A Clockless Synchronisation Framework for Cooperating Mobile Robots

Pitt

CISTER - Research Centre in
Real-Time & Embedded Computing Systems

instituto de
telecomunicações

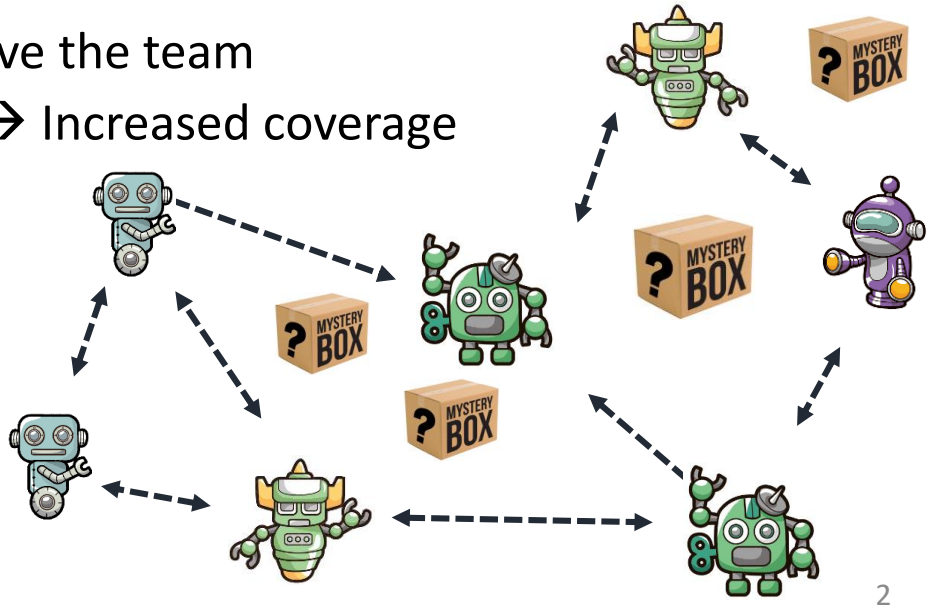
U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Luís Oliveira, Luís Almeida, Daniel Mossé

This work was partially supported by the Portuguese Government through FCT grant UID/EEA/50008/2013 - Instituto de Telecomunicações.

Motivation

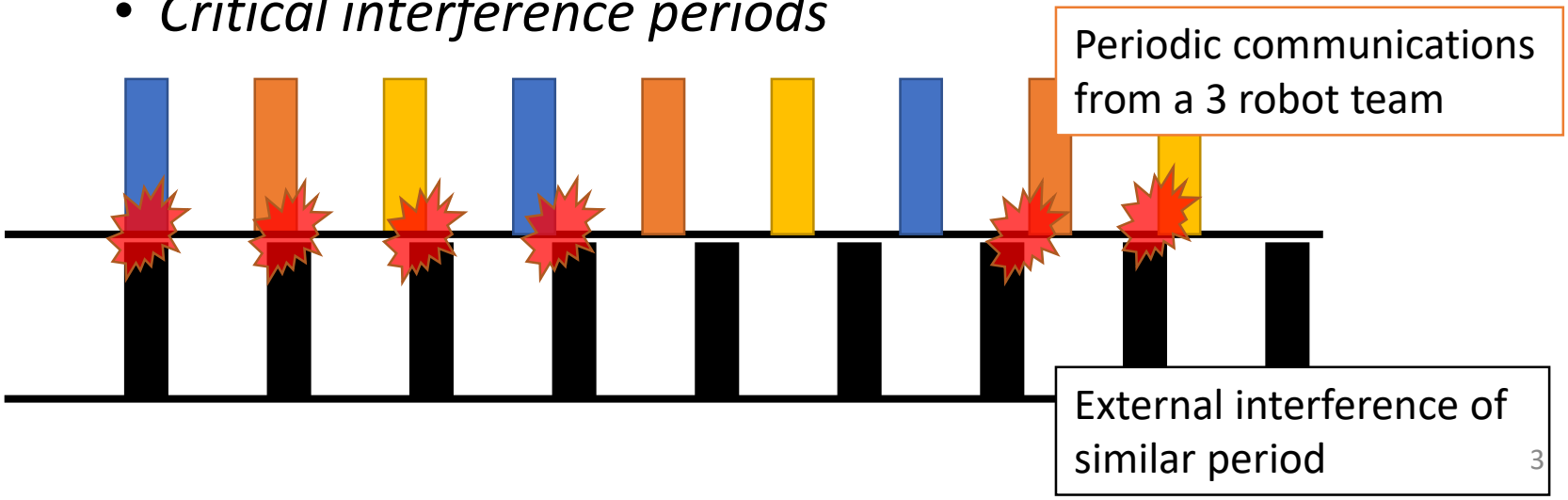
- Widespread use of small teams of cooperative mobile robots
  - Need to communicate using a wireless medium
    - Improved quality of communications → Improved cooperation
  - Are dynamic
    - Robots may join or leave the team
    - Topology can change → Increased coverage



Motivation

In order to improve wireless communications

- Synchronising transmissions is vital
  - Support periodic communications
- Global clock synchronisation
  - Does not cope with periodic external interference
  - *Critical interference periods*



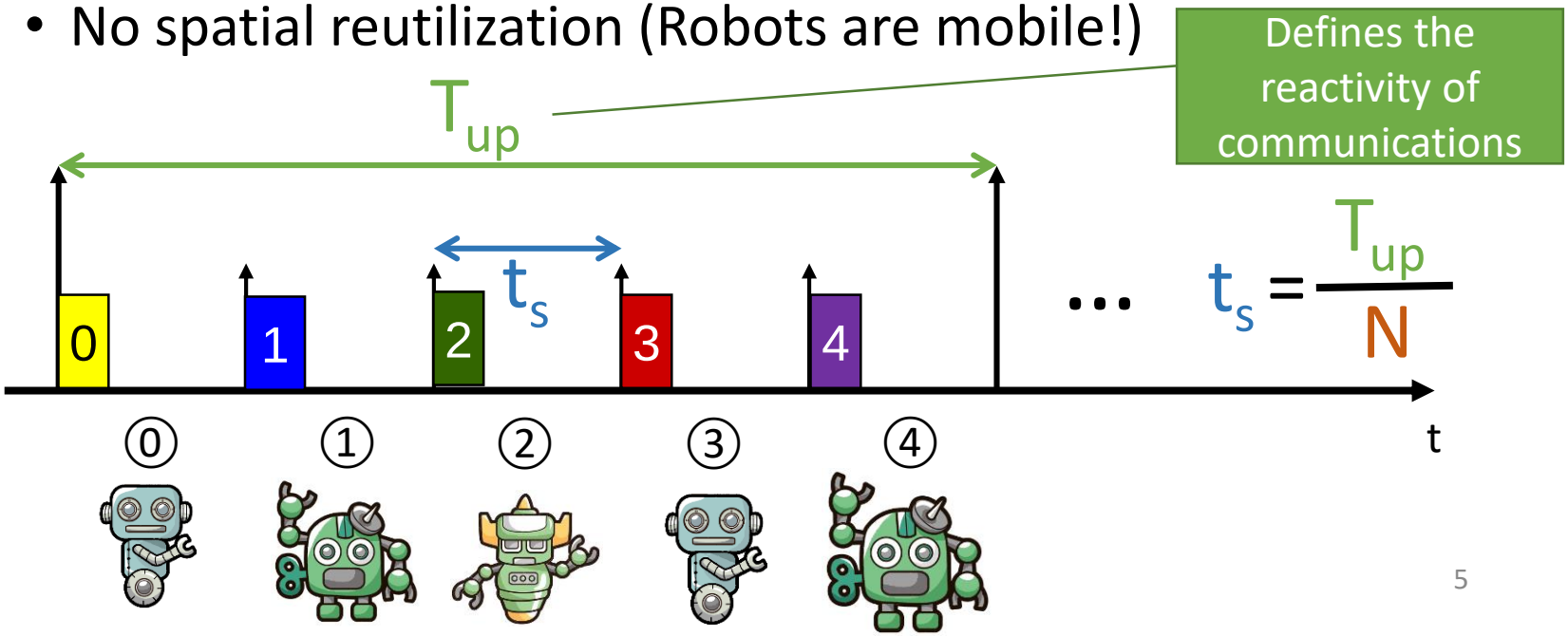
Proposed solution

Use RA-TDMA+ to synchronise transmissions

What is RA-TDMA+ ?

