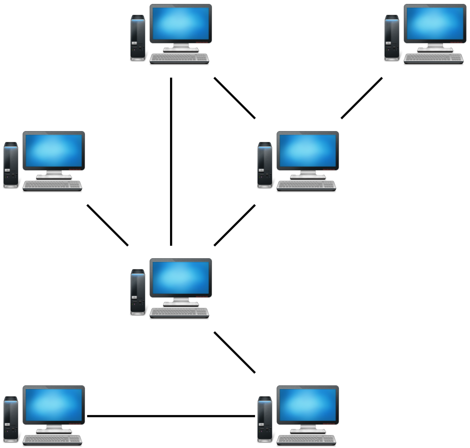
Self-Stabilizing Leader Election in Polynomial Steps[1]

Karine Altisen Alain Cournier Stéphane Devismes
Anaïs Durand Franck Petit

[1] This work has been partially supported by the LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01) and the AGIR project DIAMS.

Distributed Systems = Network + Distributed Algorithm



Processes

- Autonomous = local program, local memory
- Interconnected = communication, asynchronism

Expected Property

Fault-tolerance

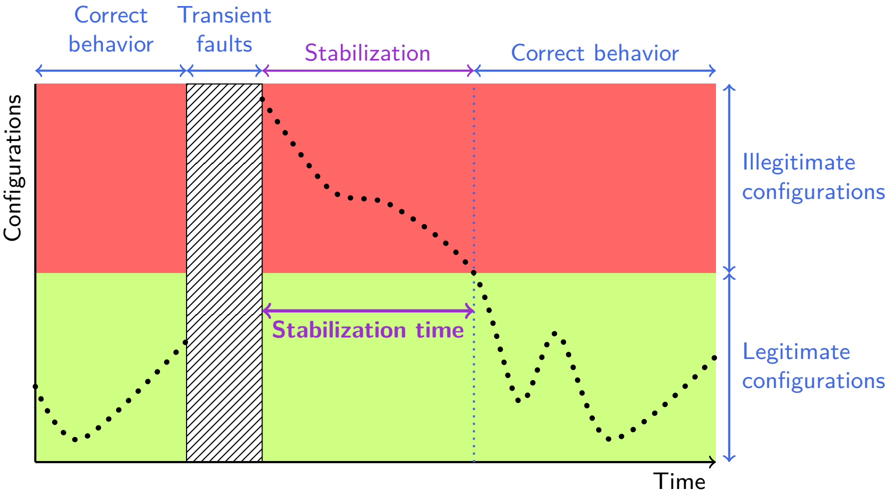
Distributed Algorithm **Design**

- Fault-Tolerance
- Abstract Model
- Design of Algorithm under a Specification
- Proof
- Performances

Distributed Algorithm **Design**

- Fault-Tolerance →**Self-Stabilization**
- Abstract Model →**Locally Shared Memory Model**
- Design of Algorithm under a Specification →**Leader Election**
- Proof
- Performances

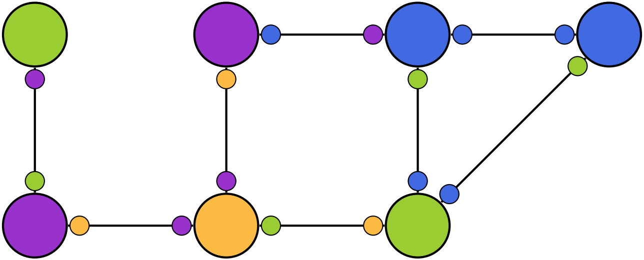
Self-Stabilization[2]



[2] Edsger W. Dijkstra. Self-stabilizing systems in spite of distributed control. 1974

Locally Shared Memory Model
Atomic Step

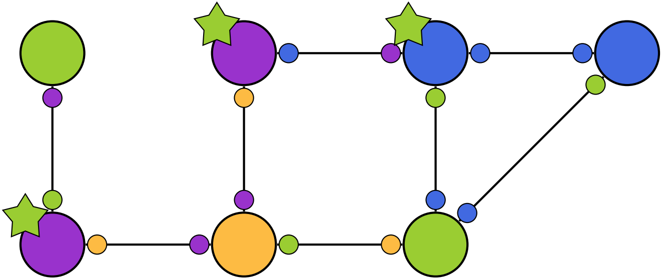
- Reading of the variables of the neighbors



$$Algo = \{rules \ := < guard > \rightarrow < assignments >\}$$

Locally Shared Memory Model
Atomic Step

- Reading of the variables of the neighbors
- Enabled nodes

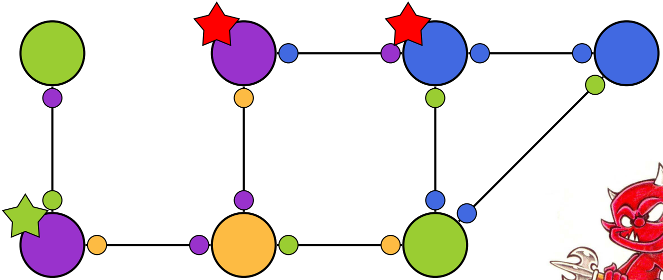


$$Algo = \{rules \ := < guard > \rightarrow < assignments >\}$$

Locally Shared Memory Model

Atomic Step

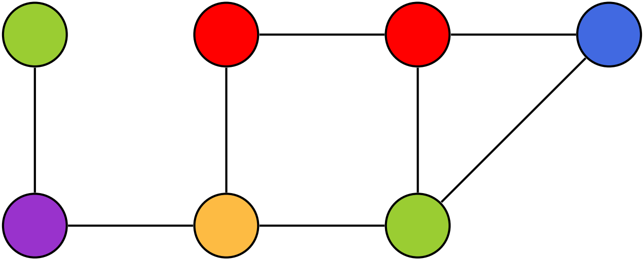
- Reading of the variables of the neighbors
- Enabled nodes
- Daemon election: models the asynchronism



$$Algo = \{rules\ := < guard > \rightarrow < assignments >\}$$

Locally Shared Memory Model
Atomic Step

- Reading of the variables of the neighbors
- Enabled nodes
- Daemon election: models the asynchronism
- Update of the local states



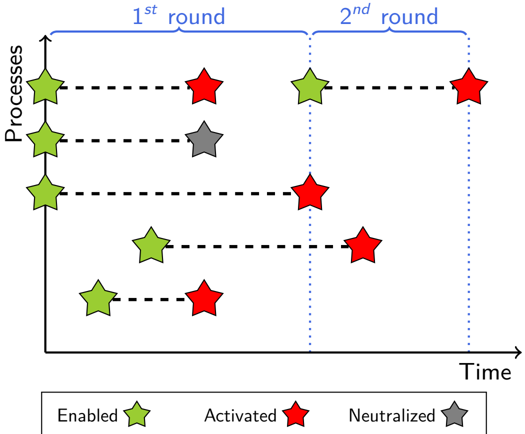
$$Algo = \{rules\ := < guard > \rightarrow < assignments >\}$$

Daemons

- **Asynchronism:** Who is activated? (among enabled nodes)
  - *Synchronous* = all
  - *Central* = exactly one
  - *Distributed* = at least one
- **Fairness:** When? / How often?
  - *Strongly Fair* = ∞ enabled $\rightarrow$ ∞ activation
  - *Weakly Fair* = cont. enabled $\rightarrow$ activation in finite time
  - *Unfair* = _

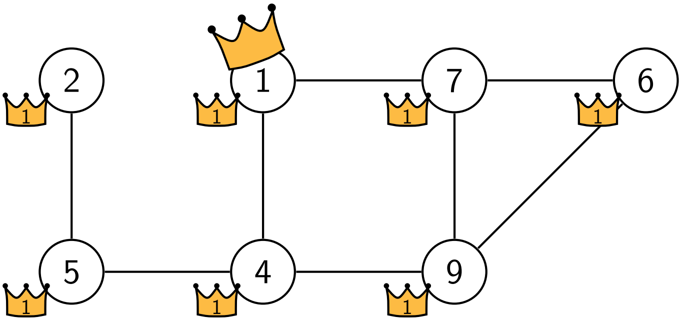
Complexity

- **In space**: memory requirement in bits
- **In time** (mainly stabilization time)
  - In (atomic) *steps*
  - In *rounds* (execution time according slowest processes)



Leader Election

- Distinguish a process: the **leader**
- Every process eventually knows the identifier of the leader



Problem

- **Silent Self-stabilizing Leader Election**
- **Locally shared memory** model:
  - Locally shared variables
  - Read/write atomicity
  - Distributed **unfair** daemon (scheduler)
- Network:
  - Any connected topology
  - Bidirectional
  - **Identified**
- **No global knowledge** on the network

State of the Art

Model	Paper	Knowledge			Daemon	Complexity			Silent
		D	N	B		Memory	Rounds	Steps	
Message Passing	Afek, Bremler, 1998			x		$\Theta(\log n)$	$O(n)$	?	✓
	Awerbuch *et al*, 1993	x				$\Theta(\log D \log n)$	$O(\mathcal{D})$	?	✓
	Burman, Kutten, 2007	x				$\Theta(\log D \log n)$	$O(\mathcal{D})$	?	✓
Locally Shared Memory	Dolev, Herman, 1997		x		Fair	$\Theta(N \log N)$	$O(\mathcal{D})$	?	
	Arora, Gouda, 1994		x		Weakly Fair	$\Theta(\log N)$	$O(N)$	?	✓
	Datta *et al*, 2010				Unfair	unbounded	$O(n)$	?	✓
	Kravchik, Kutten, 2013				Synchronous	$\Theta(\log n)$	$O(\mathcal{D})$	?	✓
	2 x Datta *et al*, 2011				Unfair	$\Theta(\log n)$	$O(n)$	?	✓

