Vrije Universiteit Brussel

CoMo
Computational Modeling Lab

DECENTRALIZED COORDINATION IN MULTI-AGENT SYSTEMS

Mihail “Mike” Mihaylov

Aim?

to give the intuition just how complex decentralized coordination is

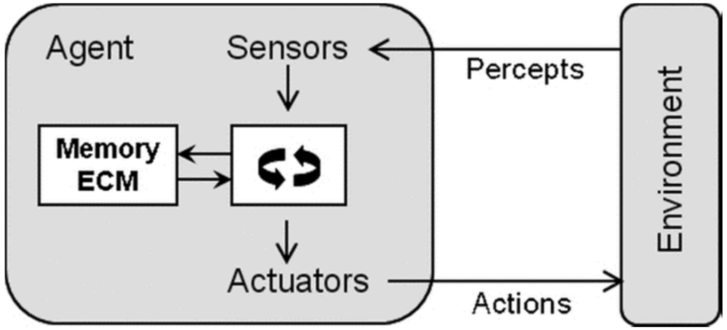
very complex!

to give the tools necessary to address decentralized coordination problems

Frameworks:

– Multi-agent systems
– Reinforcement Learning
– Collective Intelligence
– Mechanism Design

Agent



Autonomous

- has control over own actions
- able to act without human intervention

Pro-active

- takes initiative
- is opportunistic

Responsive

- perceives its environment
- responds to changes

Social

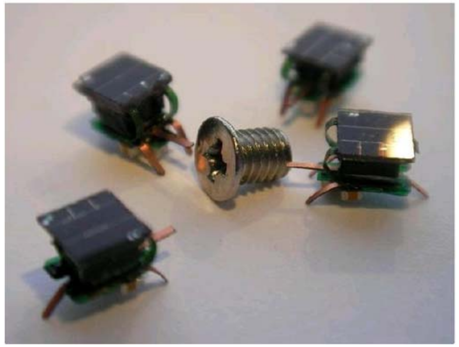
- interacts when appropriate
- helps others

"Objects do it for free.
Agents do it for money."

Multi-Agent Systems Framework

Agents have:

- incomplete information
- restricted capabilities



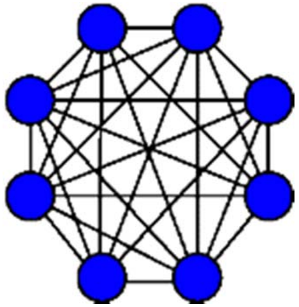
System properties:

- decentralized control
- **asynchronous** computation

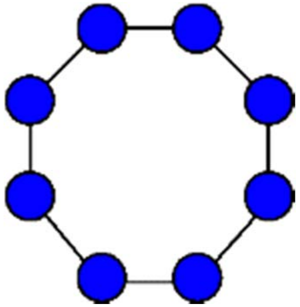
Communication is:

- costly
- delayed
- unreliable

[Wooldridge and Jennings, 1995]



Graphical games (Network games)



Normal form

- any agent interacts with any other agent
- payoffs depend on actions of all agents
- representation: exponential in the number of players

Graphical form

- only neighbors in the graph can interact
- payoffs depend on actions of neighbors
- representation: exponential in size of largest neighborhood

Context

Constraints:

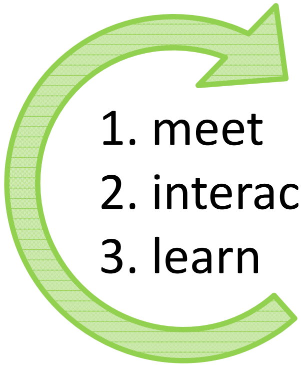
– limited battery

– communication is expensive

– low processing capabilities

– limited knowledge

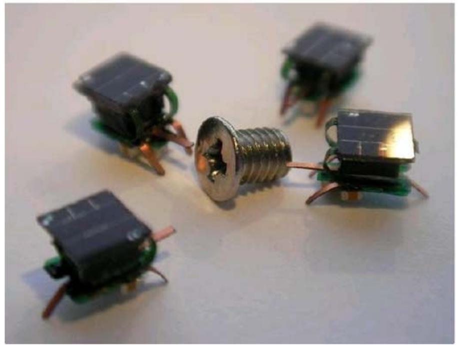
– decentralized control

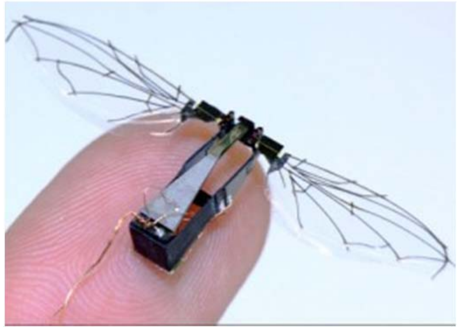


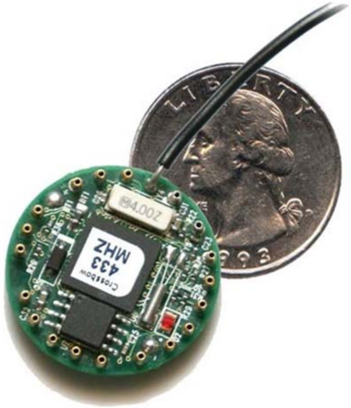
1. meet
2. interact
3. learn

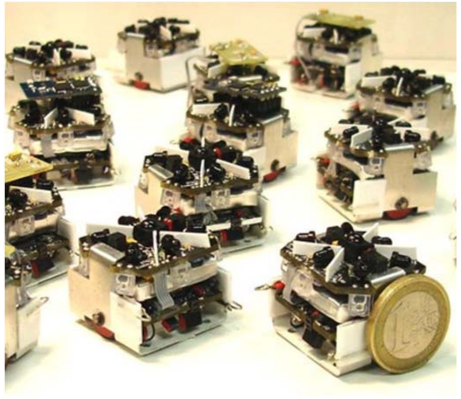
Reinforcement learning

Microcontr.: 8MHz
(as Intel processor
from year 1978)







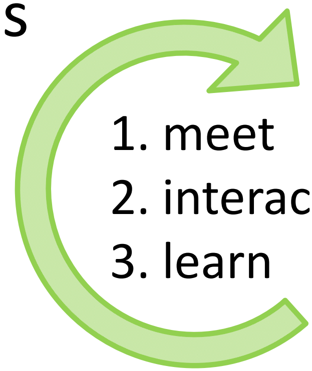


Decentralized

- central control is unavailable or costly to set up (*e.g. WSNs, Swarm robotics*)
- reduce complexity of centralized problems (*e.g. Scheduling, Planning*)
- address privacy, self-interest (*e.g. Smart grids, Transportation logistics*)

Coordination

- highly constrained agents
- with limited knowledge
- must work together to solve problems
- learn from repeated interactions