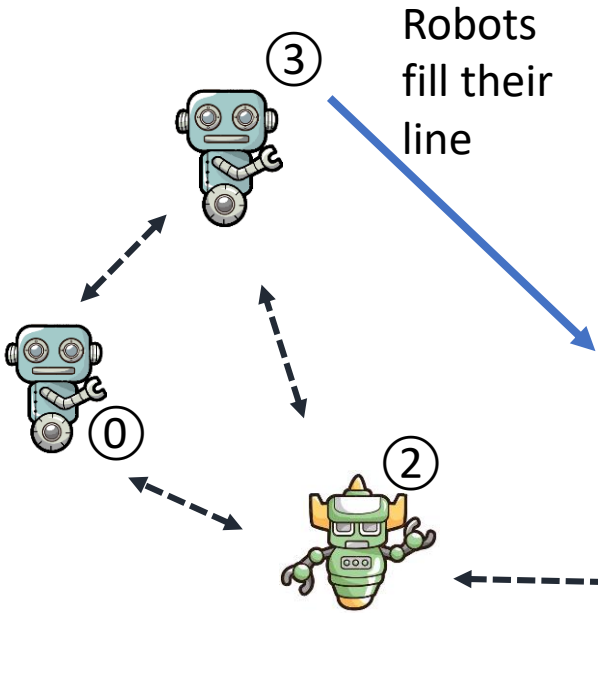
Overlay communication protocol (uses CSMA/CS)

- Creates a TDMA communications round
  - Fixed round duration ➔ UPdate period (T_{up})
  - Divides in N slots ➔ One slot dedicated to each robot
  - No spatial reutilization (Robots are mobile!)



Dynamic team topology

- Supports dynamic topologies (mesh networks)
  - Team topology may change on-line
  - Connectivity matrix is disseminated
    - Filled cooperatively



Robots fill their line

	⓪	①	②	③	④
⓪	0	0	1	1	0
①	0	0	1	0	0
②	1	1	0	1	0
③	1	0	1	0	0
④	0	0	0	0	0

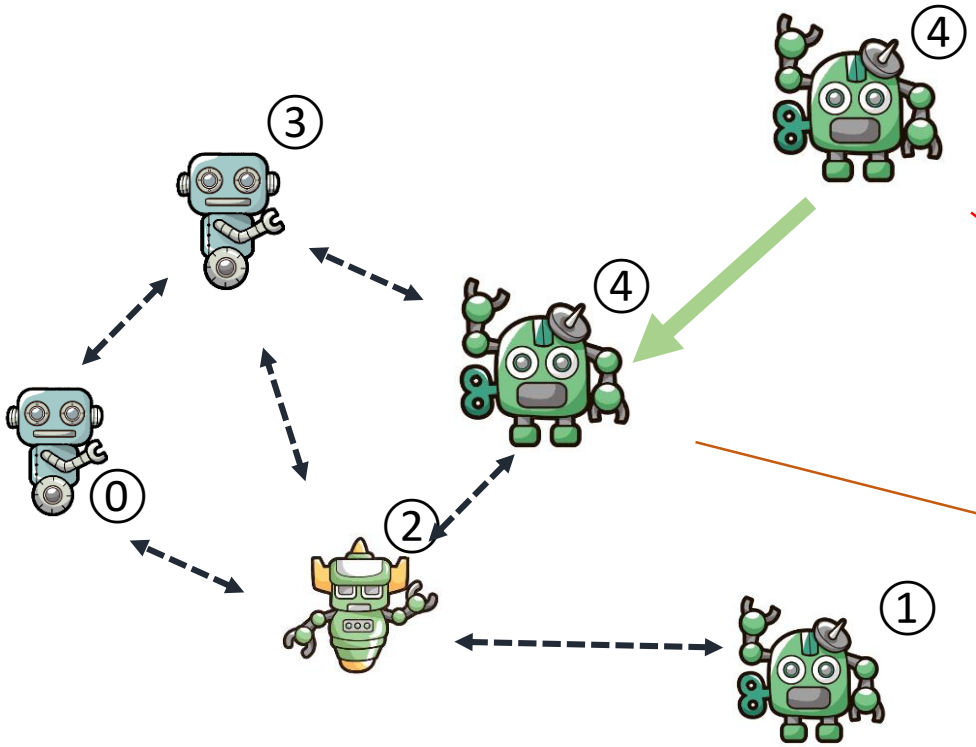
Connected

Disconnected

Dynamic team membership

- Dynamic teams
  - Team composition may change on-line
  - Slots are created/destroyed dynamically