$\mathcal{D}$: Diameter
$D \geq \mathcal{D}$: Upper bound on the diameter

n: Number of nodes
$N \geq n$: Upper bound on the number of nodes

B: Upper bound on the link-capacity

Our Contribution

Algorithm $\mathcal{LE}$

- Memory requirement asymptotically optimal: $\Theta(\log n)$ bits/process
- Stabilization time (worst case):
  - $3n + \mathcal{D}$ rounds
  - Lower Bound: $\frac{n^3}{6} + \frac{3}{2}n^2 - \frac{8}{3}n + 2$ steps,
    Upper Bound: $\frac{n^3}{2} + 2n^2 + \frac{n}{2} + 1$ steps

Analytical Study of Datta *et al*, 2011

Stabilization time not polynomial in steps:

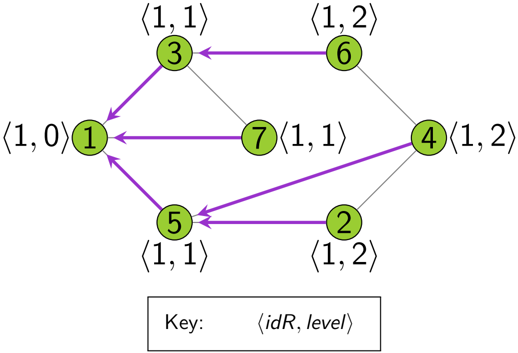
1. Self-stabilizing Leader Election in Optimal Space under an Arbitrary Scheduler:
   - $\forall \alpha \geq 3$, $\exists$ networks and executions in $\Omega(n^{\alpha+1})$ steps.
2. An $O(n)$-time Self-stabilizing Leader Election Algorithm:
   - $\forall n \geq 5$, $\exists$ networks and executions in $\Omega(2^{\lfloor \frac{n-1}{4} \rfloor})$ steps.

Design of the Leader Election Algorithm

Simplified Algorithm (Non Self-stabilizing)
Join a Tree

3 variables per process p

- $p.idR \in \mathbb{N}$: ID of the root
- $p.par \in \mathcal{N}_p \cup \{p\}$: Parent pointer
- $p.level \in \mathbb{N}$: Level



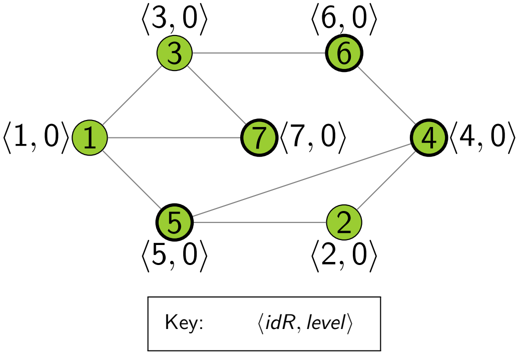
Simplified Algorithm (Non Self-stabilizing)
Join a Tree

3 variables per process p

- $p.idR \in \mathbb{N}$: ID of the root
- $p.par \in \mathcal{N}_p \cup \{p\}$: Parent pointer
- $p.level \in \mathbb{N}$: Level

Initial Configuration

- $p.idR = p$
- $p.par = p$
- $p.level = 0$



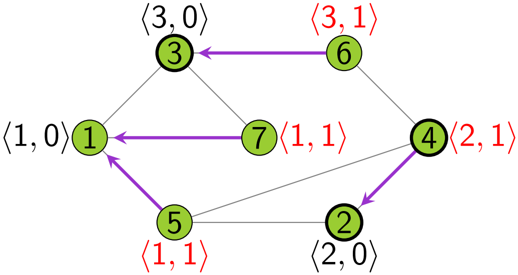
Simplified Algorithm (Non Self-stabilizing)
Join a Tree

3 variables per process p

- $p.idR \in \mathbb{N}$: ID of the root
- $p.par \in \mathcal{N}_p \cup \{p\}$: Parent pointer
- $p.level \in \mathbb{N}$: Level

Initial Configuration

- $p.idR = p$
- $p.par = p$
- $p.level = 0$



Key: $\langle idR, level \rangle$

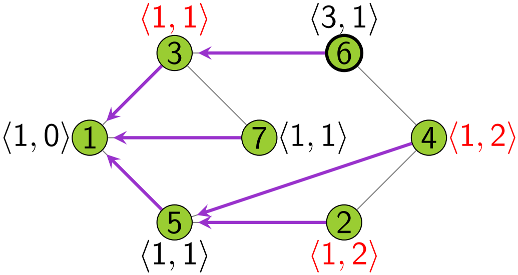
Simplified Algorithm (Non Self-stabilizing)
Join a Tree

3 variables per process p

- $p.idR \in \mathbb{N}$: ID of the root
- $p.par \in \mathcal{N}_p \cup \{p\}$: Parent pointer
- $p.level \in \mathbb{N}$: Level

Initial Configuration

- $p.idR = p$
- $p.par = p$
- $p.level = 0$



Key: $\langle idR, level \rangle$

Simplified Algorithm (Non Self-stabilizing)

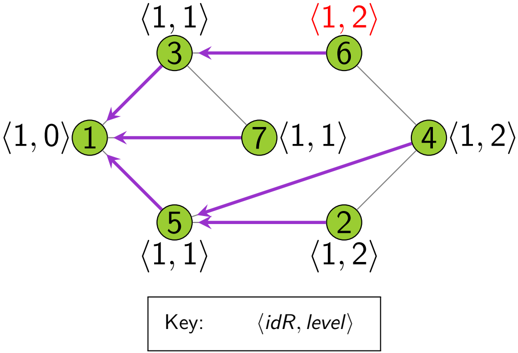
Join a Tree

3 variables per process p

- $p.idR \in \mathbb{N}$: ID of the root
- $p.par \in \mathcal{N}_p \cup \{p\}$: Parent pointer
- $p.level \in \mathbb{N}$: Level

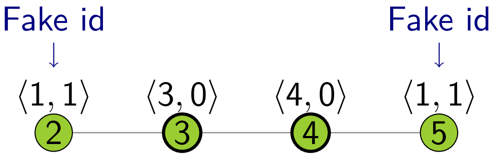
Initial Configuration

- $p.idR = p$
- $p.par = p$
- $p.level = 0$



Simplified Algorithm (Self-Stabilizing?)

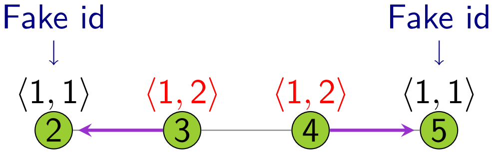
Self-stabilization $\Longrightarrow$ Arbitrary initialization $\Longrightarrow$ Fake ids



Key: $\langle idR, level \rangle$

Simplified Algorithm (Self-Stabilizing?)

Self-stabilization $\Longrightarrow$ Arbitrary initialization $\Longrightarrow$ Fake ids



Key: $\langle idR, level \rangle$

Simplified Algorithm: Removal of Fake Ids

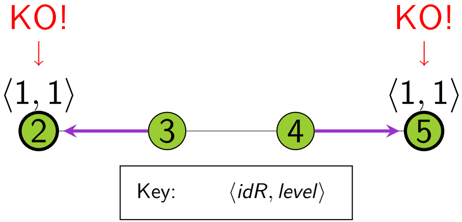
Reset

OK

- root: $p.par = p; p.level = 0; p.idR = p$
- non-root: $p > p.idR \geq p.par.idR; p.level = p.par.level + 1$

Reset

- $p.idR := p;\ p.par := p;\ p.level := 0$



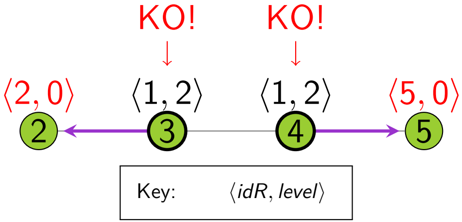
Simplified Algorithm: Removal of Fake Ids
Reset

OK

- root: $p.par = p; p.level = 0; p.idR = p$
- non-root: $p > p.idR \geq p.par.idR; p.level = p.par.level + 1$

Reset

- $p.idR := p;\ p.par := p;\ p.level := 0$



Simplified Algorithm: Removal of Fake Ids

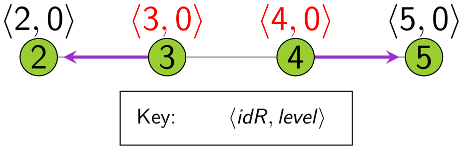
Reset

OK

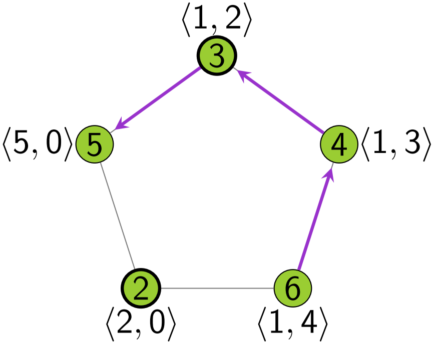
- root: $p.par = p; p.level = 0; p.idR = p$
- non-root: $p > p.idR \geq p.par.idR; p.level = p.par.level + 1$