1. meet
2. interact
3. learn

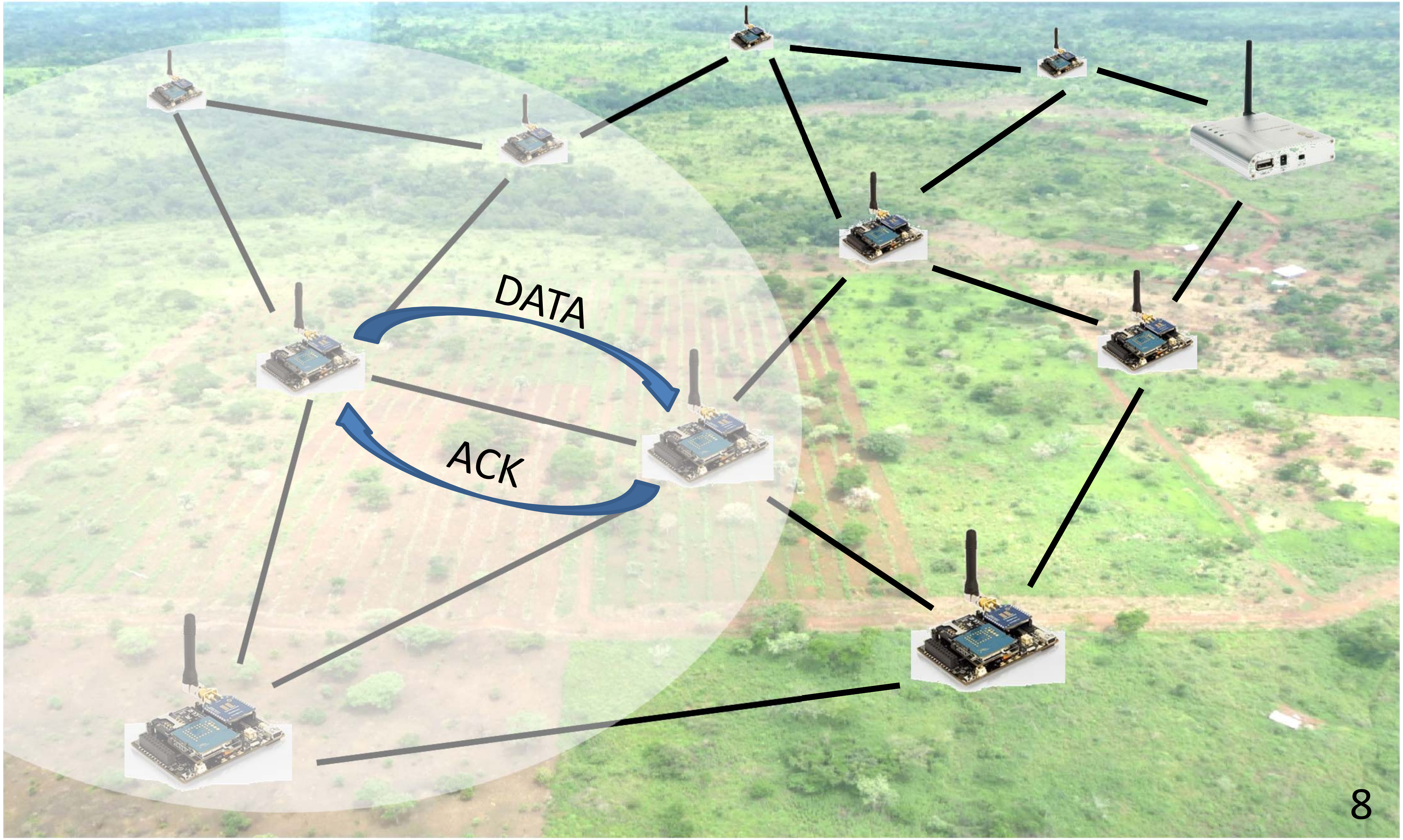
Agents

- inexpensive
- multi-purpose

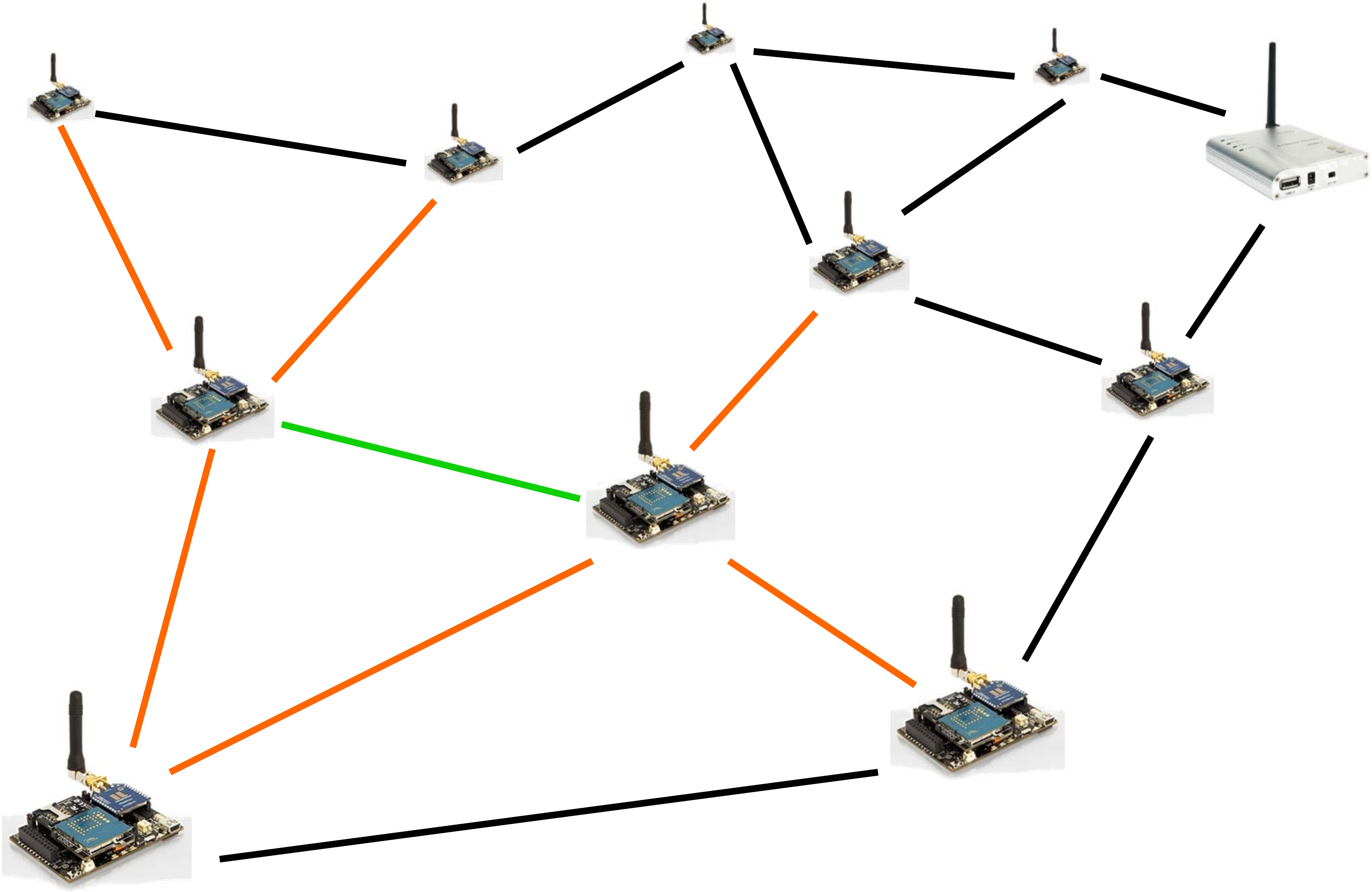
System

- scalable
- adaptive

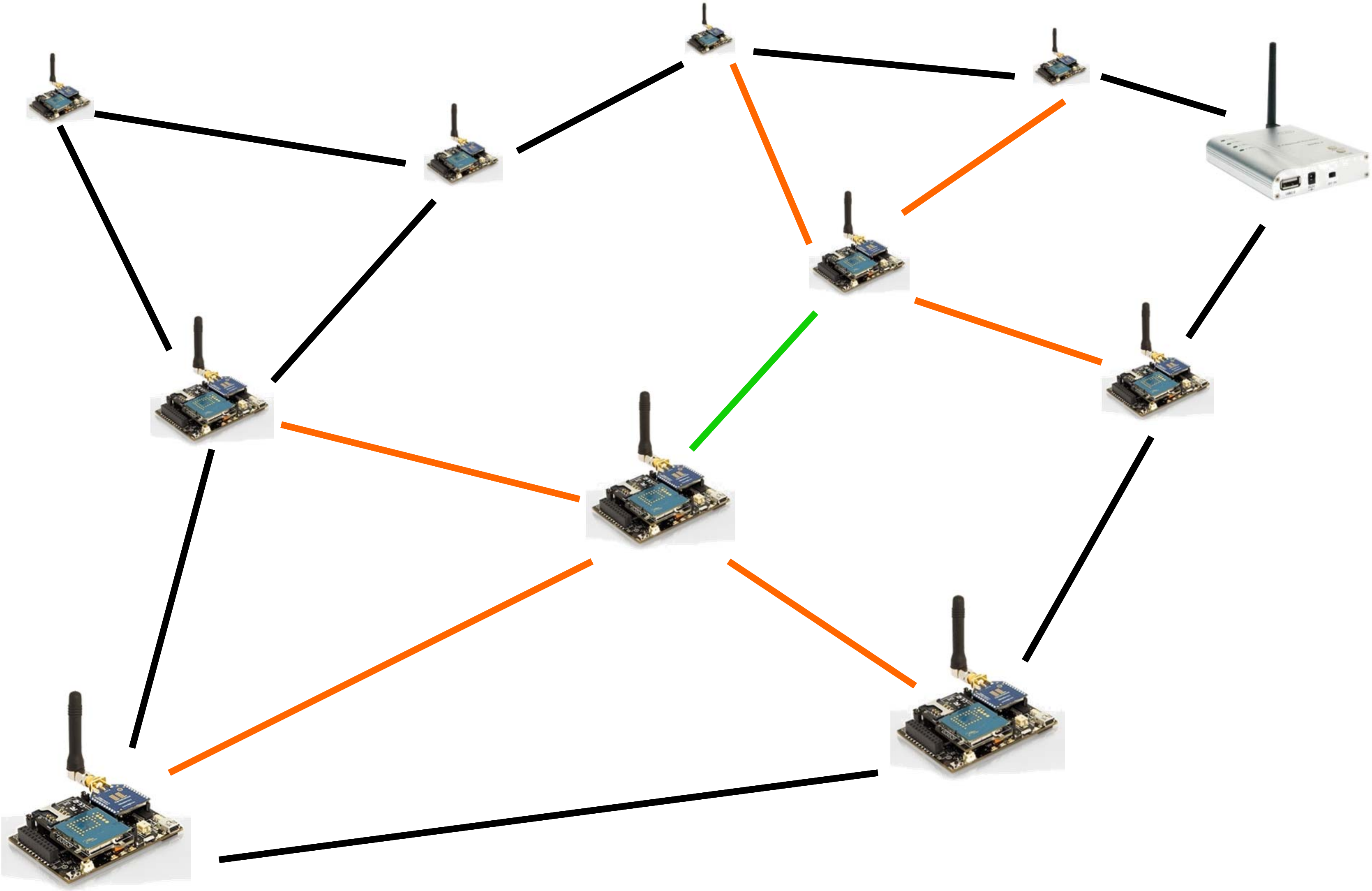
Decentralized Coordination in Multi-Agent Systems