	⓪	①	②	③	④
⓪	0	0	1	1	0
①	0	0	1	0	0
②	1	1	0	1	1
③	1	0	1	0	1
④	0	0	1	1	0

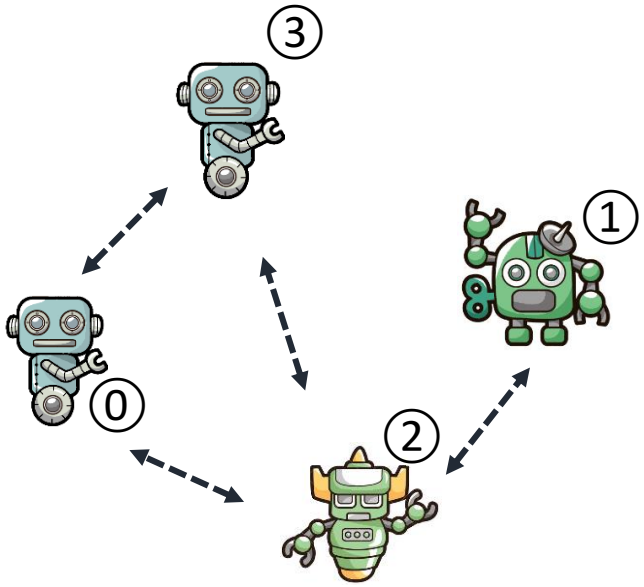


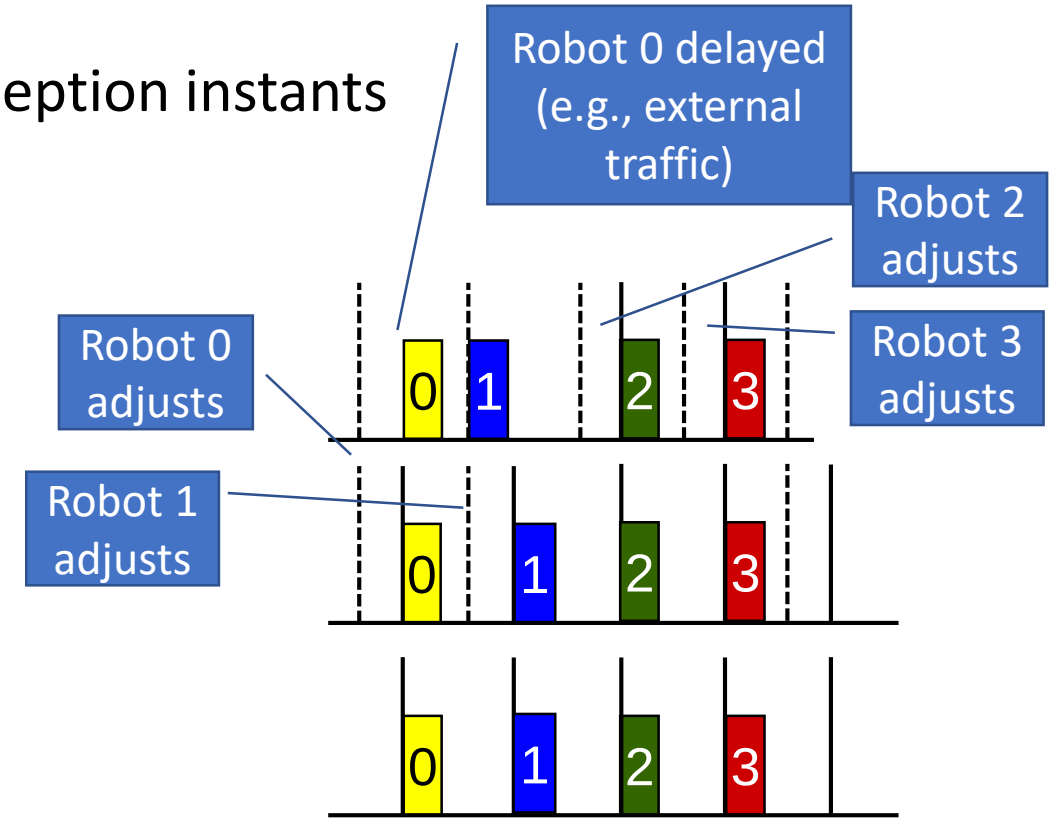
Not a team member

Joins the team

Creating and maintaining the TDMA

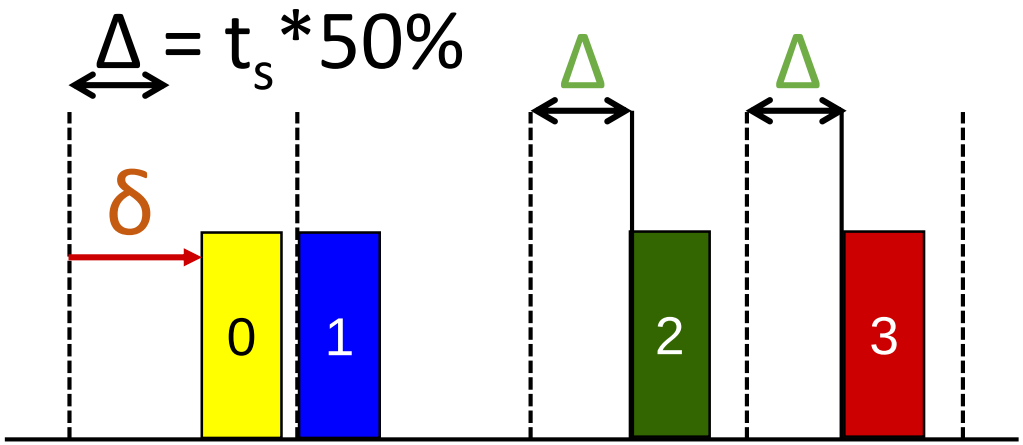
- RA-TDMA+ is a loose synchronisation protocol
  - No global clock
  - Uses message reception instants





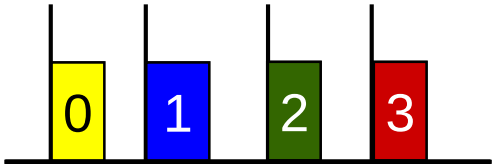
The synchronisation protocol

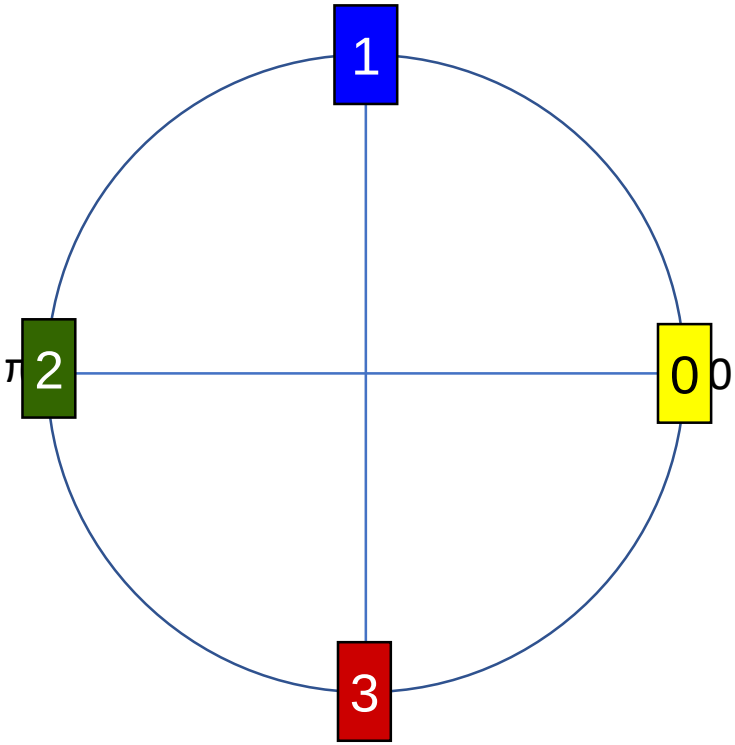
- Robots ONLY delay their communications
  - Anticipating leads to increased BW
  - Parameter Δ limits the maximum delay per round



Reframing the problem

- TDMA round is cyclic
  - We can represent round and slot time as a phase

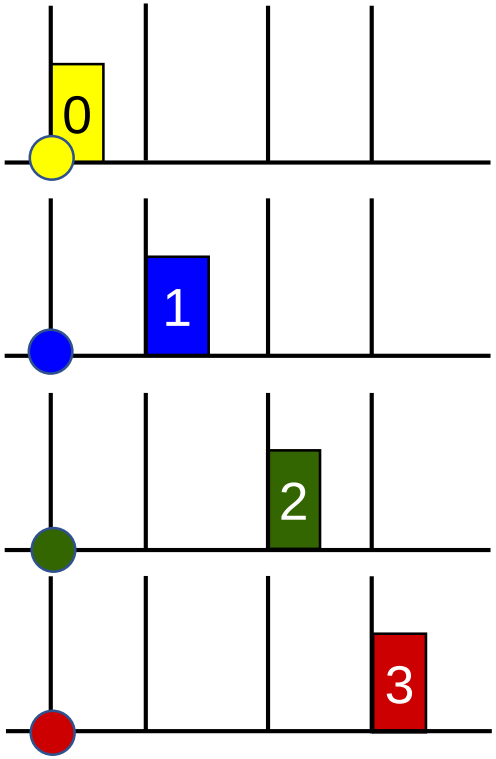


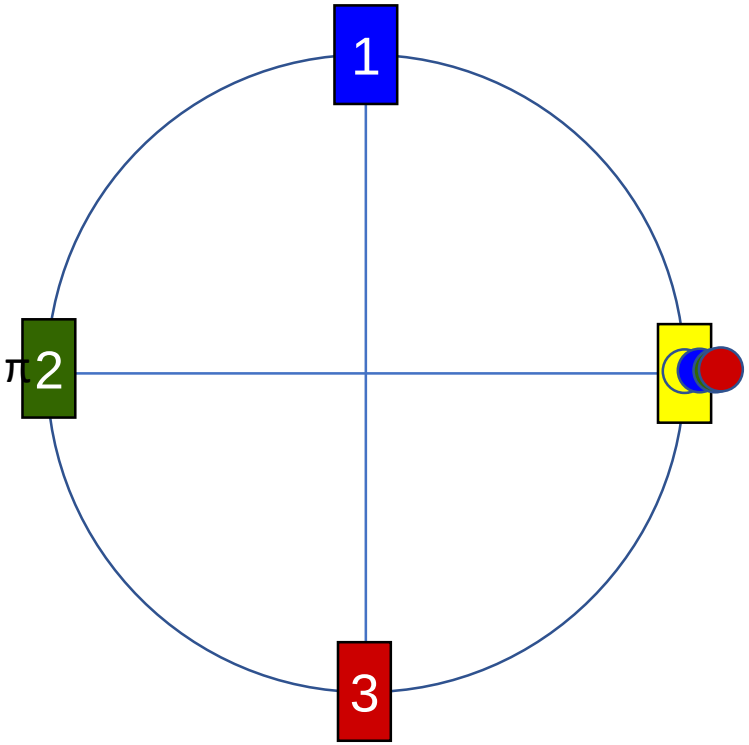


Reframing the problem

Using simple maths

- Message arrival times ➔ round start phases (Φ)