Reset

- $p.idR := p;\ p.par := p;\ p.level := 0$

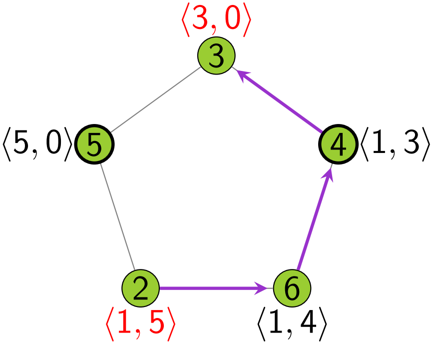


Simplified Algorithm: Removal of Fake Ids
Reset



Key: $\langle idR, level \rangle$

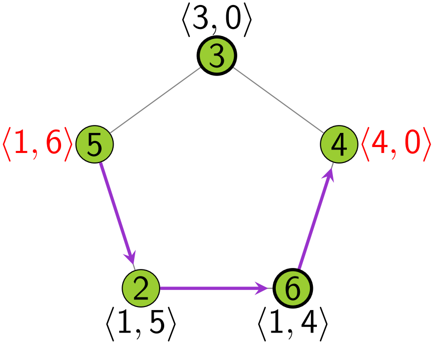
Simplified Algorithm: Removal of Fake Ids
Reset



Key:	$\langle idR, level \rangle$

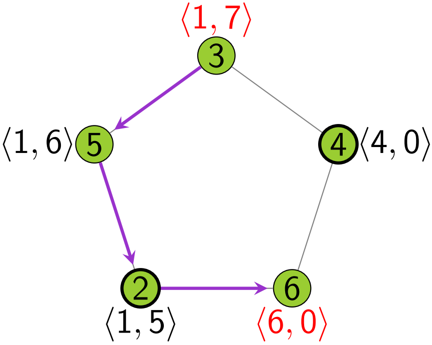
Simplified Algorithm: Removal of Fake Ids

Reset



Key:	$\langle idR, level \rangle$

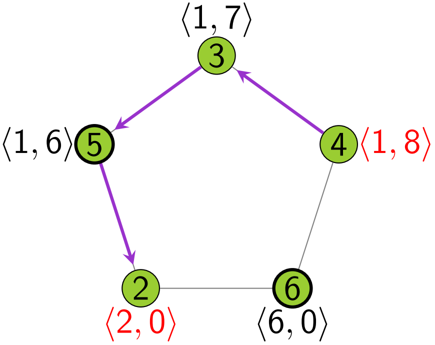
Simplified Algorithm: Removal of Fake Ids
Reset



Key: $\langle idR, level \rangle$

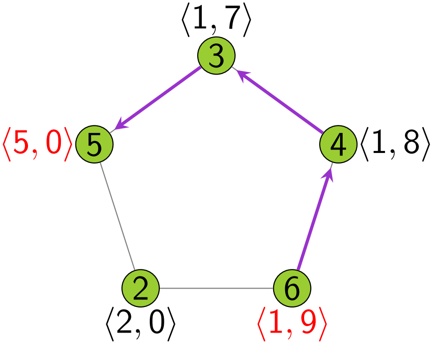
Simplified Algorithm: Removal of Fake Ids

Reset



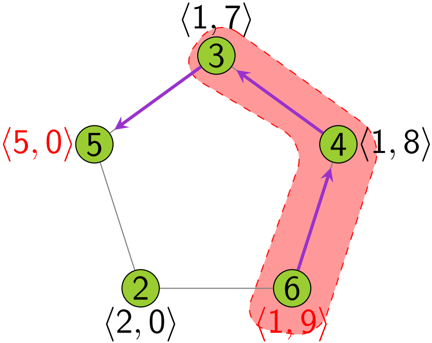
Key: $\langle idR, level \rangle$

Simplified Algorithm: Removal of Fake Ids
Reset



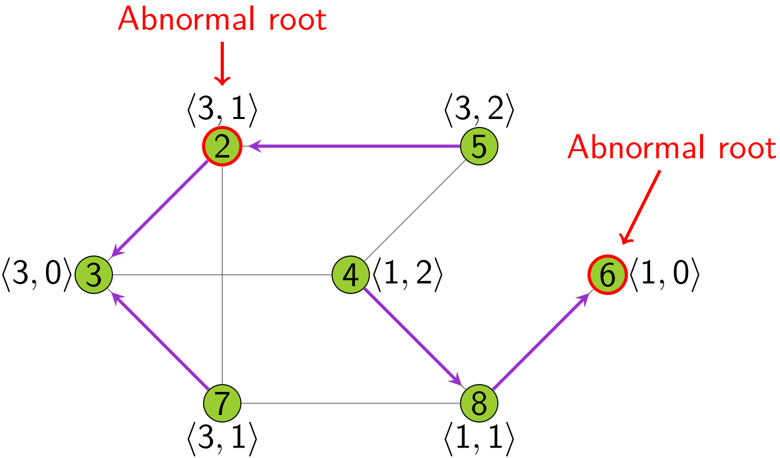
Key: ⟨*idR*, *level*⟩

Simplified Algorithm: Removal of Fake Ids
Reset



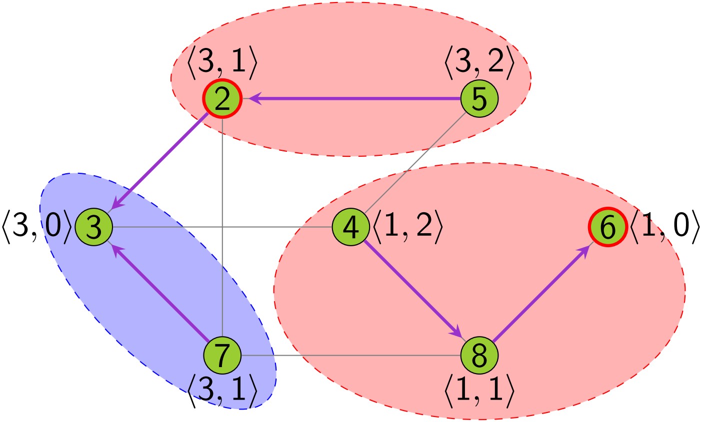
Key:	⟨*idR*, *level*⟩

Abnormal Trees



Key:	$\langle idR, level\rangle$

Abnormal Trees



Key: ⟨*idR*, *level*⟩

Abnormal Trees: Removal

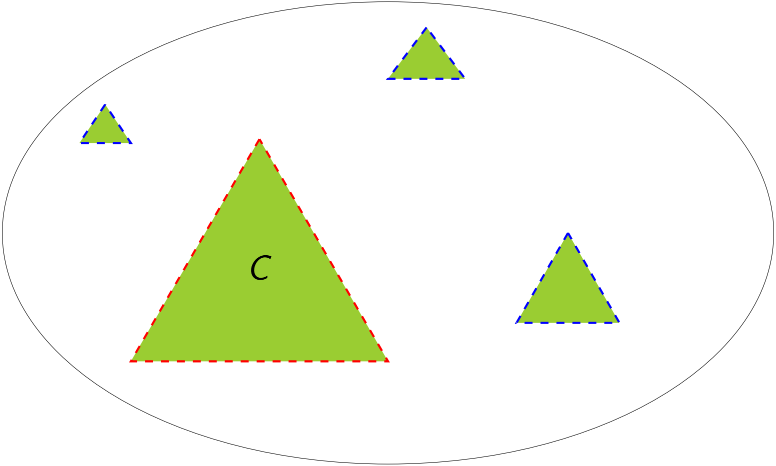
Freeze Before Remove

Add a variable $Status \in \{C, EB, EF\}$

- C means "not involved in a tree removal":
  - Only process of status C can join a tree and
  - only by choosing a process of status C as parent
- EB: Error Broadcast
- EF: Error Feedback

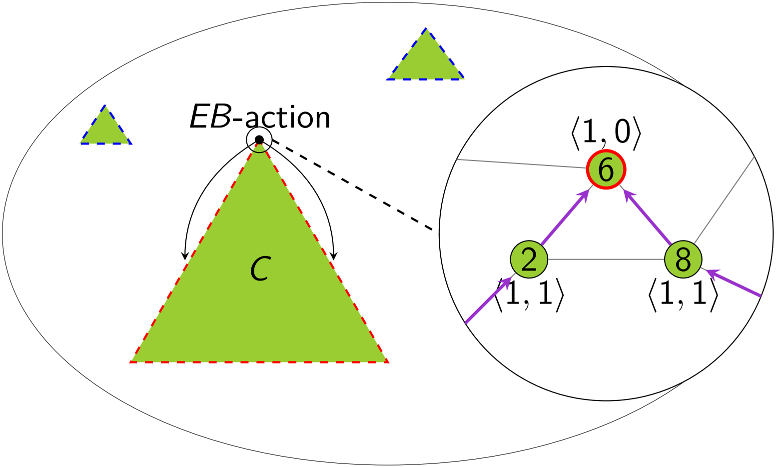
OK/KO! should be modified to take possible inconsistencies of variables *Status* into account!