Coordination and Anti-Coordination



Coordination and Anti-Coordination in time



Problem:

enable coordination & anti-coordination in time

- highly constrained agents → no complex algorithms
- local interactions → no centralized control
- limited knowledge → no global awareness
- autonomous → no human guidance

Objective:

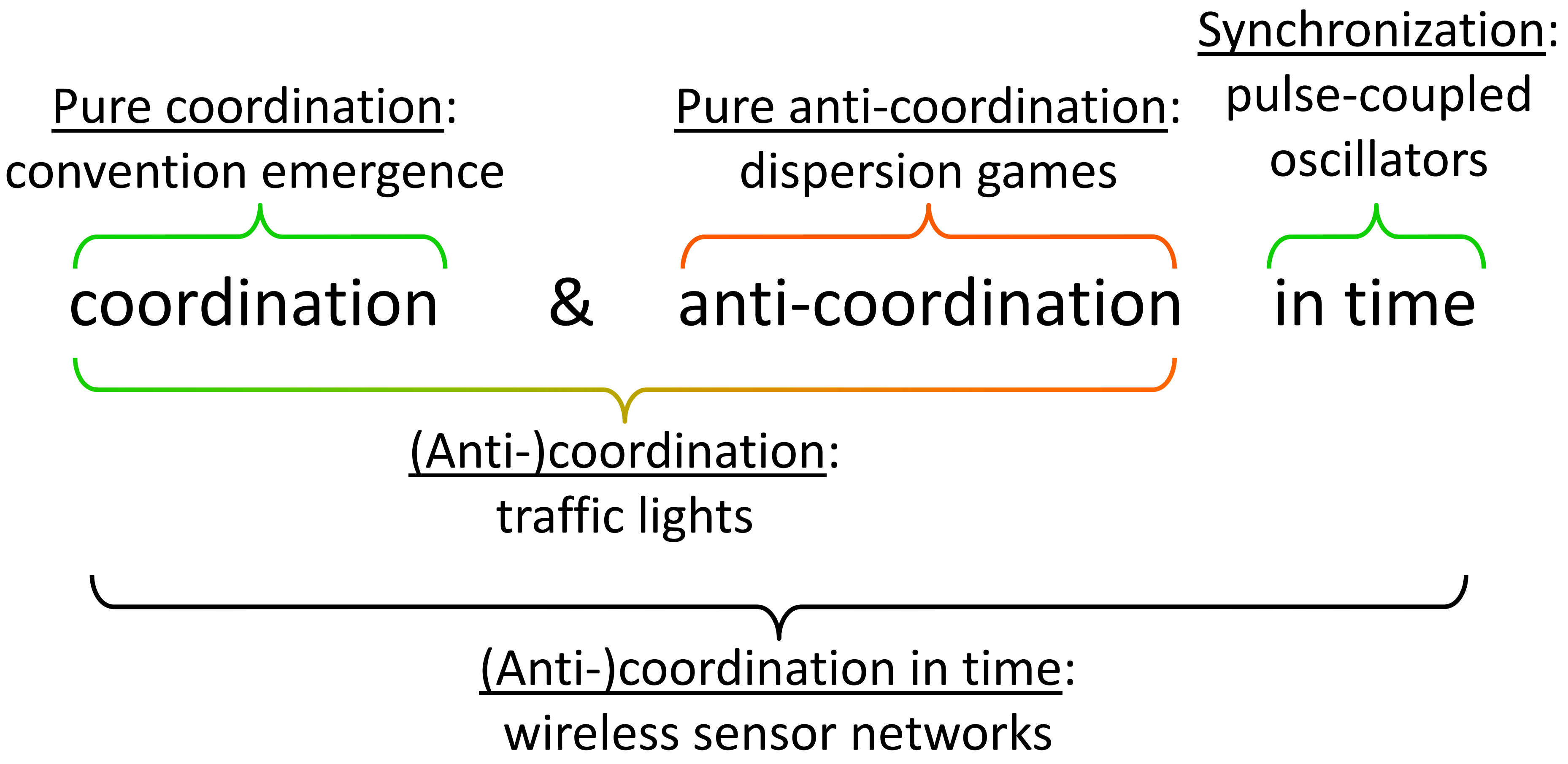
implement a coordination mechanism

- adaptive
- decentralized
- minimal requirements
- minimal overhead

→ perform well in wide range of settings

→ few parameters, little memory usage

Problem:



Pure coordination: convention emergence

Pure anti-coordination: dispersion games

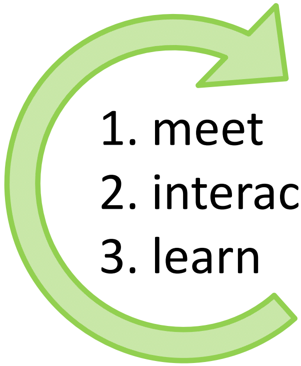
Synchronization: pulse-coupled oscillators

coordination & anti-coordination in time

(Anti-)coordination: traffic lights

(Anti-)coordination in time: wireless sensor networks

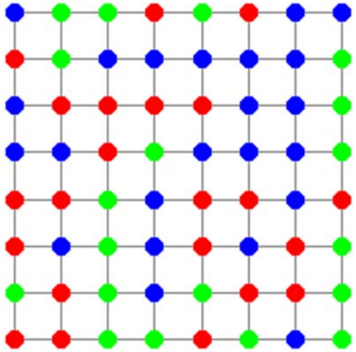
Pure coordination


1. meet
2. interact
3. learn

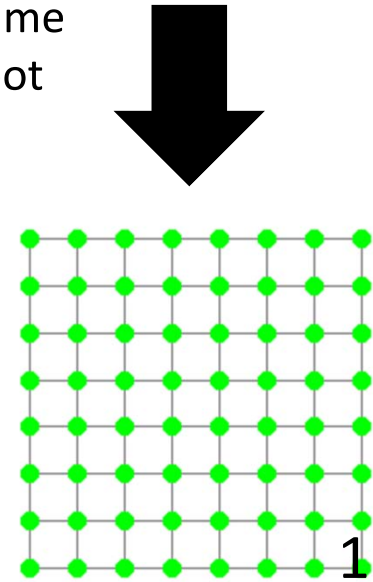
Problem: How to coordinate?

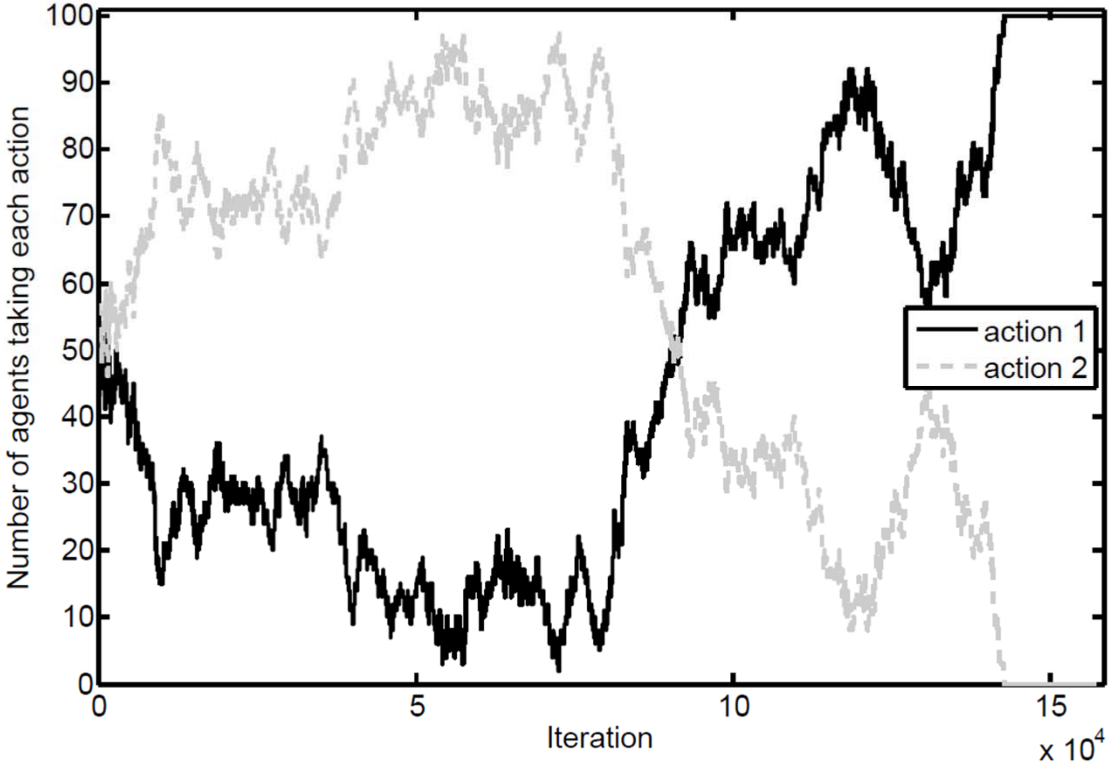
- b/n highly constrained agents
- via local interactions
- with limited knowledge