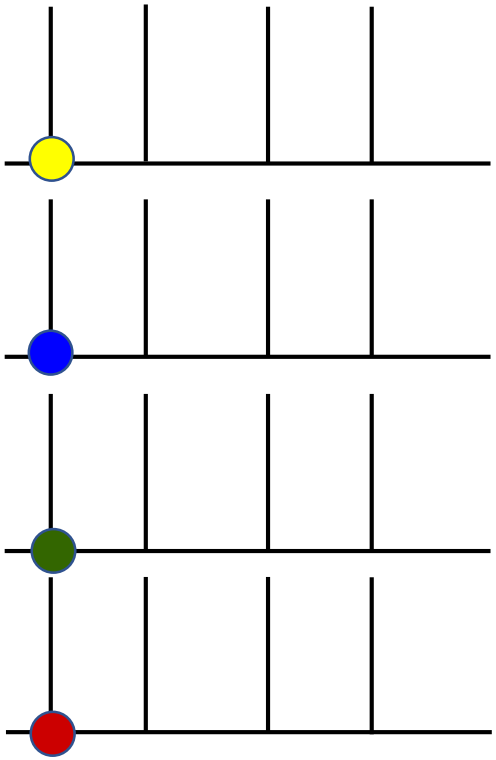


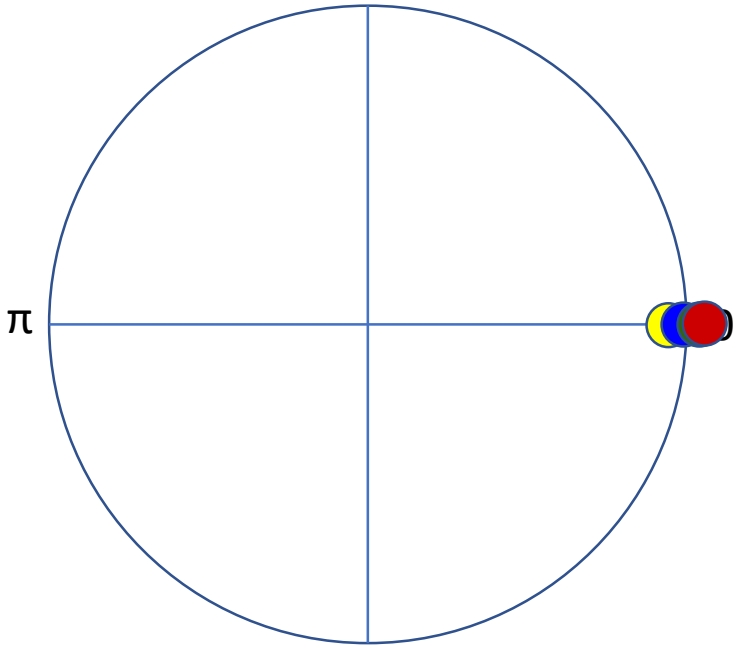


Reframing the problem

Making adjustments to the round (delaying comms.)

- Makes initial round phases increase

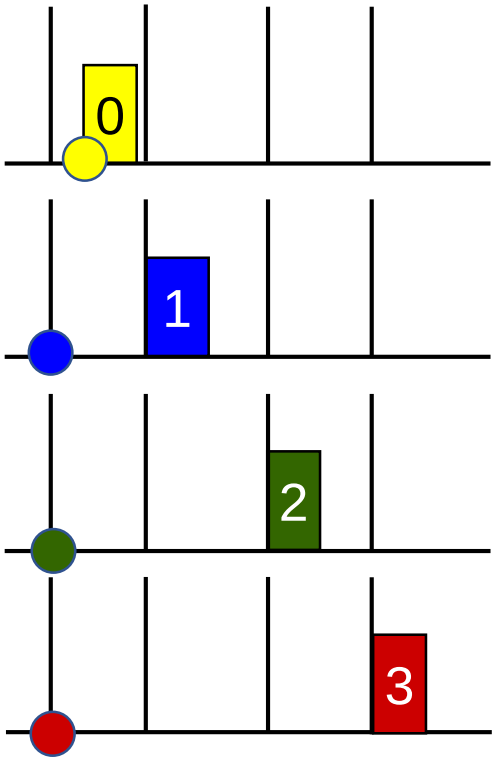


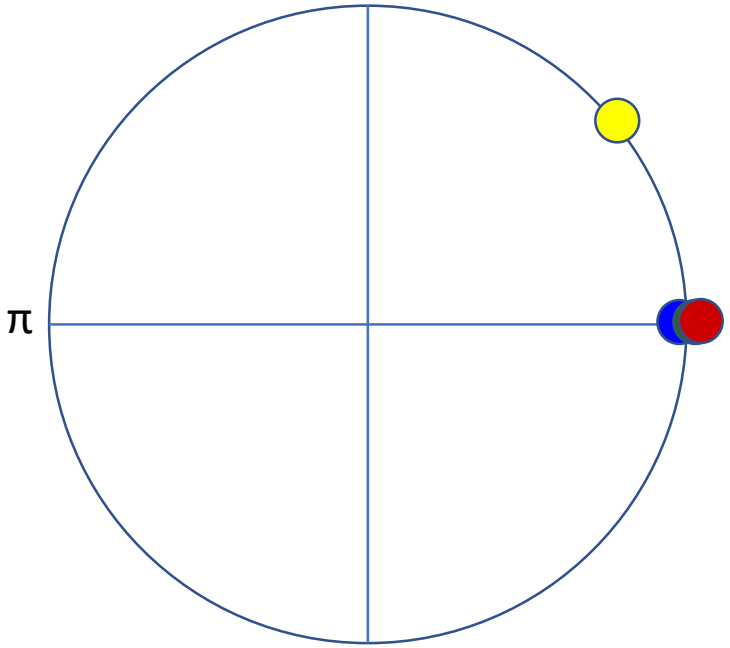


Reframing the problem

Problem can now be stated as:

- Advance the round start phase until it matches the most advanced

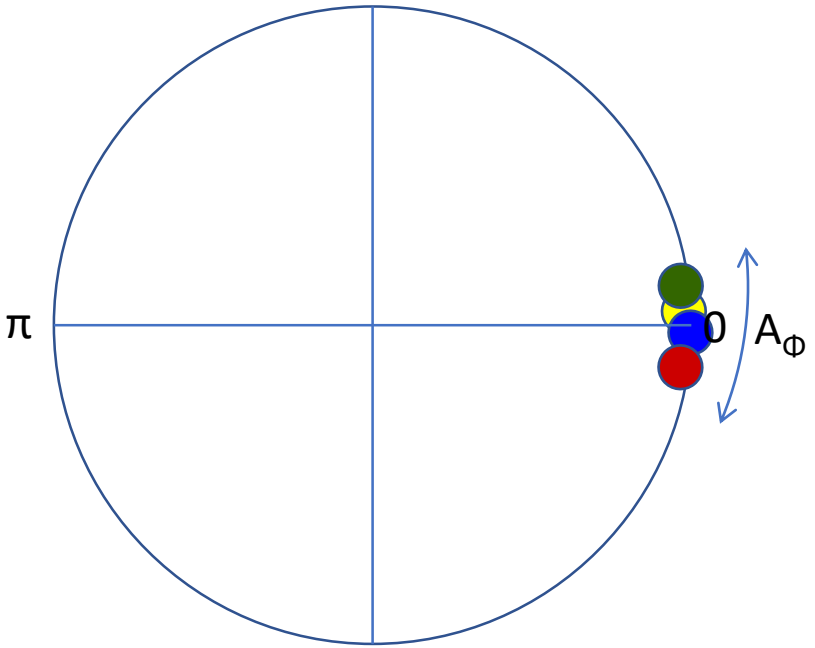




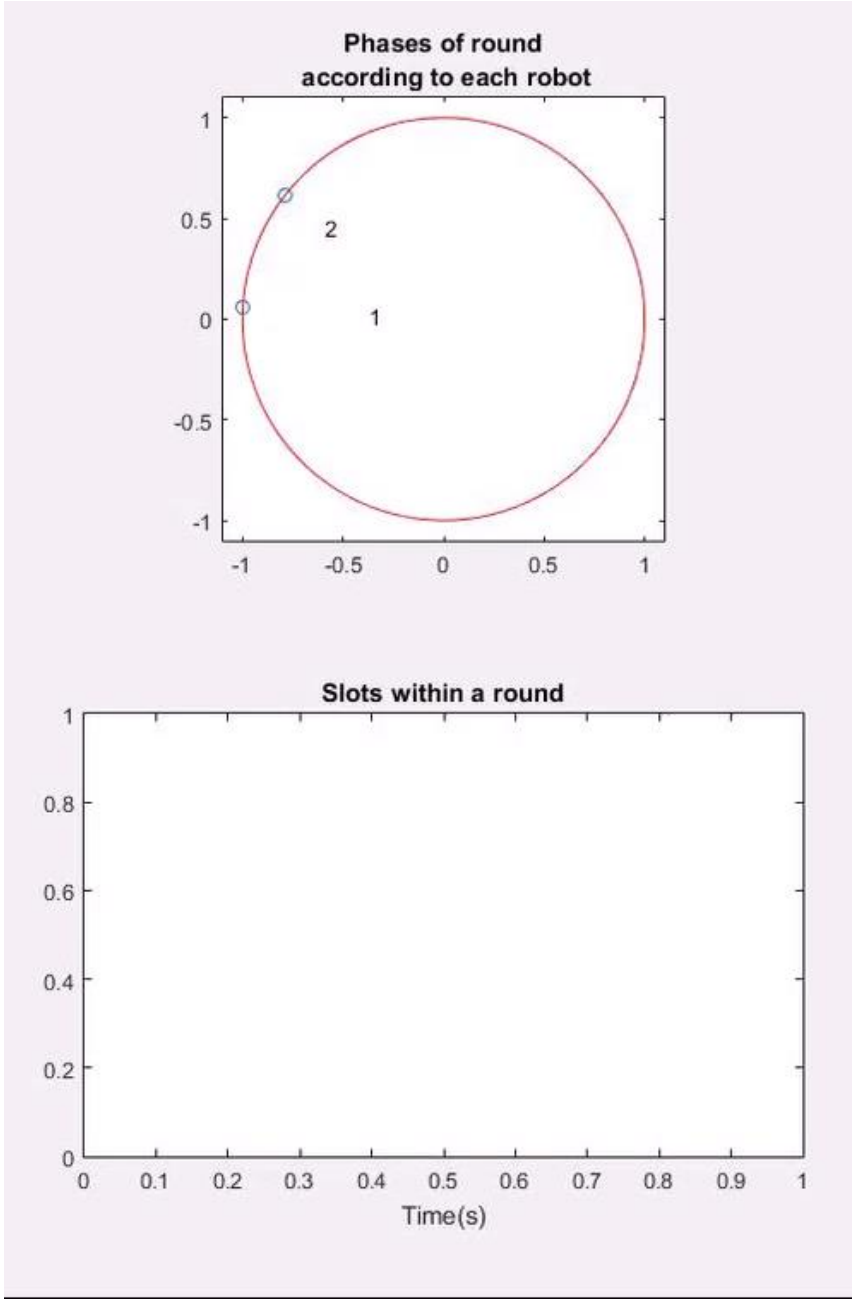
Reframing the problem

Robots are synchronised when the there is an agreement on the round start phase

Or equivalently
When the largest difference between phases (A_Φ) is zero



Video

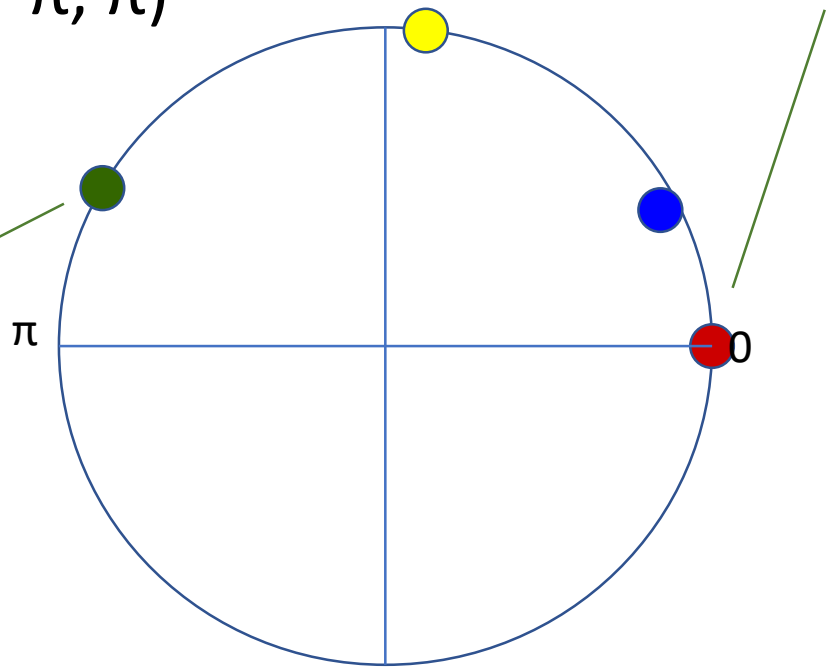

Phases of round
according to each robot
1
0.5
0
-0.5
-1
2
1
-1
-0.5
0
0.5
1
Slots within a round
1
0.8
0.6
0.4
0.2
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Time(s)

When does it work?

If $A_\Phi < \pi$

And we consider Φ in $[-\pi, \pi)$

Independently from the point of observation



Then the green Φ is the most advanced

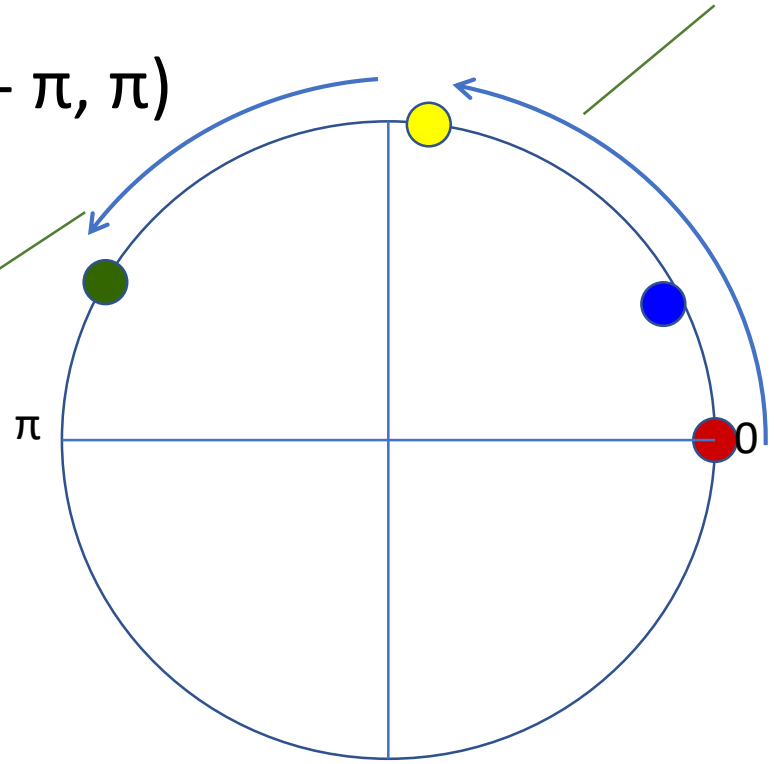
When does it work?