Freeze Before Remove



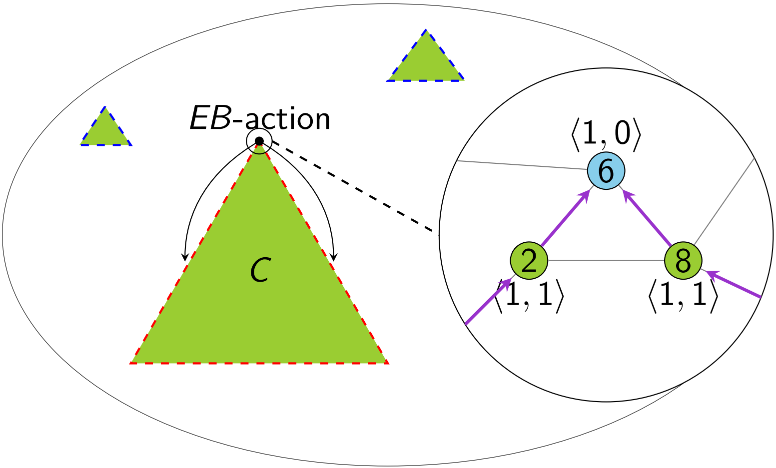
Key: ⟨*idR*, *level*⟩ *Clean* *EBroadcast* *EFeedback*

Freeze Before Remove



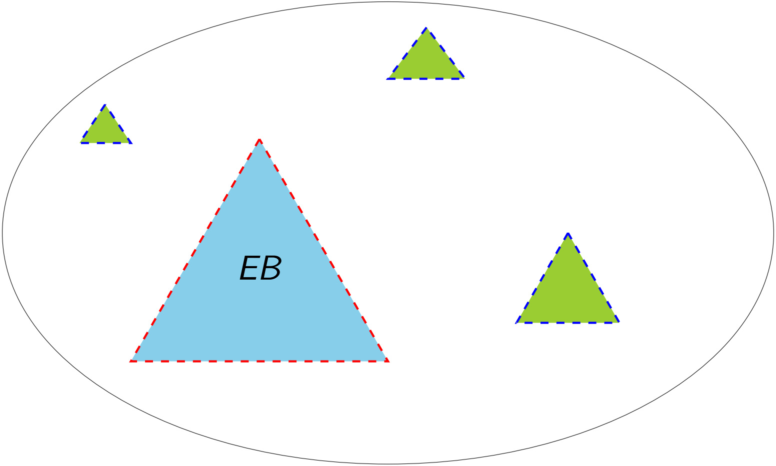
Key: $\langle idR, level \rangle$ $Clean$ $EBroadcast$ $EFeedback$

Freeze Before Remove



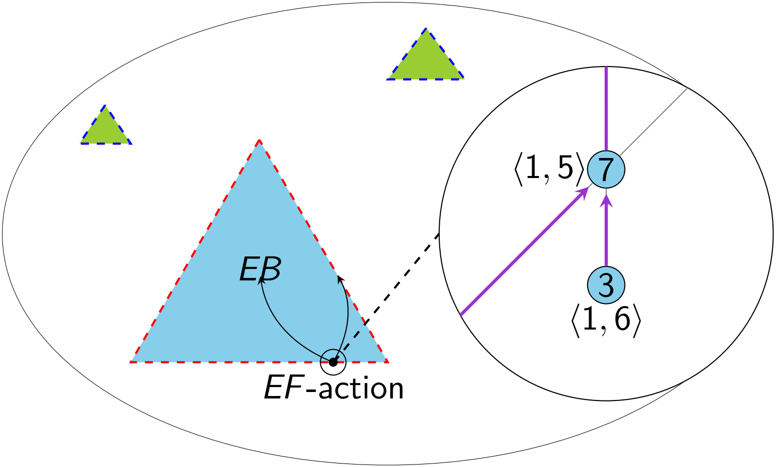
Key: ⟨*idR*, *level*⟩ *Clean* *EBroadcast* *EFeedback*

Freeze Before Remove



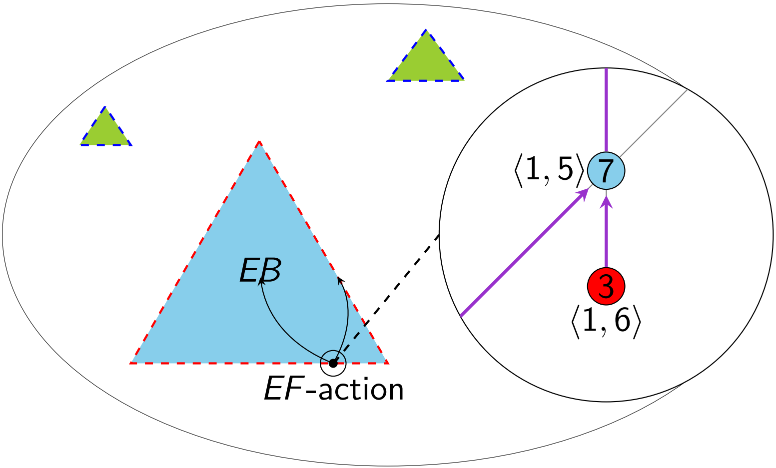
Key:	$\langle idR, level \rangle$	$Clean$	$EBroadcast$	$EFeedback$

Freeze Before Remove



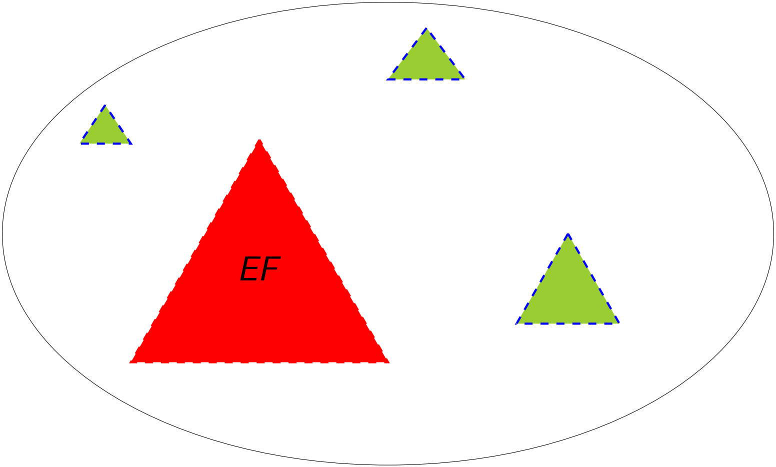
Key:	$\langle idR, level\rangle$	$Clean$	$EBroadcast$	$EFeedback$

Freeze Before Remove



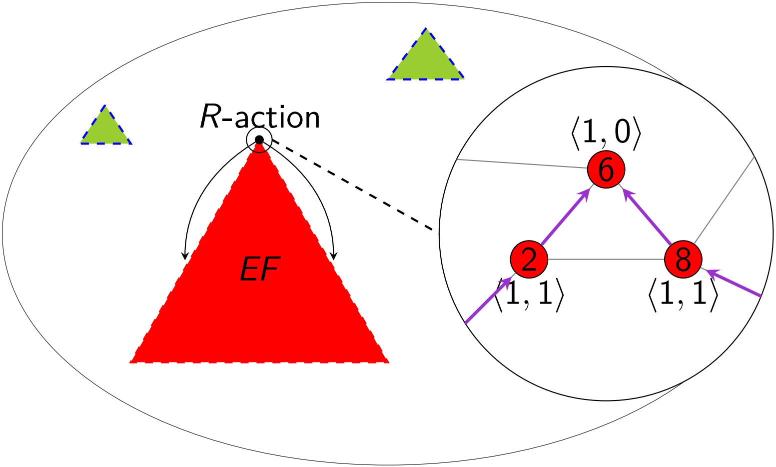
Key: $\langle idR, level \rangle$ $Clean$ $EBroadcast$ $EFeedback$

Freeze Before Remove



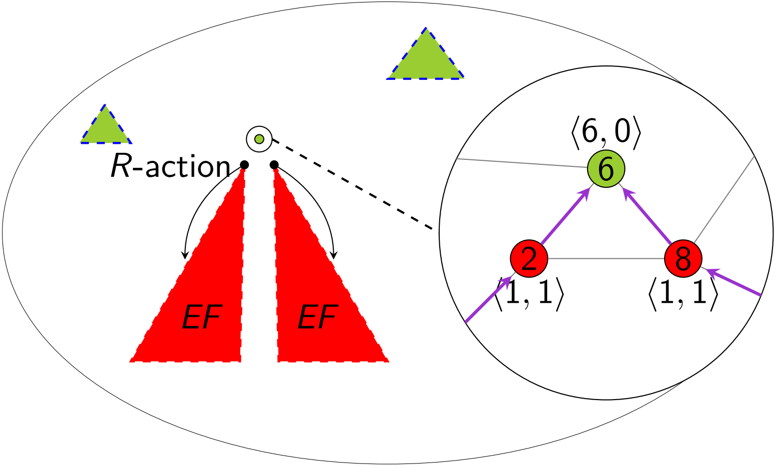
Key: $\langle idR, level \rangle$ $Clean$ $EBroadcast$ $EFeedback$

Freeze Before Remove



Key: $\langle idR, level \rangle$ $Clean$ $EBroadcast$ $EFeedback$

Freeze Before Remove



Key: $\langle idR, level \rangle$ $Clean$ $EBroadcast$ $EFeedback$

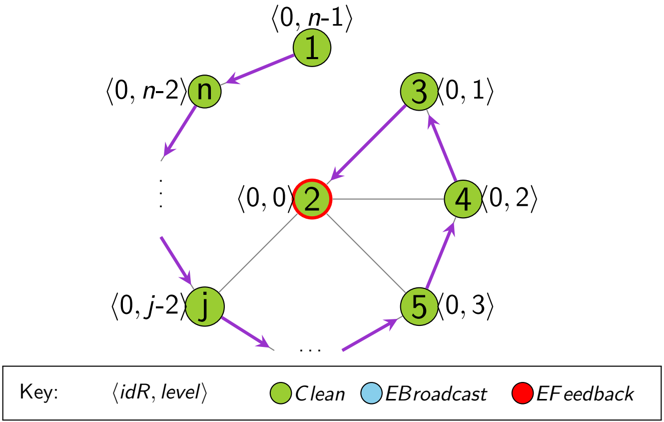
Stabilization Time in Rounds

- No alive abnormal tree created
- Height of an abnormal tree: at most n
- **Cleaning:**
  - EB-wave : n
  - EF-wave : n
  - R-wave : n
- **Building of the Spanning Tree:** $\mathcal{D}$
- **Stabilization Time:** $O(3n + \mathcal{D})$ **rounds**