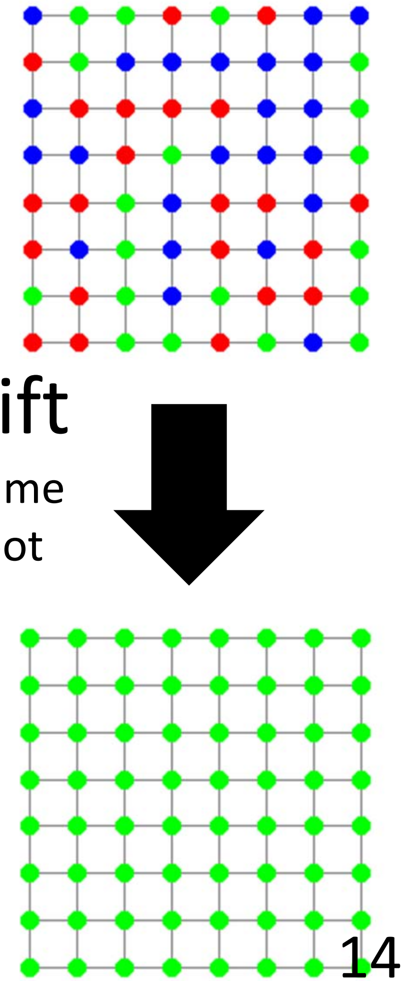
		Agent 1		
		🔴	🟢	🔵
Agent 2	🔴	1	0	0
	🟢	0	1	0
	🔵	0	0	1



e.g. select the same joint task in robot swarms







e.g. select the same joint task in robot swarms

Approach: Win-Stay Lose-probabilistic-Shift

Requirements:

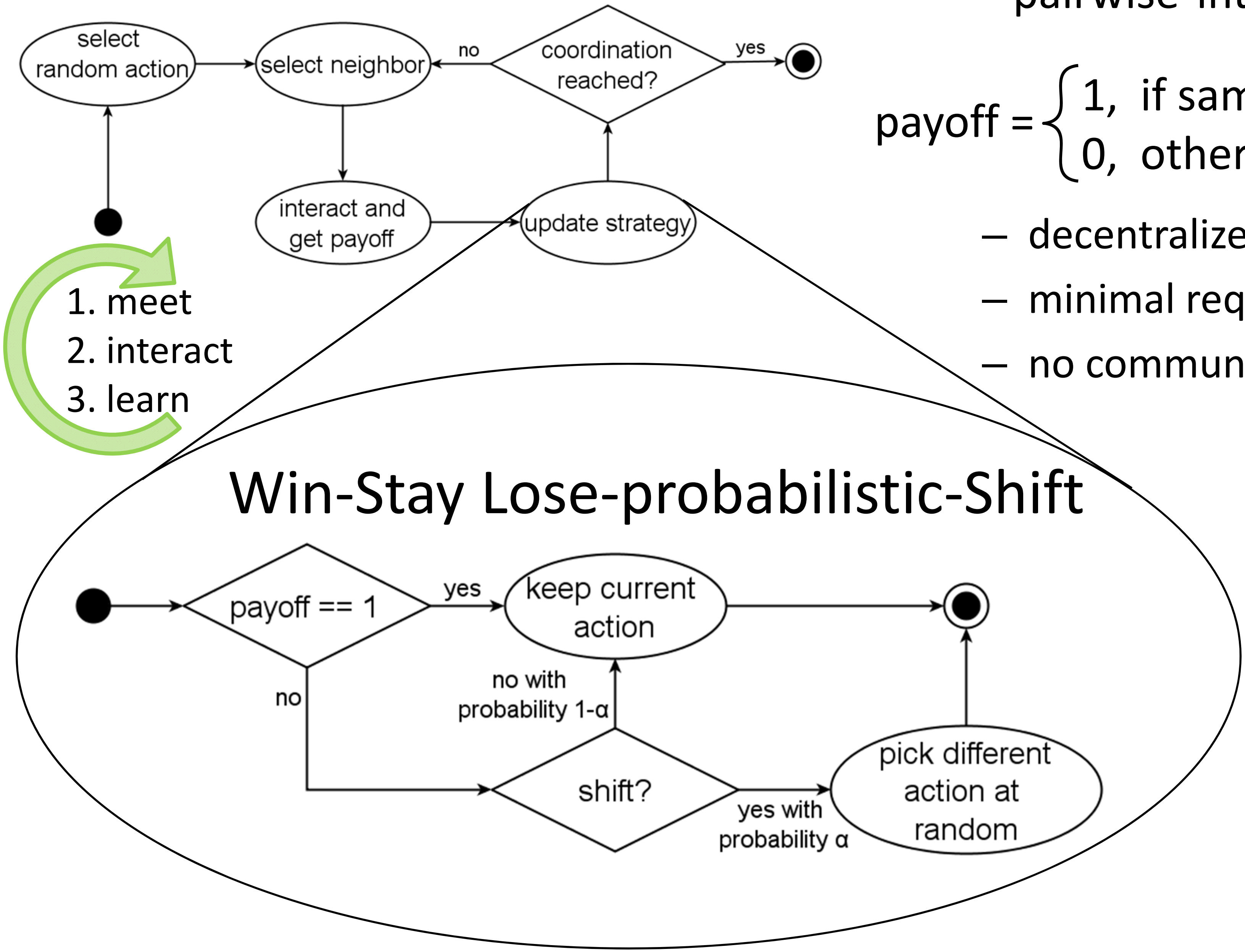
- decentralized mechanism
- minimal requirements & overhead
- guaranteed full convergence
- absorbing state

Coordination Game:

pairwise interactions

$$\text{payoff} = \begin{cases} 1, & \text{if same action} \\ 0, & \text{otherwise} \end{cases}$$