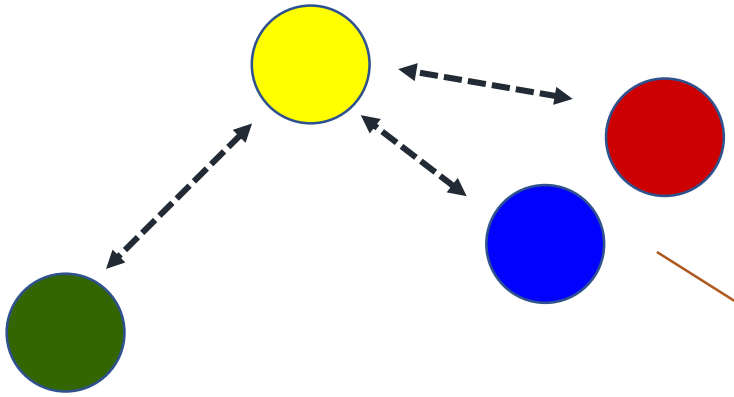
If $A_\Phi < \pi$

And we consider Φ in $[-\pi, \pi)$

Or towards a phase that is moving towards the most advanced

Phases will either advance towards the most advanced



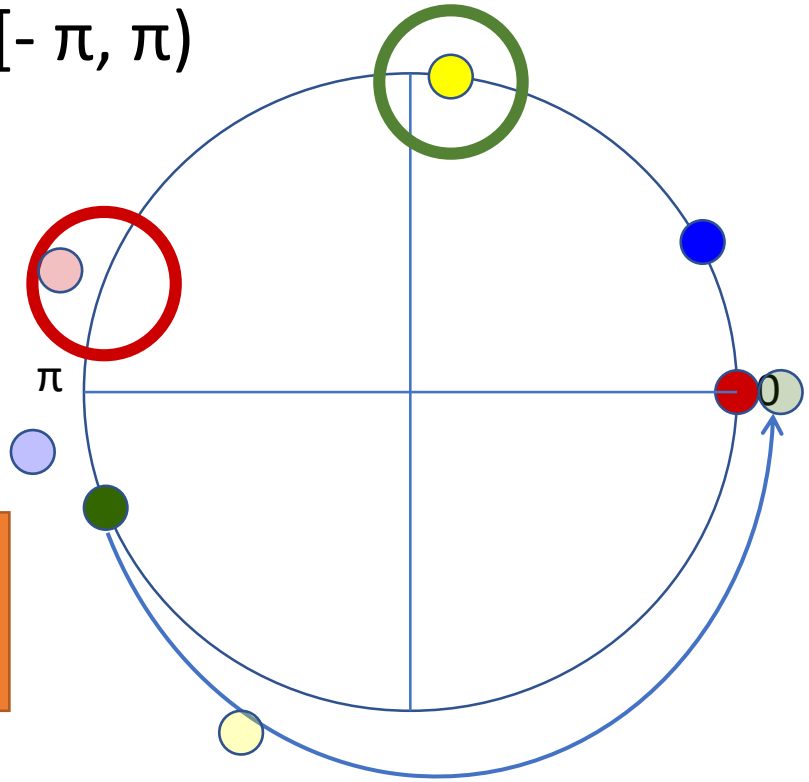


The topology is not (necessarily) fully connected

When it may fail?

If $A_\Phi >= \pi$

And we consider Φ in $[- \pi, \pi)$



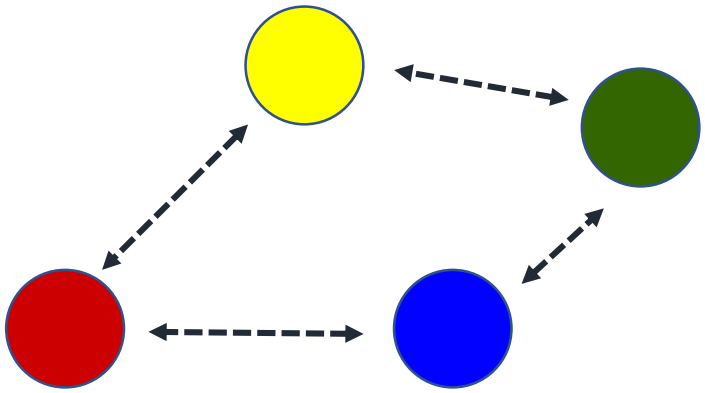
The most advanced Φ depends on the point of observation

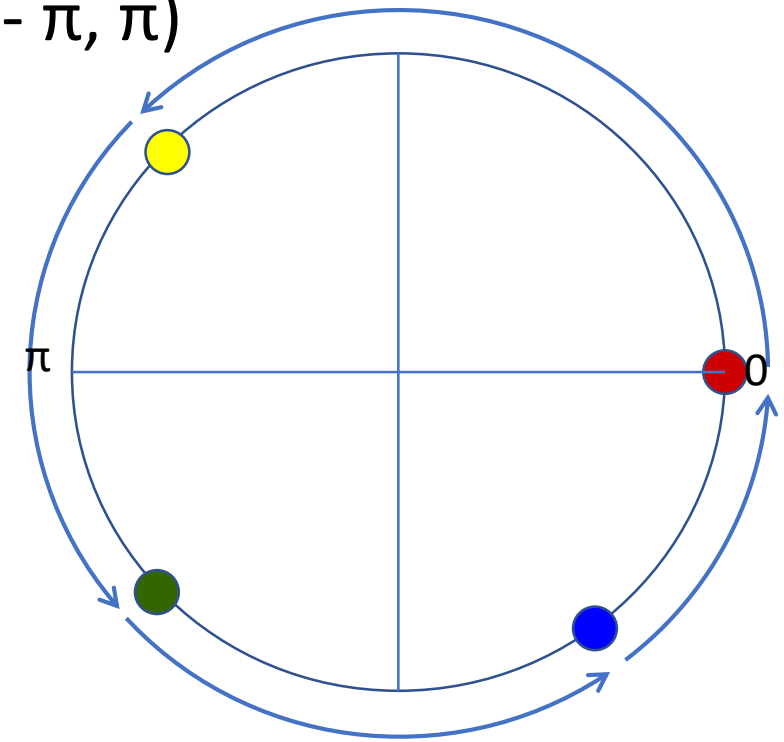
When it may fail?

If $A_\Phi >= \pi$

And we consider Φ in $[- \pi, \pi)$

Phases may chase each other continually for large Δ



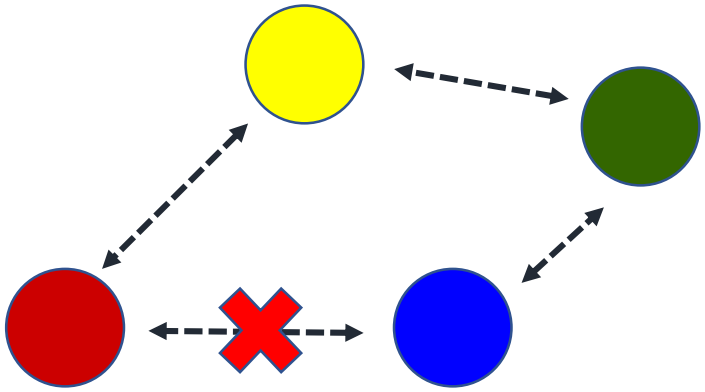


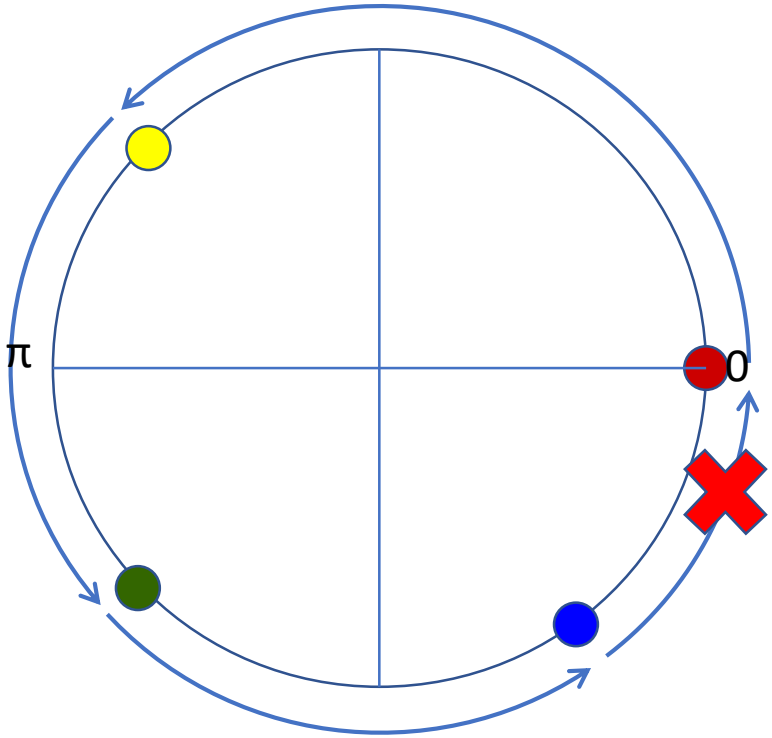
Heuristic to address $A_\Phi >= \pi$

Solution:
Break the cycle

Using the topology information on the connectivity matrix

Create a spanning tree (for synchronisation ONLY)