$n =$ number of nodes ; $\mathcal{D} =$ diameter

Lower Bound on the Worst Case Stabilization Time in Rounds (synchronous execution)

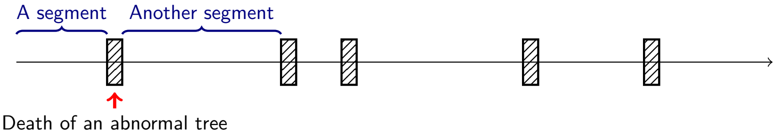
- path (1 - n - ...2)
- $+\ k$ links from 2
- $j = k + 3$
- $\mathcal{D} = (n{+}1{-}j){+}2$
  $= n - k$



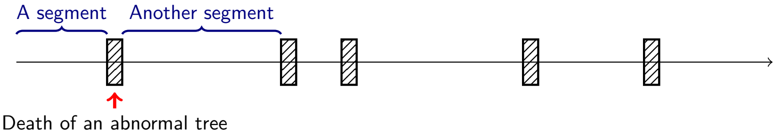
$n + n + n + (n - k)$

$= \text{exactly } 3n + \mathcal{D} \text{ rounds}$

Stabilization Time in Steps



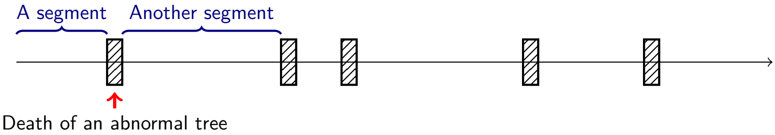
Stabilization Time in Steps



At most n alive abnormal trees $+$ No alive abnormal tree created

$\longrightarrow$ At most $n+1$ segments

Stabilization Time in Steps



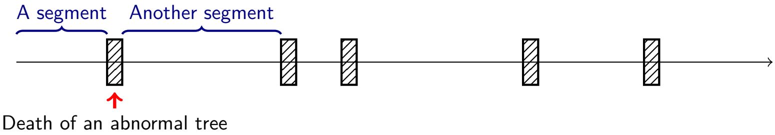
At most n alive abnormal trees $+$ No alive abnormal tree created

$\longrightarrow$ At most $n+1$ segments

In a segment, in a process

$$idR : 7 \xrightarrow{J\text{-action}} 5 \xrightarrow{J\text{-action}} 3 \xrightarrow{J\text{-action}} 2 \underbrace{\xrightarrow{EB\text{-action}} \xrightarrow{EF\text{-action}} \xrightarrow{R\text{-action}}}_{\text{Death of an abnormal tree} = \text{End of the segment}} 7 \xrightarrow{J\text{-action}} 3$$

Stabilization Time in Steps



At most n alive abnormal trees $+$ No alive abnormal tree created

$\longrightarrow$ At most $n+1$ segments

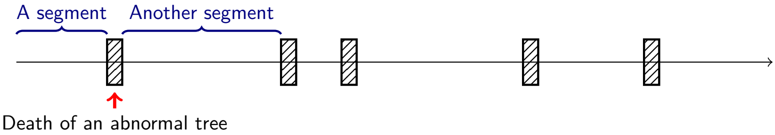
In a segment, in a process

$$idR : 7 \xrightarrow{J\text{-action}} 5 \xrightarrow{J\text{-action}} 3 \xrightarrow{J\text{-action}} 2 \underbrace{\xrightarrow{EB\text{-action}} \xrightarrow{EF\text{-action}} \xrightarrow{R\text{-action}}}_{\text{Death of an abnormal tree = End of the segment}} 7 \xrightarrow{J\text{-action}} 3$$

- $n-1$ J-actions
- 1 EB-action
- 1 EF-action
- 1 R-action

$\Rightarrow O(n)$ actions per process

Stabilization Time in Steps



At most n alive abnormal trees $+$ No alive abnormal tree created

$\longrightarrow$ At most $n+1$ segments

In a segment, in a process

$$idR : 7 \xrightarrow{J\text{-action}} 5 \xrightarrow{J\text{-action}} 3 \xrightarrow{J\text{-action}} 2 \underbrace{\xrightarrow{EB\text{-action}} \xrightarrow{EF\text{-action}} \xrightarrow{R\text{-action}}}_{\text{Death of an abnormal tree = End of the segment}} 7 \xrightarrow{J\text{-action}} 3$$

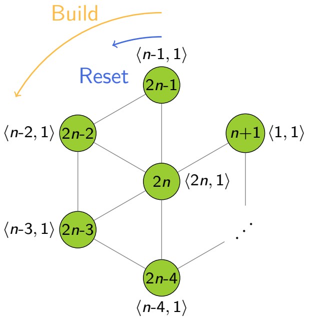
- $n-1$ *J*-actions
- 1 *EB*-action
- 1 *EF*-action
- 1 *R*-action

$\Rightarrow O(n)$ actions per process

$O(n^3)$ steps

Lower Bound: $\frac{n^3}{6} + \frac{3}{2}n^2 - \frac{8}{3}n + 2$ steps Upper Bound: $\frac{n^3}{2} + 2n^2 + \frac{n}{2} + 1$ steps

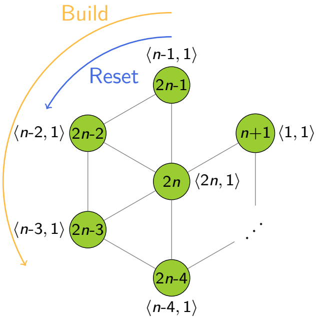
Lower Bound on the Worst Case Stabilization Time in Steps



Key: $\langle idR, level \rangle$

$Clean$ $EBroadcast$ $EFeedback$

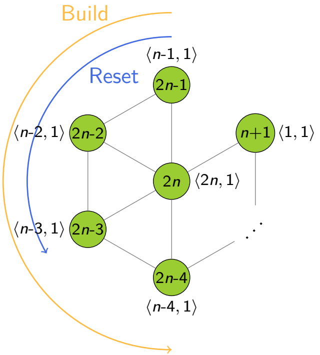
Lower Bound on the Worst Case Stabilization Time in Steps



Key: ⟨*idR*, *level*⟩

Clean *EBroadcast* *EFeedback*

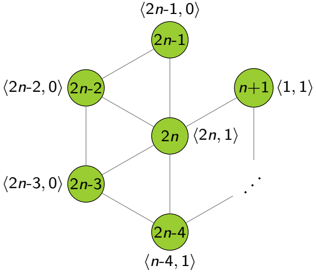
Lower Bound on the Worst Case Stabilization Time in Steps



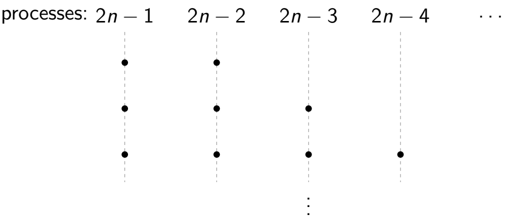
Key: ⟨*idR*, *level*⟩

Clean *EBroadcast* *EFeedback*

Lower Bound on the Worst Case Stabilization Time in Steps



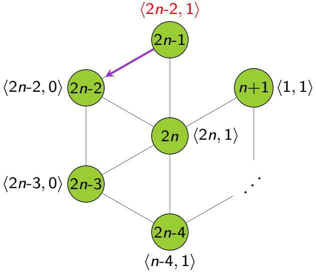
Case of the building on $2n-4$



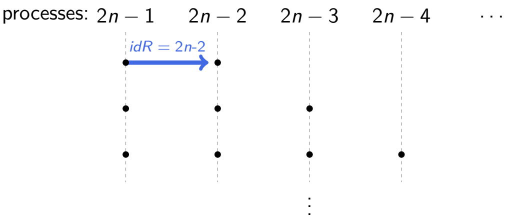
Key: $\langle idR, level\rangle$

Clean *EBroadcast* *EFeedback*

Lower Bound on the Worst Case Stabilization Time in Steps



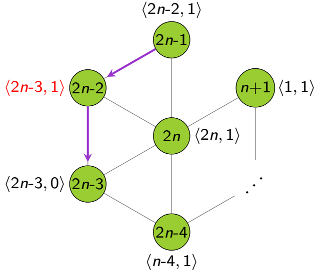
Case of the building on $2n-4$



Key: $\langle idR, level \rangle$

Clean *EBroadcast* *EFeedback*

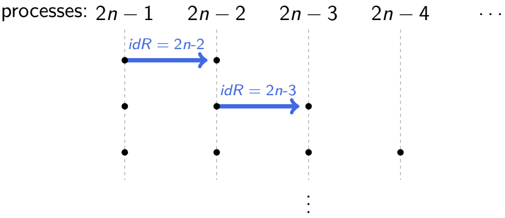
Lower Bound on the Worst Case Stabilization Time in Steps