- decentralized approach
- minimal requirements
- no communication

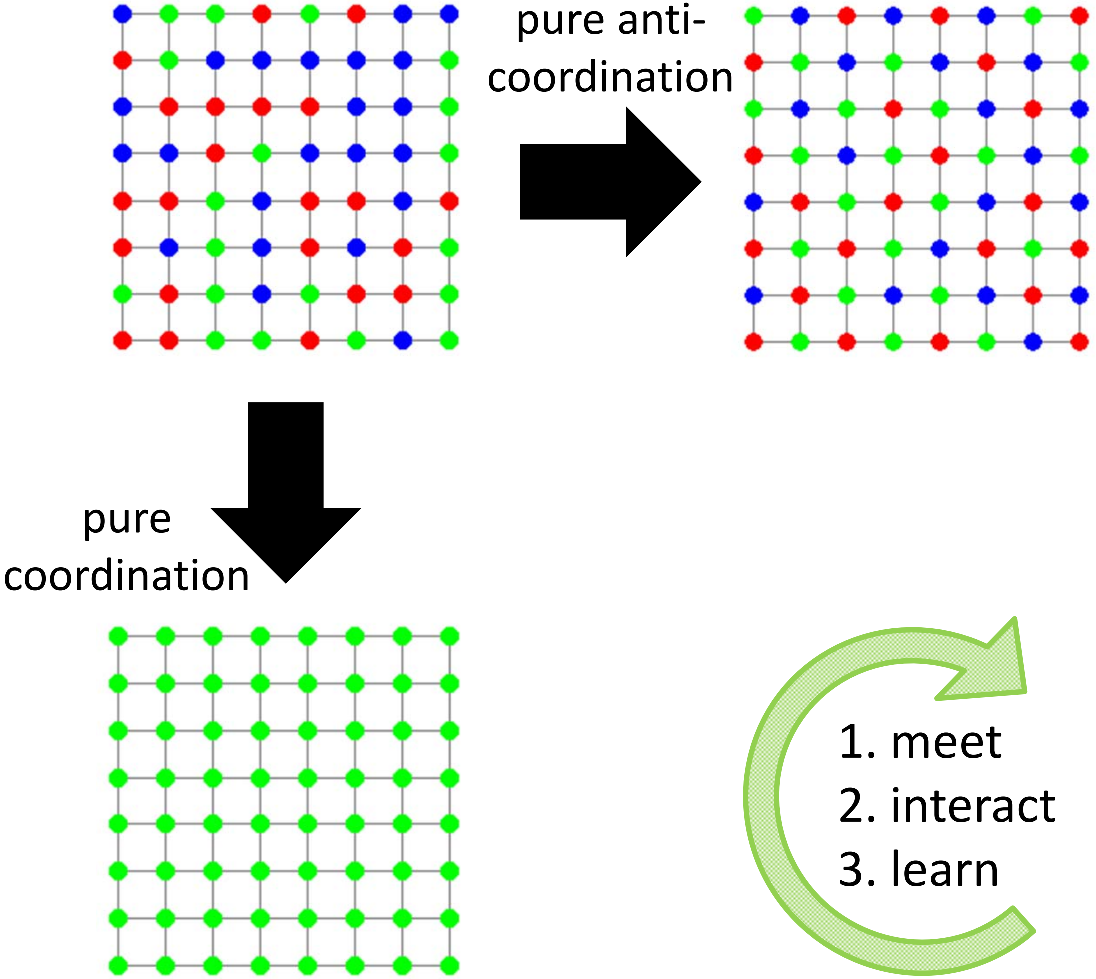


1. meet
2. interact
3. learn

Win-Stay Lose-probabilistic-Shift

Conclusions (pure coordination)

- Pure coordination → difficult, but always possible
- Convergence time → exponential in number of agents and actions
- Denser networks → faster convergence



pure anti-coordination

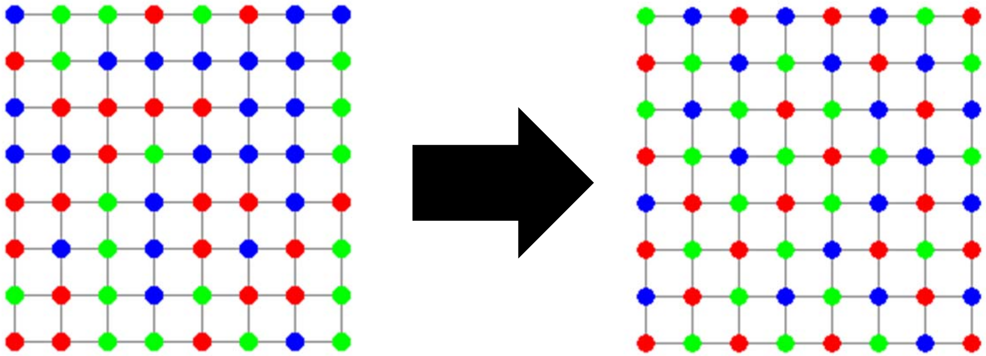
pure coordination

1. meet
2. interact
3. learn

Anti-coordination

Problem: How to anti-coordinate?

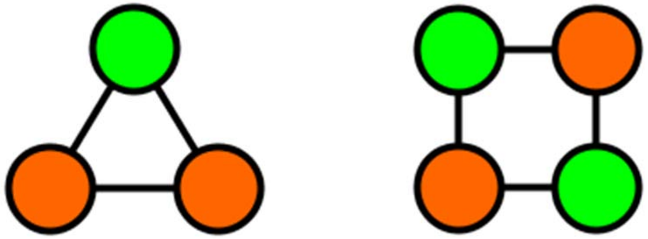
(select an action different than others')



e.g. select different channels for parallel communication in WSNs

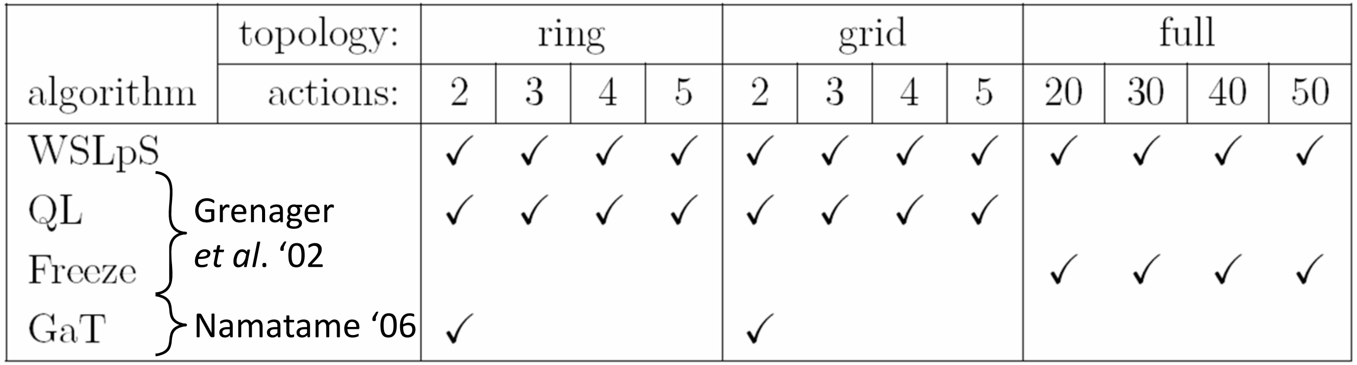
Conclusions (anti-coordination)

- Pure anti-coordination → easy but not always feasible

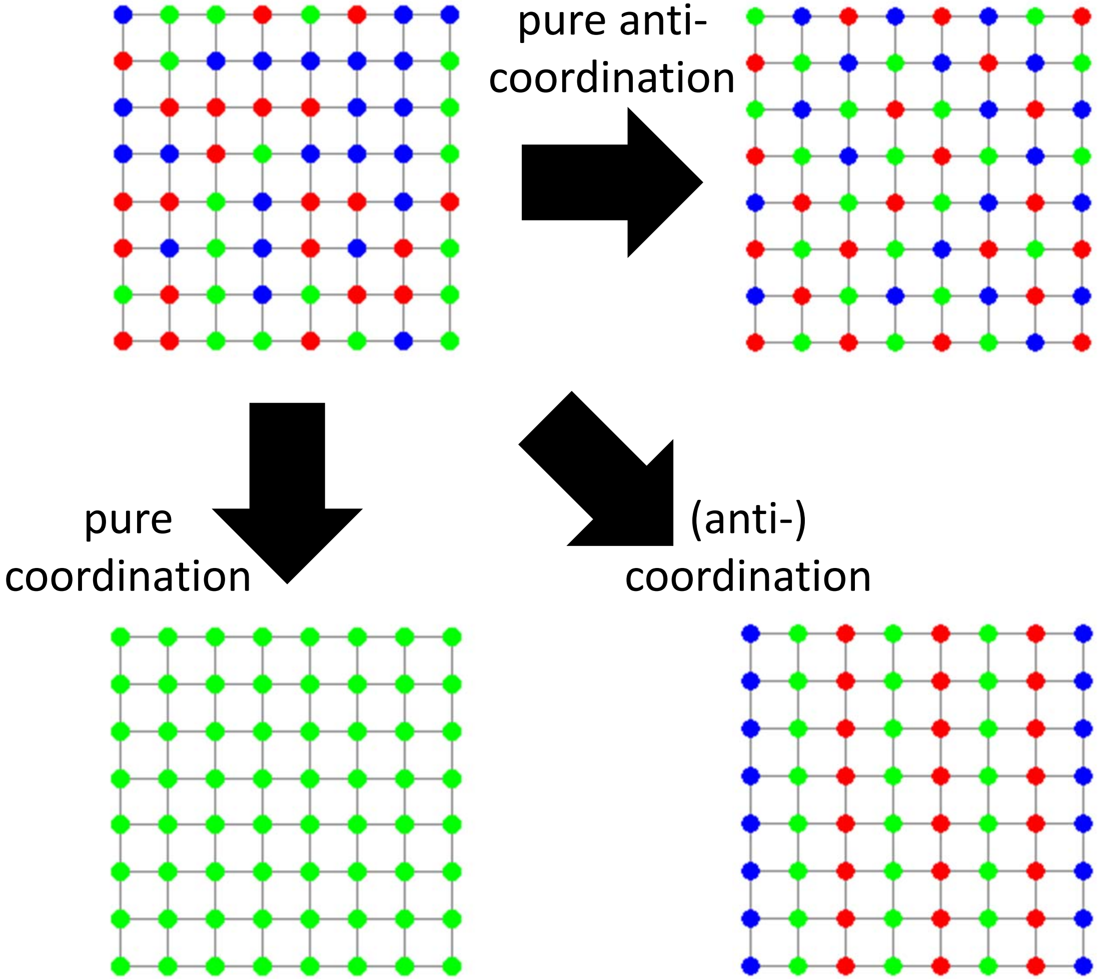


- Convergence time → faster with more actions

- WSLpS: applicable in wide range of scenarios



	topology:	ring				grid				full			
algorithm	actions:	2	3	4	5	2	3	4	5	20	30	40	50
WSLpS		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
QL	Grenager *et al.* '02	✓	✓	✓	✓	✓	✓	✓	✓				
Freeze										✓	✓	✓	✓
GaT	Namatame '06	✓				✓							