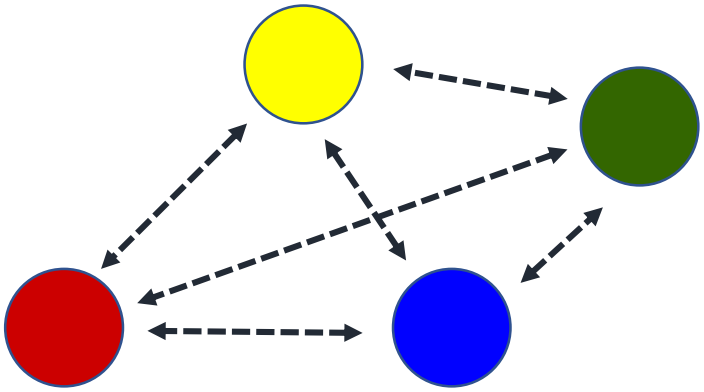
Drawbacks!!!
Longer time to propagate

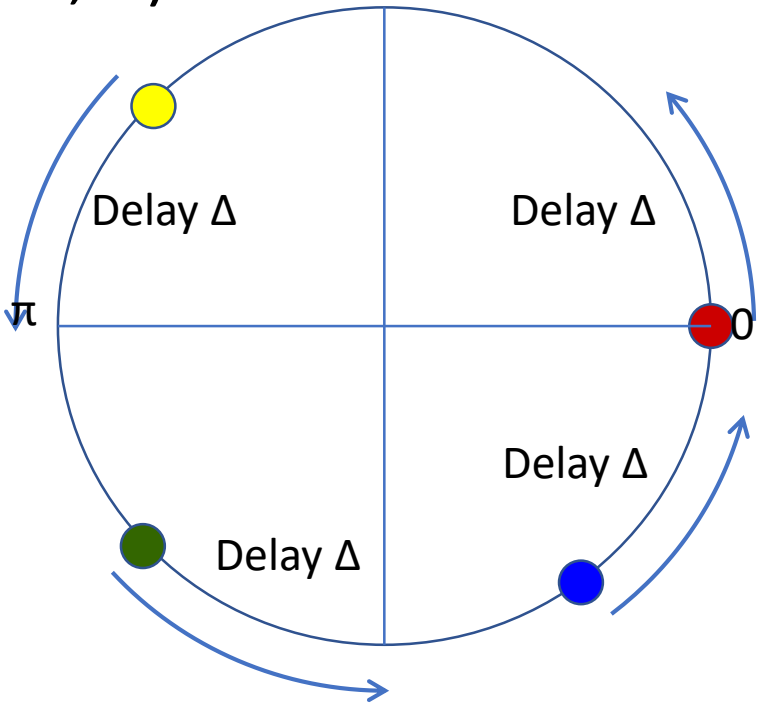
When it may fail?

If $A_\Phi >= \pi$

And we consider Φ in $[- \pi, \pi)$

Phases may all delay the same amount every round
(Δ – The maximum delay)

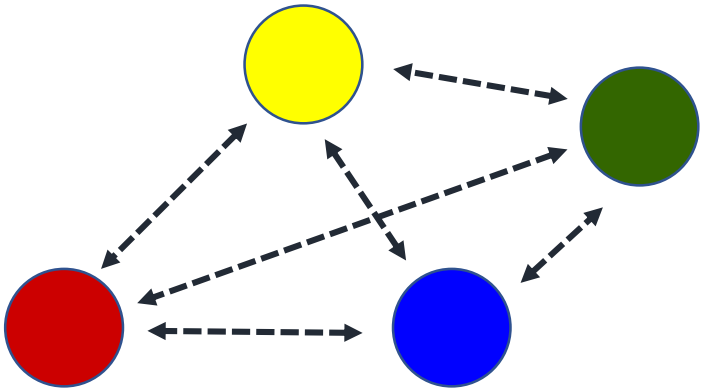


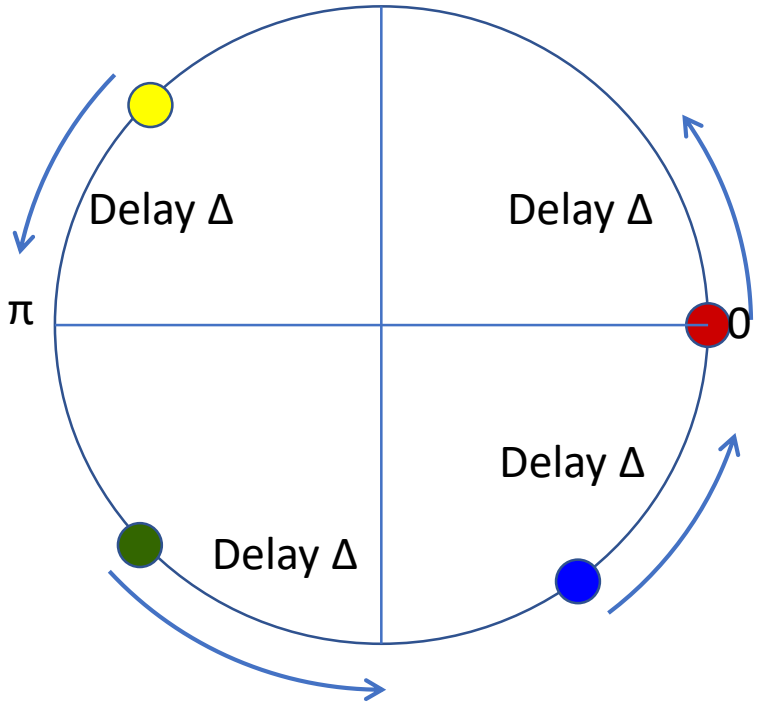


Heuristic to address $A_\Phi >= \pi$

Solution:
Randomise Δ

E.g., Δ' = Δ(0.8+0.2*rnd)





Experimental results

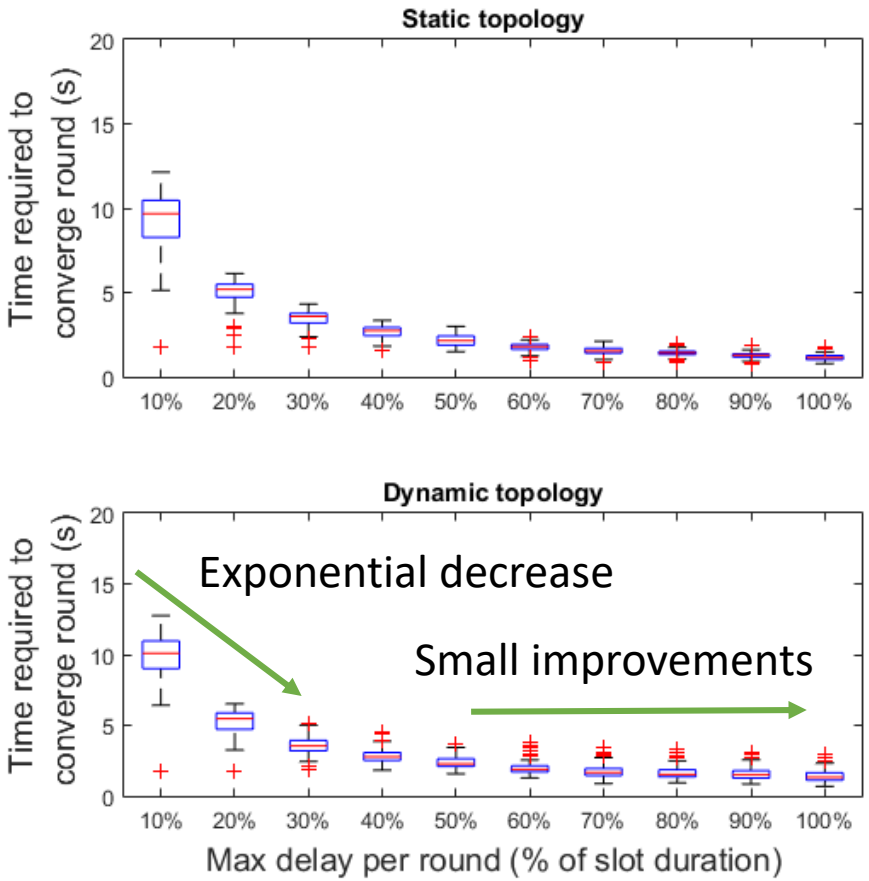
The following experimental results measure