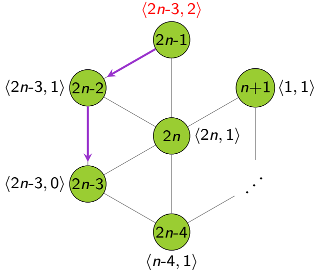
Key: $\langle idR, level \rangle$

Clean *EBroadcast* *EFeedback*

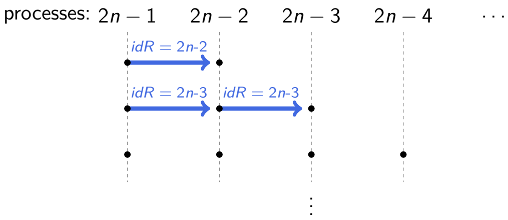
Case of the building on $2n-4$



Lower Bound on the Worst Case Stabilization Time in Steps



Case of the building on $2n-4$

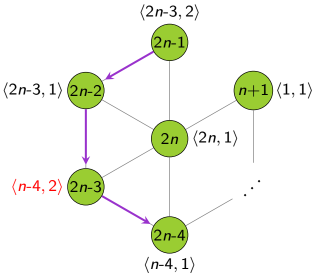


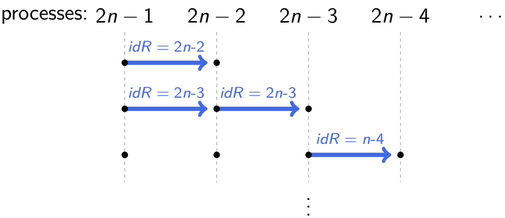
Key: $\langle idR, level\rangle$

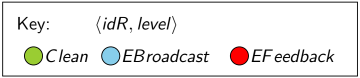
Clean *EBroadcast* *EFeedback*

Lower Bound on the Worst Case Stabilization Time in Steps

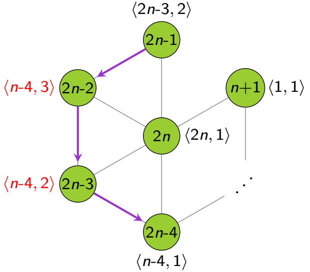
Case of the building on $2n-4$



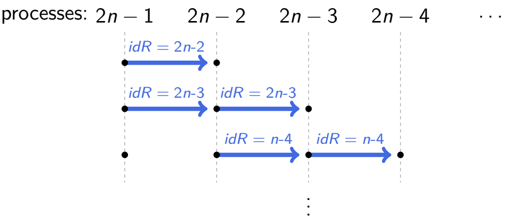




Lower Bound on the Worst Case Stabilization Time in Steps



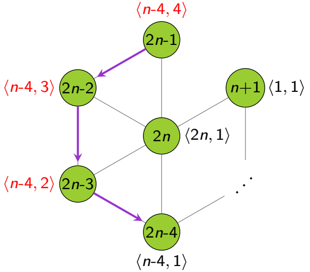
Case of the building on $2n-4$



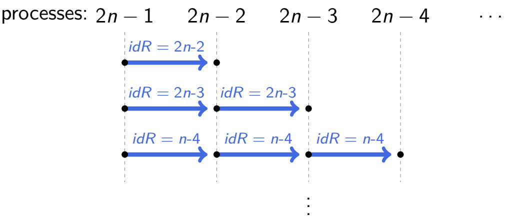
Key: $\langle idR, level \rangle$

Clean *EBroadcast* *EFeedback*

Lower Bound on the Worst Case Stabilization Time in Steps



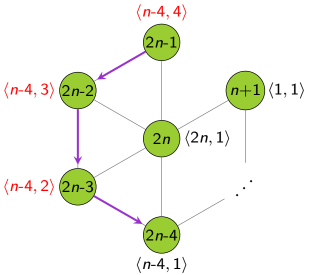
Case of the building on $2n-4$



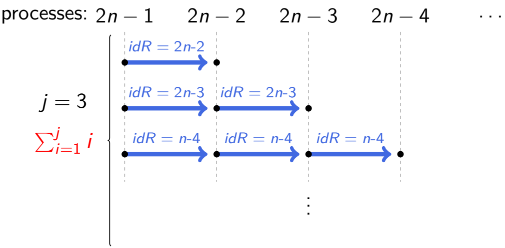
Key: $\langle idR, level \rangle$

Clean *EBroadcast* *EFeedback*

Lower Bound on the Worst Case Stabilization Time in Steps



Case of the building on $2n - 4$



Key: $\langle idR, level \rangle$

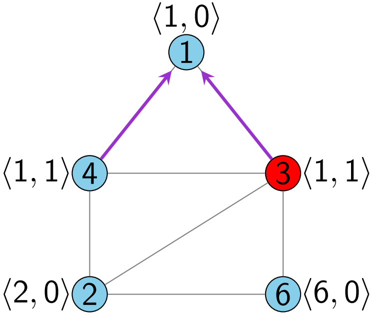
Clean *EBroadcast* *EFeedback*

$$\Theta(n) \text{ reset } \Rightarrow \sum_{j=1}^{n} \sum_{i=1}^{j} i \Rightarrow \Theta(n^3) \text{ steps}$$

Analytical Study of Datta *et al*, Self-stabilizing Leader Election in Optimal Space under an Arbitrary Scheduler. 2011

