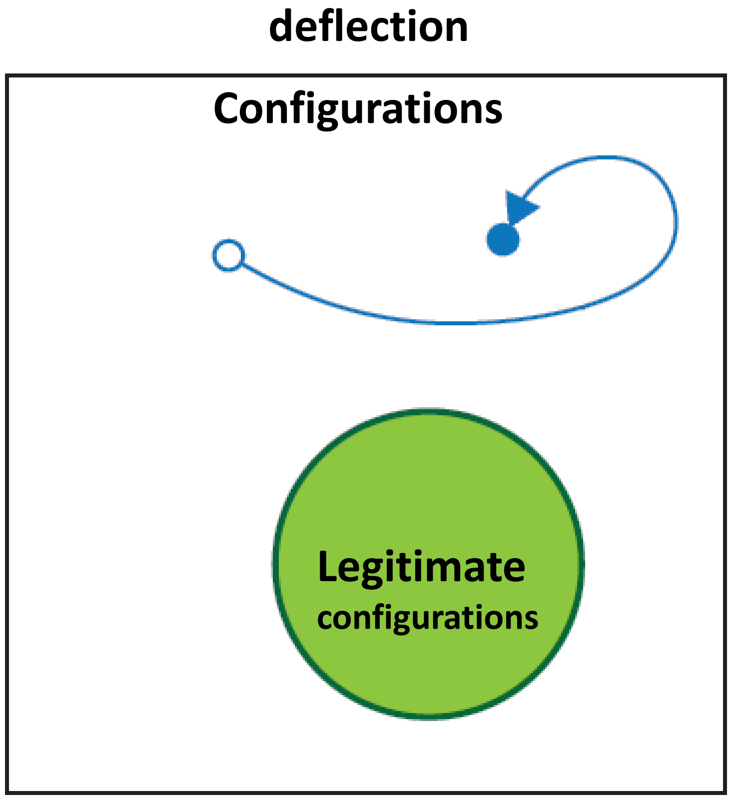
Deviation Types

- No deviation
- Perturbation
- Violation



Deviation Types

- No deviation
- Perturbation
- Violation
- Deflection



Goal

- **Question**: *how do we design a self-stabilizing algorithm for a distributed system given that agents may deviate from the algorithm during or after convergence because of their private goals?*

- ***We will focus on violation!***

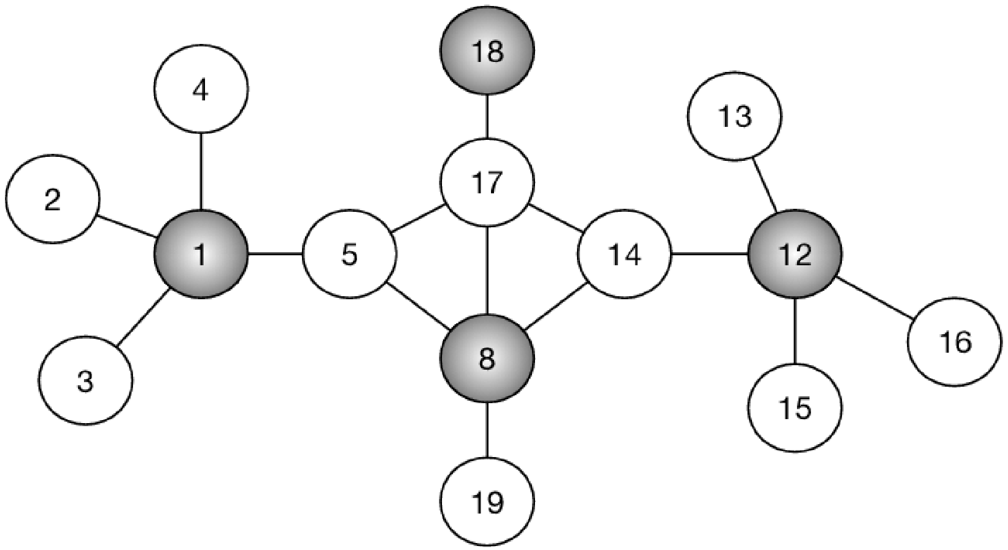
Violation-Tolerant Approach

- Probabilistic self-stabilization
- Bayesian stochastic games
  - Bayesian Perfect Equilibrium
- Randomized strategy ⟶ Probabilistic rules
- ***Theorem***:

Under the assumption that illegitimate configurations are not Nash equilibria, the convergence is guaranteed in the face of violations with randomizing the rules based on the equilibrium.