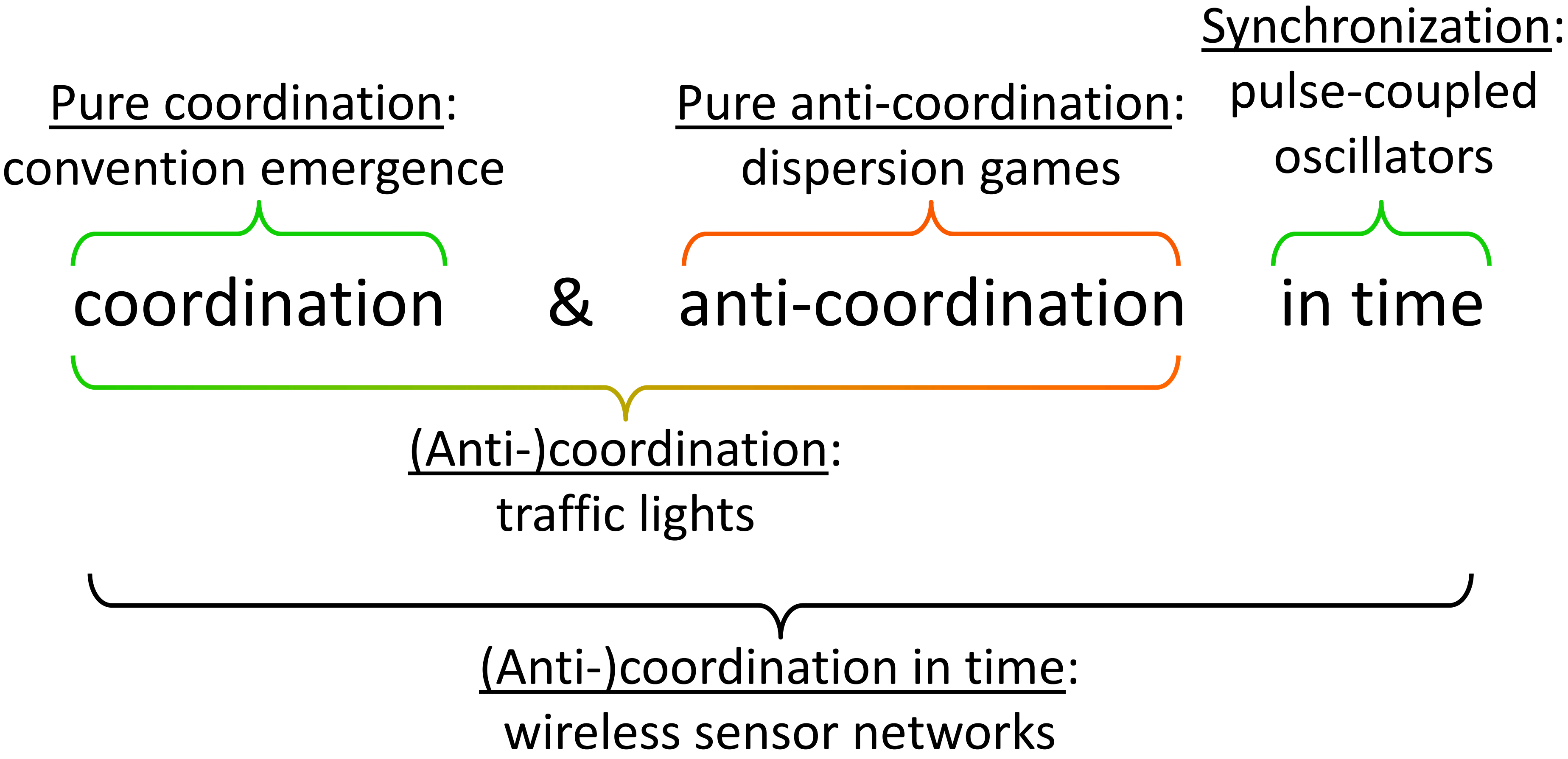


pure anti-coordination

pure coordination

(anti-) coordination

e.g. traffic lights in a city or routing branches in WSNs



Pure coordination:
convention emergence

Pure anti-coordination:
dispersion games

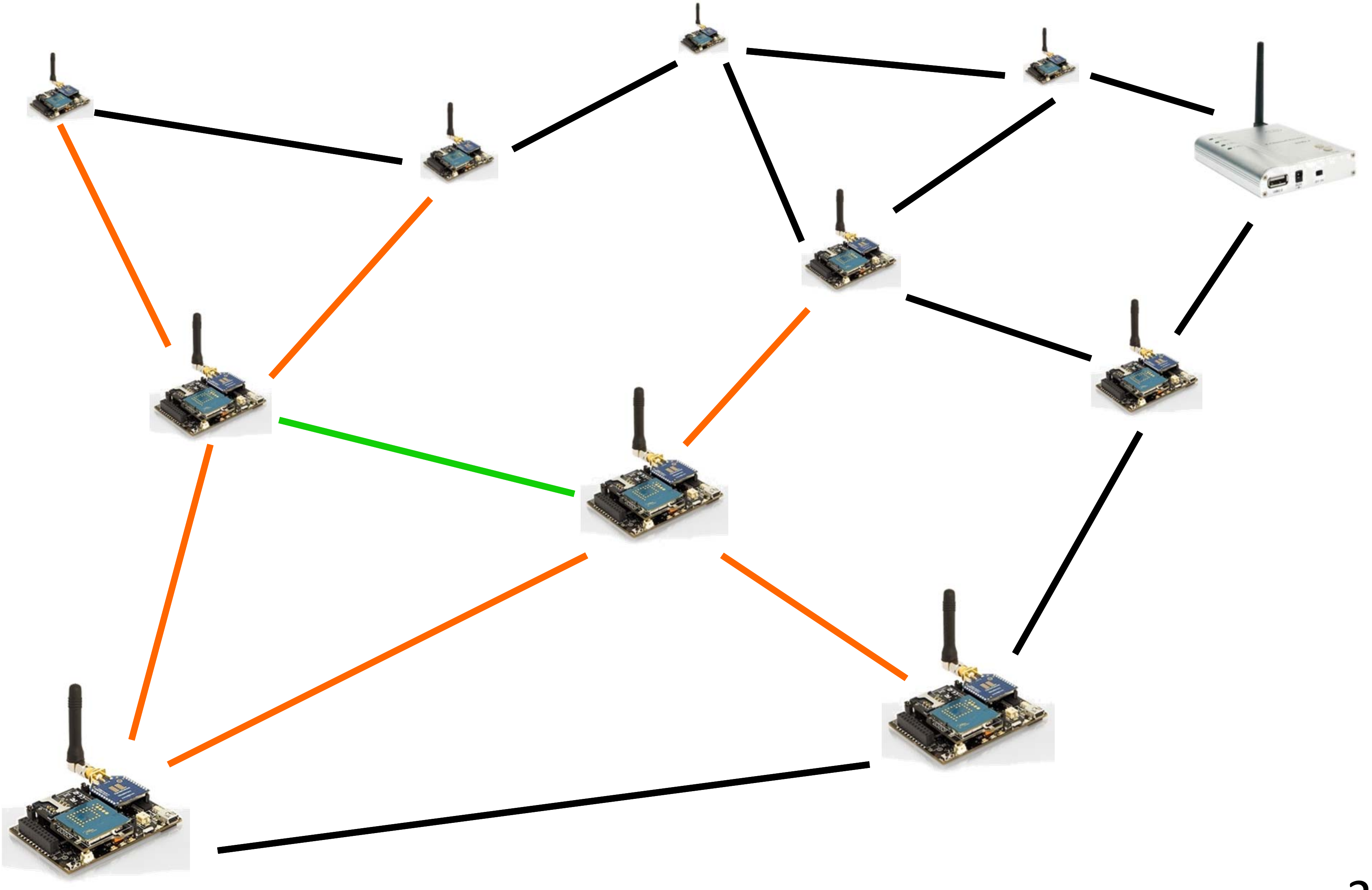
Synchronization:
pulse-coupled
oscillators