Principles

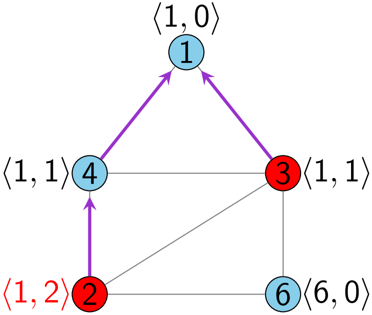
Join a tree



Key: ⟨*idR*, *level*⟩ Can be joined Cannot be joined

Principles

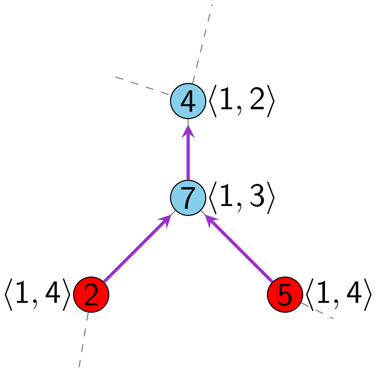
Join a tree



Key:	⟨*idR*, *level*⟩	Can be joined	Cannot be joined

Principles

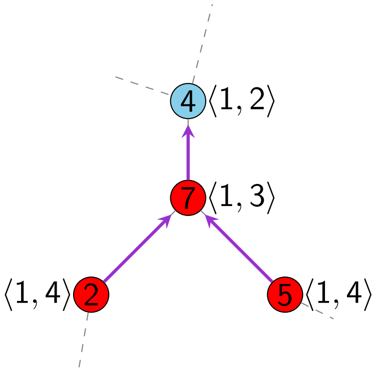
Change of color



Key: ⟨*idR*, *level*⟩ Can be joined Cannot be joined

Principles

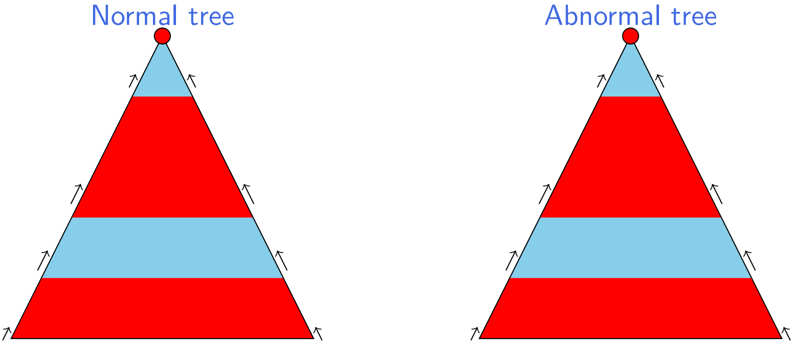
Change of color



Key: $\langle idR, level \rangle$ Can be joined Cannot be joined

Principles

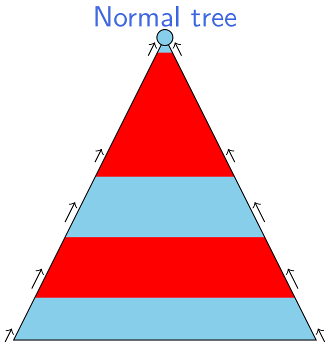
Color Waves Absorption

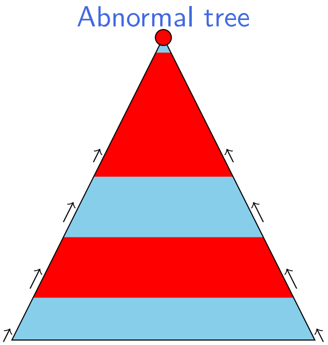


Key:	$\langle idR, level \rangle$	Can be joined	Cannot be joined

Principles

Color Waves Absorption

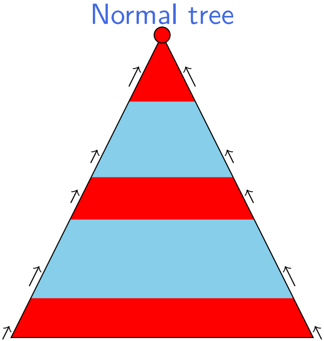


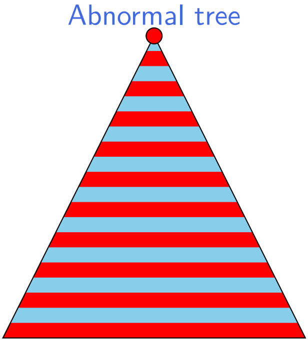


Key: $\langle idR, level \rangle$ Can be joined Cannot be joined

Principles

Color Waves Absorption





Key:	$\langle idR, level \rangle$	Can be joined	Cannot be joined

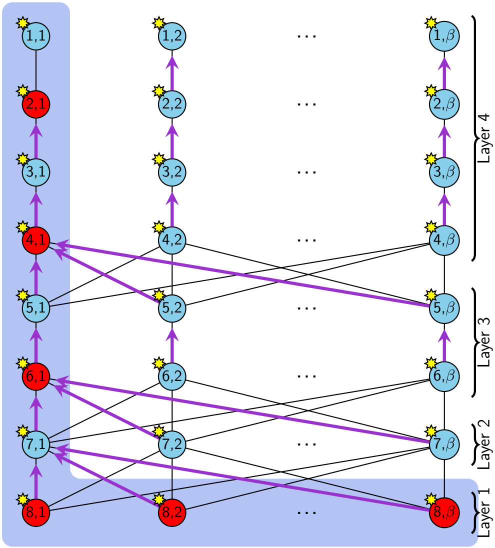
Datta et al, 2011

Execution in $\Omega(n^4)$ steps: $\beta = \frac{n}{8}$

Layer 1 resets

β

Key:
$(i,j).ID = (i-1)\beta + j$
$(i,j).idR = 0$
Can be joined
Cannot be joined

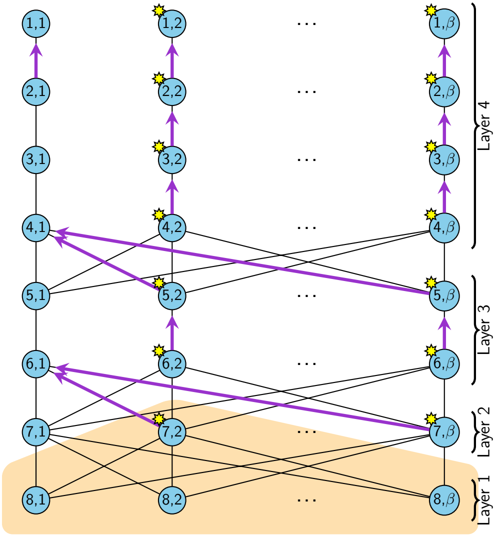


Datta et al, 2011

Execution in $\Omega(n^4)$ steps: $\beta = \frac{n}{8}$

Layer 1 joins (7,2)

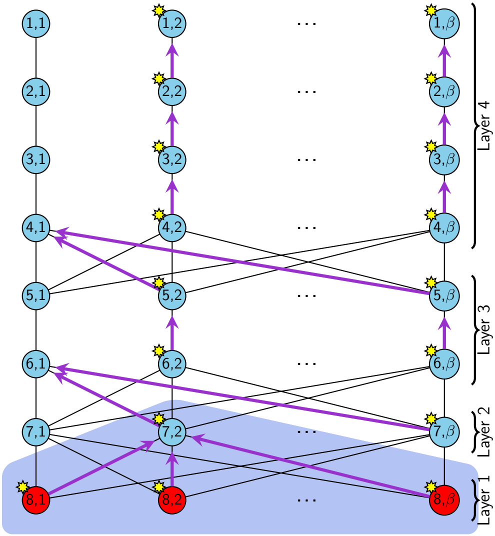
Key:
$(i,j).ID = (i-1)\beta + j$
$(i,j).idR = 0$
Can be joined
Cannot be joined



Datta et al, 2011

Execution in $\Omega(n^4)$ steps: $\beta = \frac{n}{8}$

Abnormal tree rooted at (7,2) resets



Key:

$(i, j).ID = (i-1)\beta + j$

$(i, j).idR = 0$

Can be joined

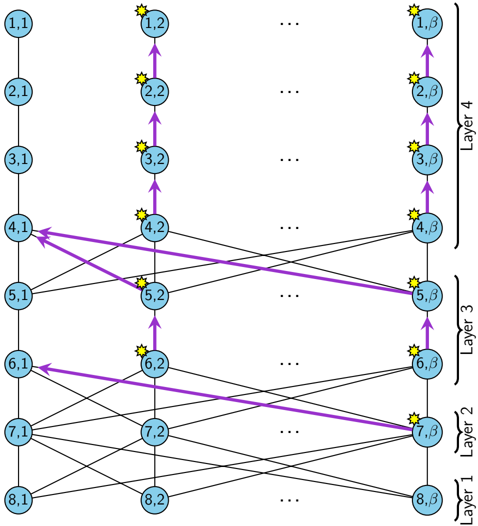
Cannot be joined

Datta et al, 2011

Execution in $\Omega(n^4)$ steps: $\beta = \frac{n}{8}$

Repeat until root (7,β)

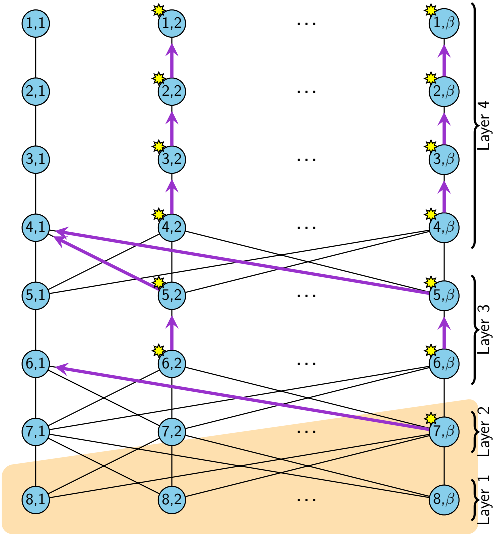
Key:
$(i, j).ID = (i-1)\beta + j$
$(i, j).idR = 0$
Can be joined
Cannot be joined



Datta et al, 2011

Execution in $\Omega(n^4)$ steps: $\beta = \frac{n}{8}$

Layer 1 joins (7,β)



Key:
$(i,j).ID = (i-1)\beta + j$
$(i,j).idR = 0$
Can be joined
Cannot be joined

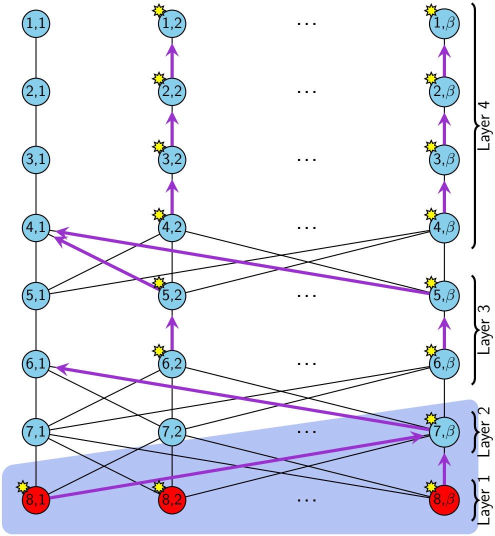
Datta et al, 2011

Execution in $\Omega(n^4)$ steps: $\beta = \frac{n}{8}$

Abnormal tree rooted at $(7,\beta)$ resets

β^2

Key:
$(i,j).ID = (i-1)\beta + j$
$(i,j).idR = 0$
Can be joined
Cannot be joined

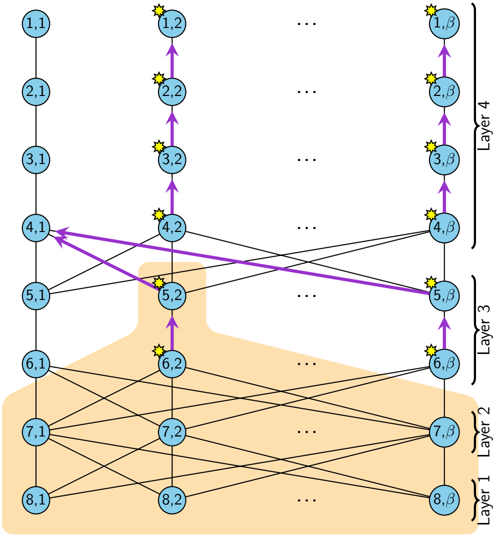


Datta et al, 2011

Execution in $\Omega(n^4)$ steps: $\beta = \frac{n}{8}$

Layers 1 and 2 join (5,2)

Key:
$(i, j).ID = (i-1)\beta + j$
$(i, j).idR = 0$
Can be joined
Cannot be joined

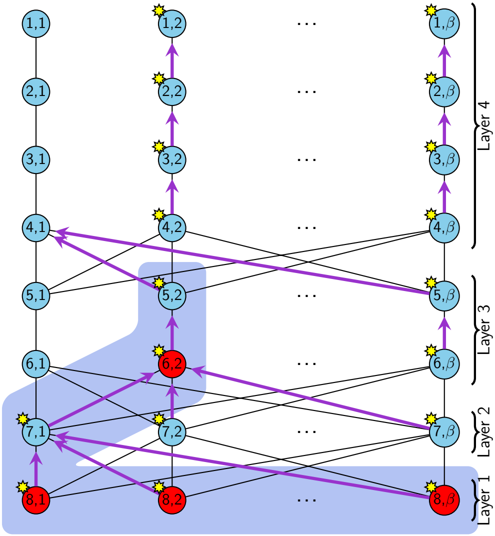


Datta et al, 2011

Execution in $\Omega(n^4)$ steps: $\beta = \frac{n}{8}$

Abnormal tree rooted at (5,2) resets

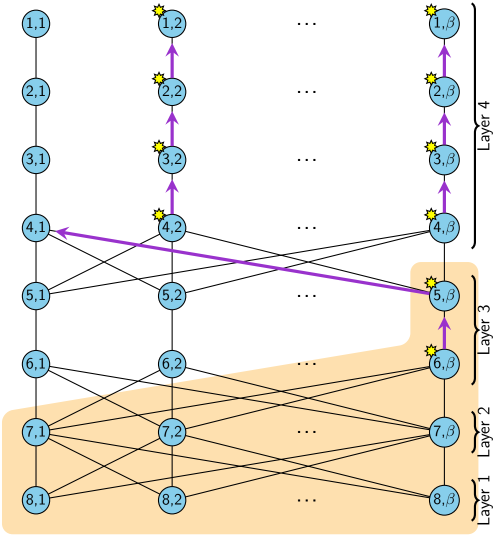
Key:
$(i, j).ID = (i-1)\beta + j$
$(i, j).idR = 0$
Can be joined
Cannot be joined