Case Study: Clustering

- Clustering
- Maximal Independent Set (MIS)



Case Study: Clustering

- Clustering
- Maximal Independent Set (MIS)

Basic self-stabilizing MIS algorithm (*b*MIS)

Rule 1: the agent **exits** the independent set if **it is a cluster-head and has a neighboring cluster-head**.

Rule 2: the agent **enters** the independent set **if it is not a cluster-head and has no neighboring cluster-head**.

Case Study: Clustering

- Clustering
- Maximal Independent Set (MIS)
- Utility function
  - A cluster member gets a reward r.
  - A cluster-head gets a reward r-c

Proposed Self-Stabilizing MIS Algorithms

- Violation-tolerant MIS (*vt*MIS)
  - entering MIS by a probability *p*
  - *p* → radomized strategy of action *state-transition*

Proposed Self-Stabilizing MIS Algorithms

- Violation-tolerant MIS *(vt*MIS*)*
  - entering MIS by a probability *p*
  - *p* → radomized strategy of action *state-transition*

Violation-tolerant self-stabilizing MIS algorithm (*vt*MIS)

Rule 1: the agent exits the independent set if it is a cluster-head and has a neighboring cluster-head.

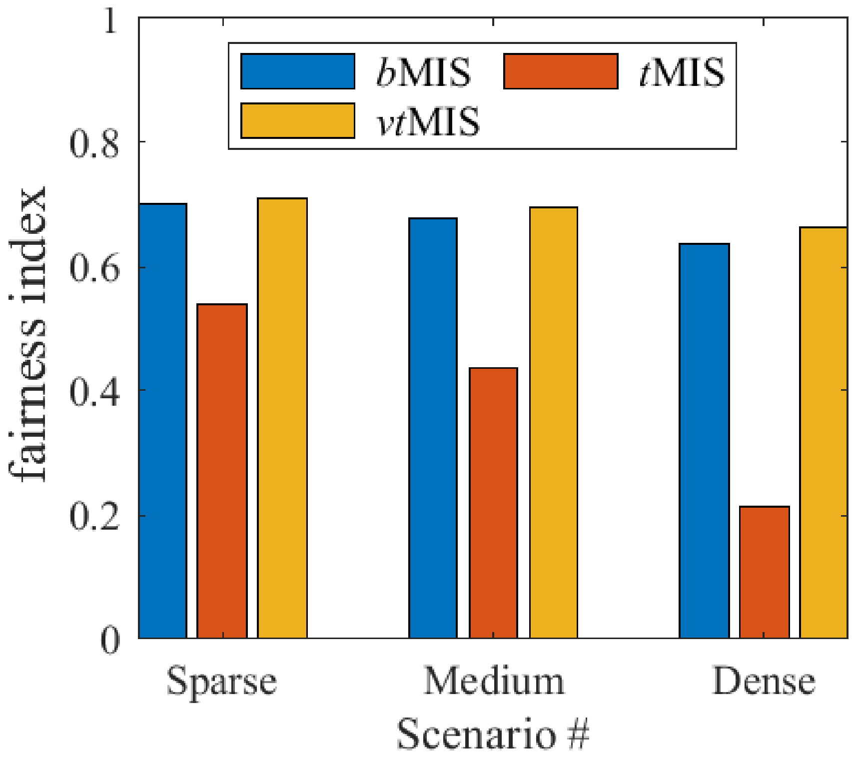
Rule 2: the agent enters the independent set **with a probability *p*** if it is not a cluster-head and has no neighboring cluster-head.