coordination & anti-coordination in time

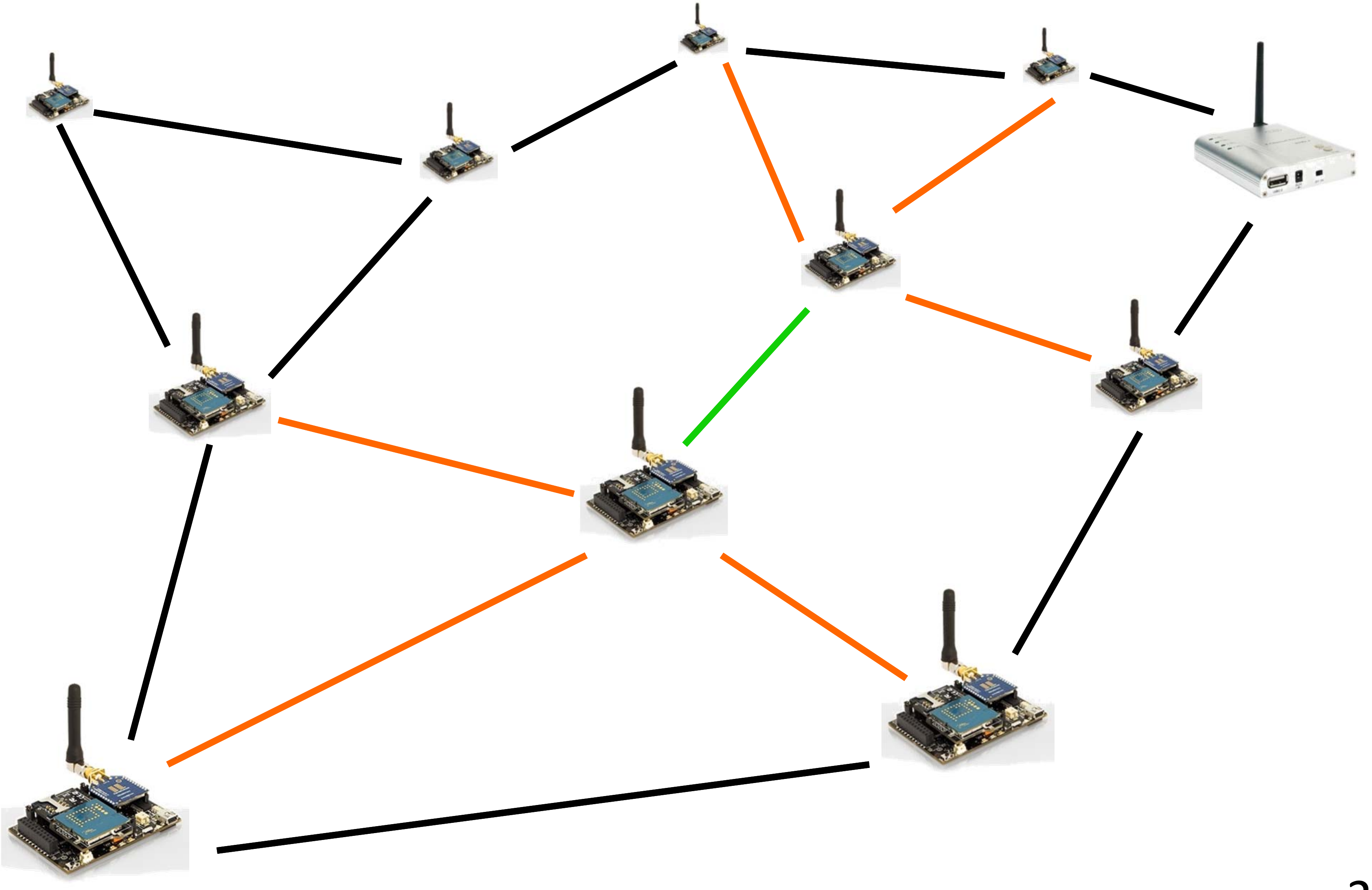
(Anti-)coordination:
traffic lights

(Anti-)coordination in time:
wireless sensor networks

Coordination and Anti-Coordination



Coordination and Anti-Coordination in time



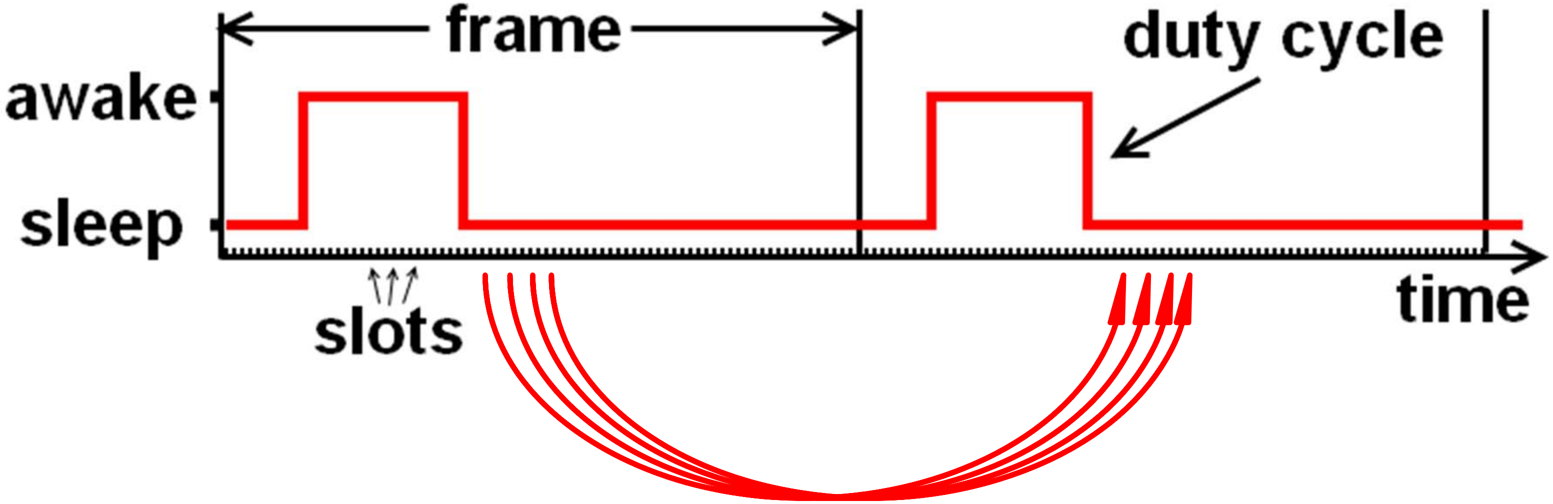
(Anti-)coordination in time: WSNs

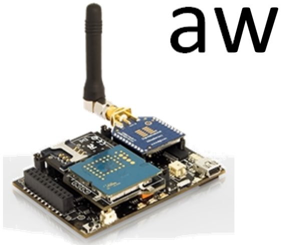
Problem: How to (anti-)coordinate in time?

Challenges in WSNs:

- no central control → decentralized comm. protocol
- only local information → learn by local interactions
- expensive communication → implicit coordination
- communication interference → anti-coordination
- limited memory & processing → simple algorithm
- no observation of actions of others → own actions

(Anti-)coordination in time: WSNs





awake
- Transmit
  - + Forward messages
  - – Interfere with neighbors
- Listen
  - + Receive messages
  - – Waste energy