Datta et al, 2011

Execution in $\Omega(n^4)$ steps: $\beta = \frac{n}{8}$

Layers 1 and 2 join $(5,\beta)$



Key:
$(i,j).ID = (i-1)\beta + j$
$(i,j).idR = 0$
Can be joined
Cannot be joined

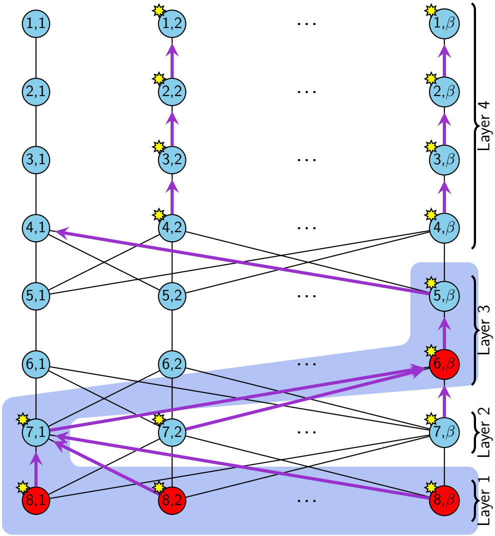
Datta et al, 2011

Execution in $\Omega(n^4)$ steps: $\beta = \frac{n}{8}$

Abnormal tree rooted at (5,β) resets

β^3

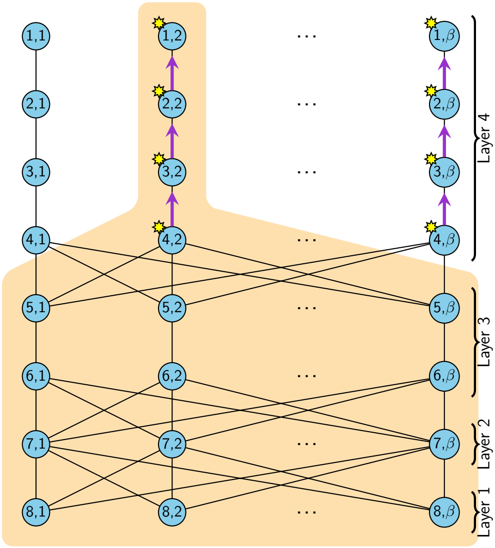
Key:
$(i,j).ID = (i-1)\beta + j$
✸ $(i,j).idR = 0$
● Can be joined
● Cannot be joined



Datta et al, 2011

Execution in $\Omega(n^4)$ steps: $\beta = \frac{n}{8}$

Layers 1,2 and 3 join (1,2)



Key:

$(i,j).ID = (i-1)\beta + j$

$(i,j).idR = 0$

Can be joined

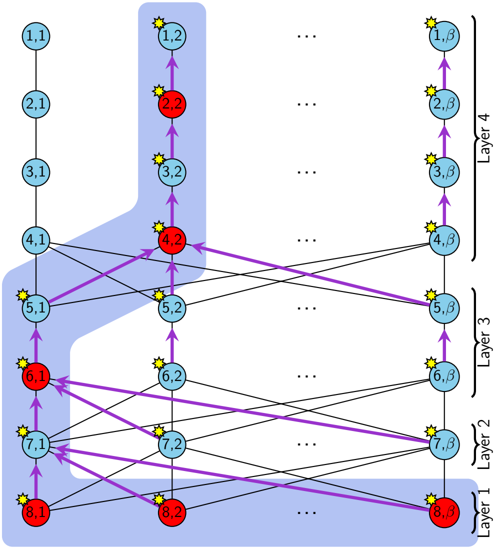
Cannot be joined

Datta et al, 2011

Execution in $\Omega(n^4)$ steps: $\beta = \frac{n}{8}$

Abnormal tree rooted at (1,2) resets

Key:
$(i, j).ID = (i - 1)\beta + j$
$(i, j).idR = 0$
Can be joined
Cannot be joined



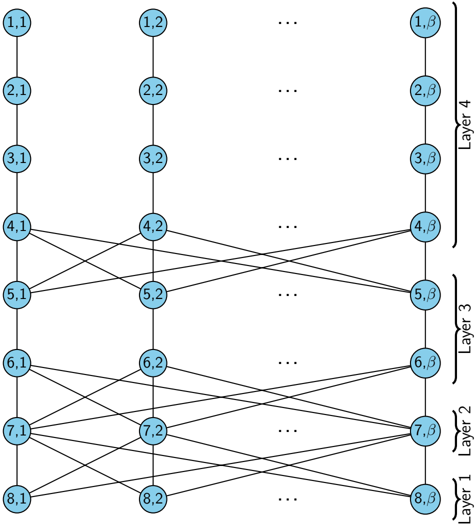
Datta et al, 2011

Execution in $\Omega(n^4)$ steps: $\beta = \frac{n}{8}$

β^4

$$\beta = \Omega(n) \Rightarrow \Omega(n^4)$$

Key:
$(i,j).ID = (i-1)\beta + j$
$(i,j).idR = 0$
Can be joined
Cannot be joined

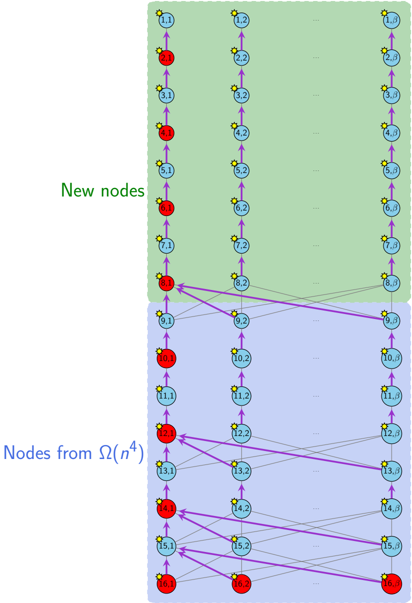


Datta et al, 2011

Network for $\Omega(n^5)$ steps

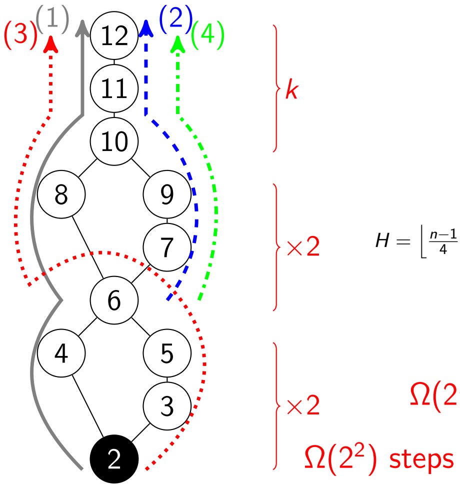
$\forall \alpha \geq 3$, $\exists$ networks and executions in $\Omega(n^{\alpha+1})$ steps.

Worst Case:
$\Omega\left((2n)^{\frac{1}{4}\log_2(2n)} \right)$ steps

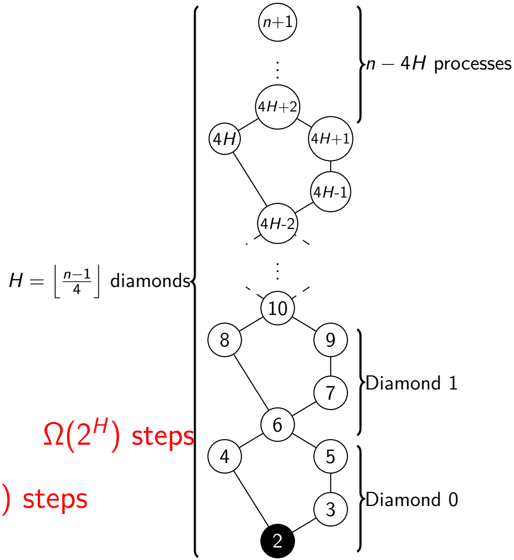


Analytical Study of Datta *et al*, An $O(n)$-time Self-stabilizing Leader Election Algorithm. 2011

Execution for $n = 11$



Network for $n \geq 5$



Perspectives

Goal

Design a self-stabilizing leader election algorithm that stabilizes in $O(\mathcal{D})$ rounds.

Hypotheses

- Unfair daemon
- Memory requirement of $\Theta(\log n)$ bits/process

- With the knowledge of $D \geq \mathcal{D}$, $(D = O(\mathcal{D}))$: ✓
- Without any global knowledge : **??**

Thank you for your attention.

Do you have any questions ?

Self-Stabilizing Leader Election in Polynomial Steps.

Karine Altisen, Alain Cournier, Stéphane Devismes, Anaïs Durand, Franck Petit