- Impact of different Δ values
- Ability to maintain synchronisation
  - under different load conditions
- Ability to maintain synchronisation
  - Varying team composition and topology

Matlab simulations

Experimental results – Impact of Δ

$A_\Phi < \pi$



Changing positions every 10s

Normal operation
Small delays introduced by external traffic, or OS induced

10 robots
T_{up} = 200ms

Random starting phases Φ

Negligible impact of movement

All robots synchronised in every case

Experimental results – Impact of Δ

Exceptional conditions
Team membership changes

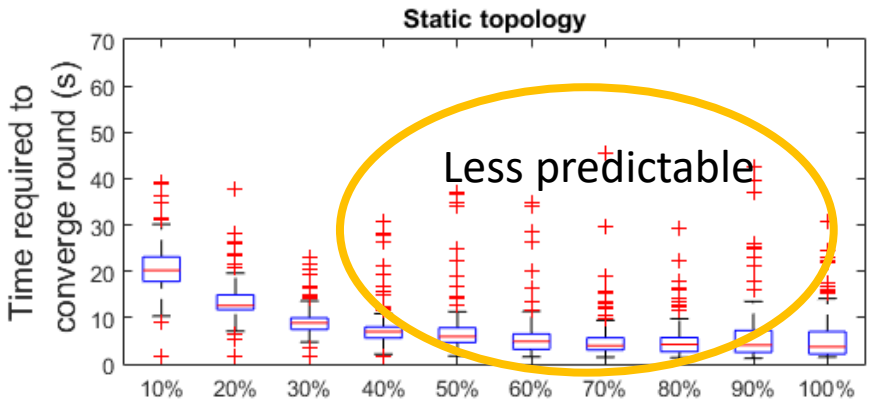
10 robots
T_{up} = 200ms

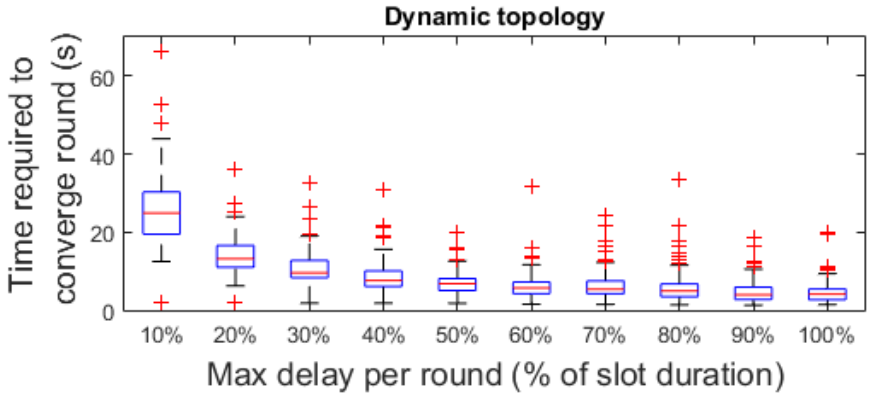
Random starting phases Φ

Negligible impact of movement

All robots synchronised in every case

$A_\Phi >= \pi$



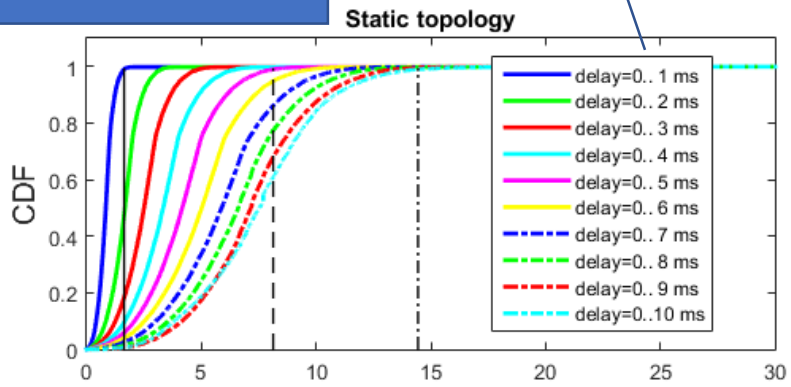


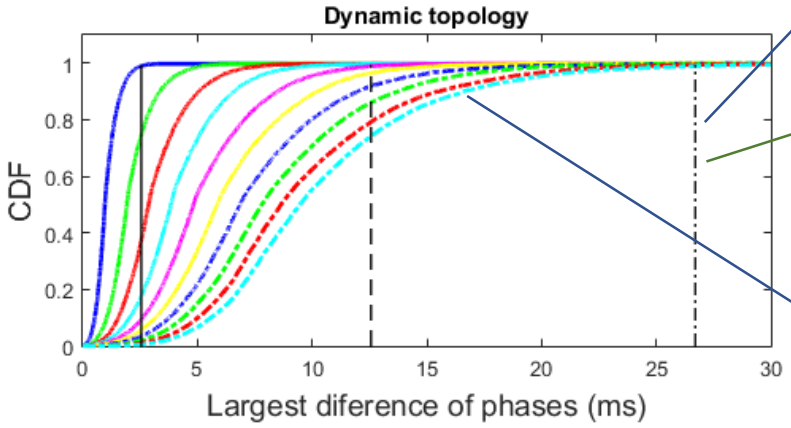
Changing positions every 10s

Experimental results
Impact of delays

Δ=40% of slot duration

Injected uniform random delays





$A_\Phi < \pi$ (<100ms)
Due to no membership changes

Graceful degradation!

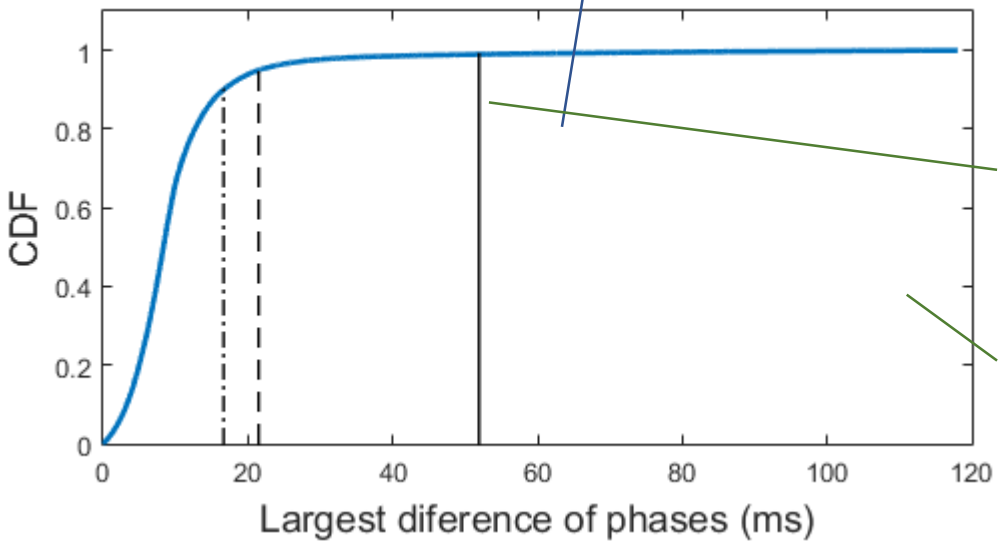
99 percentile for 1,5,and 10ms

Shifted to the right with dynamic topology (<30ms)

Measured the maximum difference of phases for each maximum delay

Experimental results
Changing team membership

Number of robots vary from 3 → 10 → 3 …
Add/remove robot every 50s for 10000s
(200 team membership changes)



Δ=40% of slot duration

Random delays
Uniform 0..10ms

99% << π
(100ms)

> π
When team changes

Conclusions

Proposed a mechanism to create a global TDMA

- Fully distributed
- Dynamic robot teams
- No global clock
- In normal operation → convergence guaranteed
- Exceptional conditions → heuristic was proposed

Questions?