Experiments

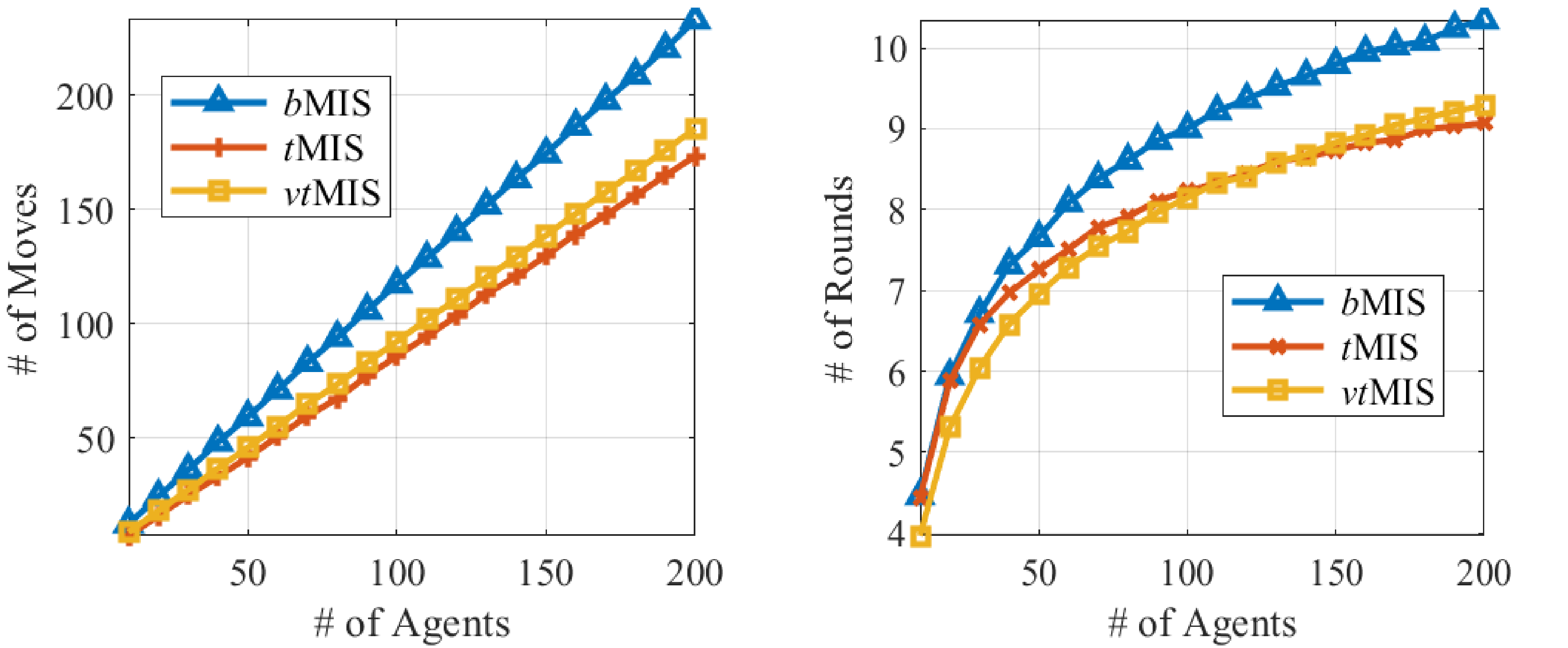
- Scale-free connected undirected graphs.
- Scenarios → different density networks.
- Random initial configurations.
- $r = 10$, and $c = 1$.
- State-of-the-art algorithm *t*MIS

Fairness

scenario	# of agents	avg. degree
Sparse	50	4
Medium	50	6
Dense	500	24



Time Complexity



Performance in the Face of Deviations

- We observe that
  - *vt*MIS is reliable to the onset of violations.
  - *t*MIS can sometimes protect against violations.
  - # of deviations is inversely proportional to reliability.
  - Deflections have a very negative effect on availability of clusters.

Conclusion

- We proposed three game-theoretic approaches for designing self-stabilizing algorithms in the face of selfish agents.
- We applied our solution methods to the problem of self-stabilizing clustering.
- The analysis of the results suggests that our solutions perform well concerning the fairness and ability to deal with selfishness.

Thanks for listening

Questions?