sleep
- Sleep
  - + Save energy
  - – Increase latency

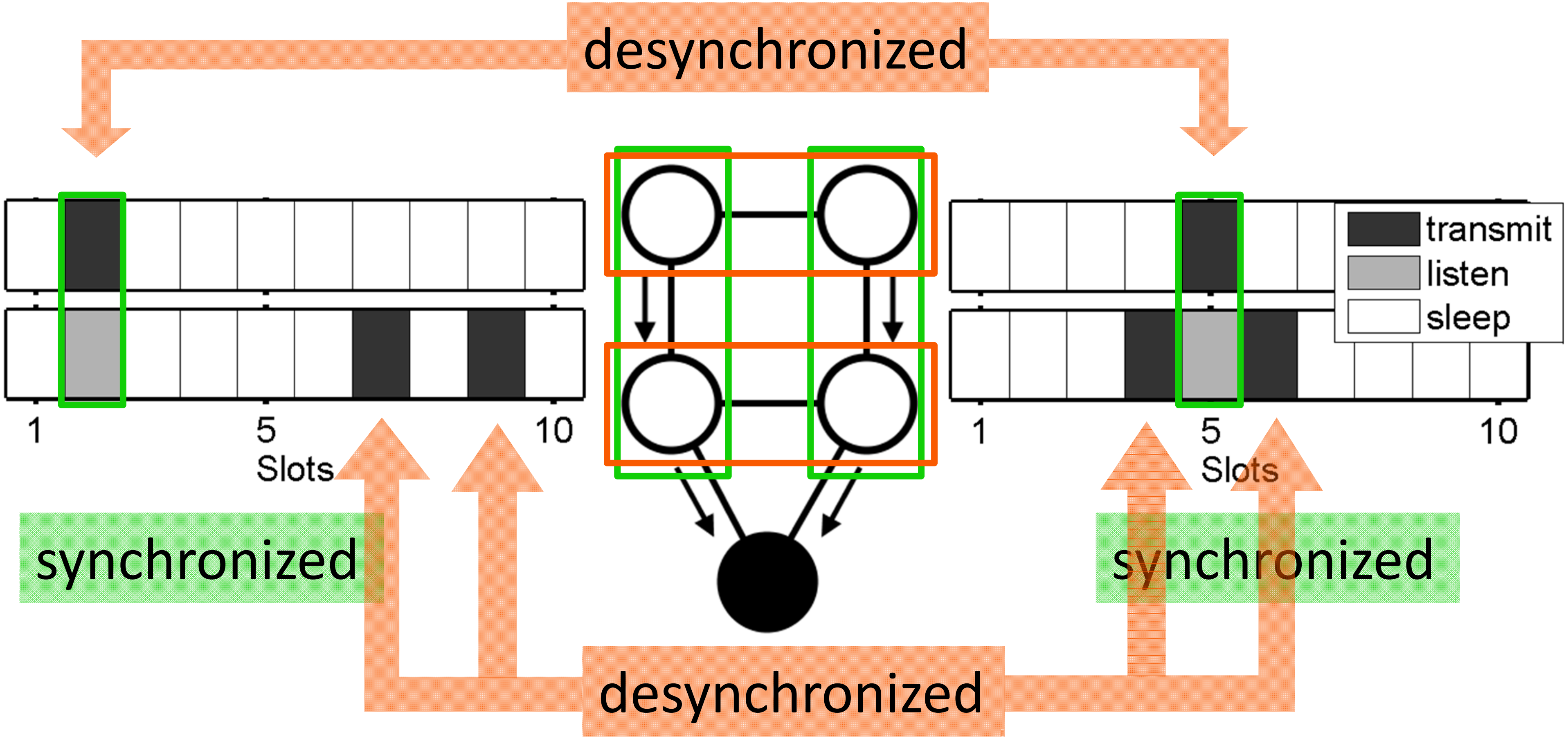
(Anti-)coordination in time: WSNs

Agents learn by only observing outcome of own actions!

action	outcome	payoff
Transmit	ACK received no ACK received	1 0
Listen	DATA received communication overheard nothing received messages collided	1 0 0 0
Sleep	saved energy	?

DESYDE

– decentralized approach
– minimal requirements
– no communication overhead



desynchronized

synchronized

synchronized

desynchronized

Conclusions (WSNs)

- Coordination emerges rather than is agreed upon

- Clever coordination mechanism is required

Outlook

Covered topics	Other topics	Framework
Common interest game	Conflicting interest game	Cooperative game theory
Fully cooperative agents	Self-interested agents	Mechanism design
Static topology	Dynamic topology	Evolutionary game theory
Wireless sensor networks	Collaborative platforms	Collective intelligence

Collective Intelligence Framework

Components:

- private utility → measures performance of individual agents
- world utility → measures performance of the entire system

Objective:

- align private utility with world utility

e.g.: optimizing the routing of internet traffic

Challenges:

- compute world utility in a decentralized manner

[Wolpert and Tumer, 2003]

Mechanism Design Framework

Agents:

- are self-interested
- have private information
- maximize utility functions

Objective:

- implement a protocol that achieves designer's goals, despite agents' self interest

e.g.: designing an electronic auction

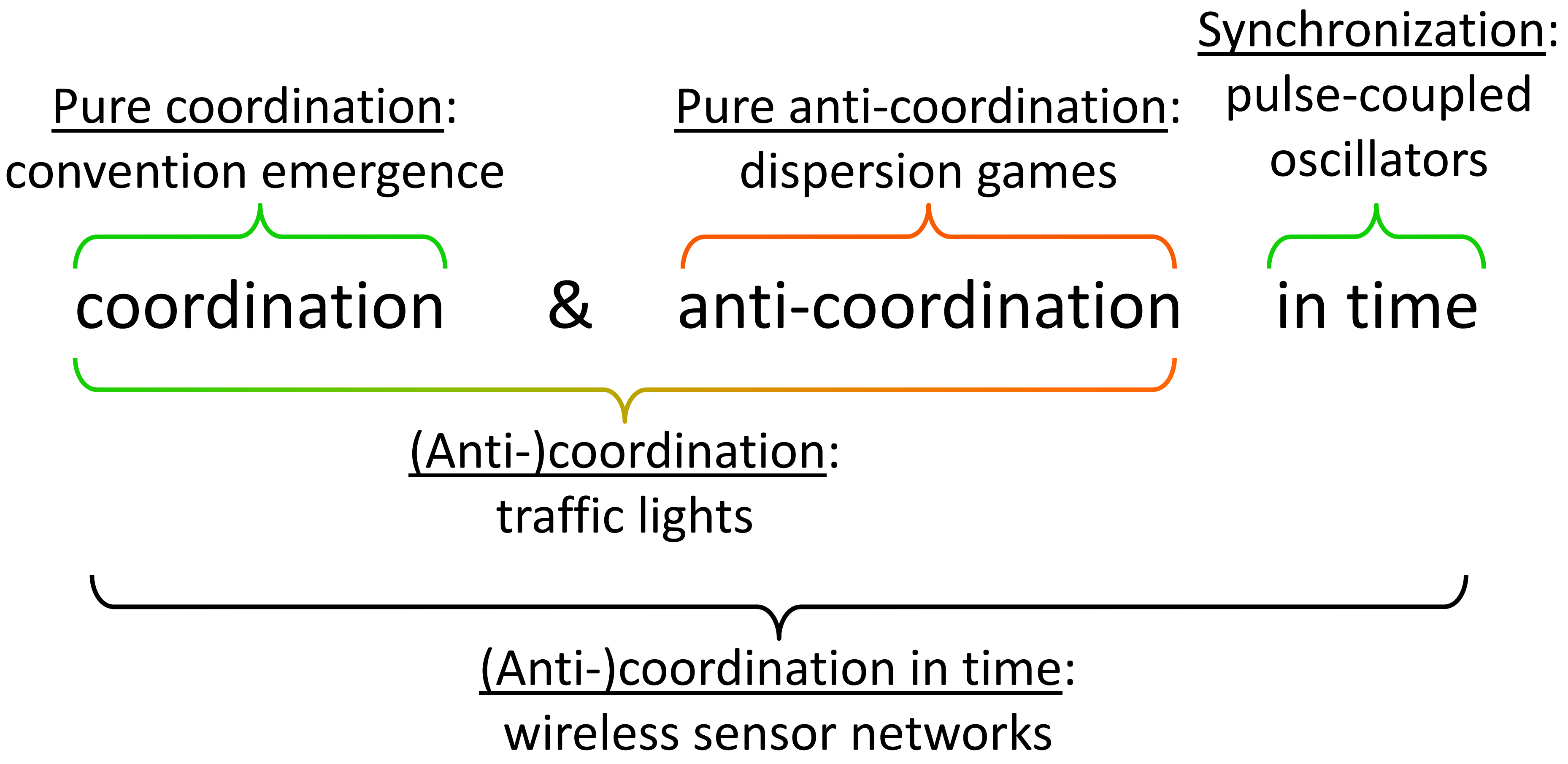
Game theory:

– *"Given a game, what is a rational strategy for an agent?"*

Mechanism design:

– *"Given that agents are rational, how should we design the game?"*

[Conitzer and Sandholm, 2003]



Pure coordination:
convention emergence

Pure anti-coordination:
dispersion games

Synchronization:
pulse-coupled
oscillators

coordination & anti-coordination in time

(Anti-)coordination:
traffic lights

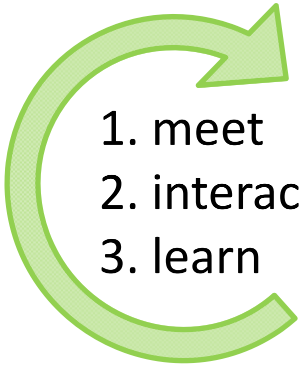
(Anti-)coordination in time:
wireless sensor networks

Synchronization

Desynchronization

Firefly
(*Photuris lucicrescens*)

Japanese tree frog
(*Hyla japonica*)

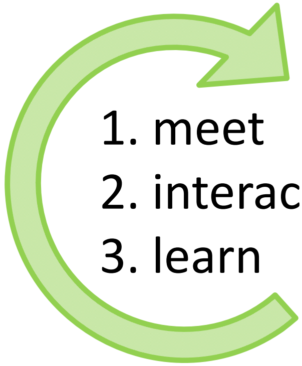


1. meet
2. interact
3. learn

Agents are able to observe each other’s actions!

Summary

Complex problems	Simple solution: WSLpS
– no central control	– decentralized approach
– explicit coordination is costly	– minimal overhead
– restricted capabilities of agents	– minimal requirements
– unknown topology	– wide applicability
– expensive learning	– fast convergence



1. meet
2. interact
3. learn

WSLpS: good starting point to address decentralized coordination problems

Conclusions

Simple learning techniques work surprisingly well.

+ less parameters to tune
+ quite generic, wider application
– lower representational power

Decentralized solutions are a powerful paradigm.

+ lower computational complexity & communication overhead
+ no single-point-of-failure problems
– difficult to organize

Multi-agent systems is a quite useful framework.

+ suitable representation of decentralized problems
+ scalable, fault-tolerant
– easy to overdo