Workshop DUCAT/ESTATE — 16 mars 2022

ESTATE

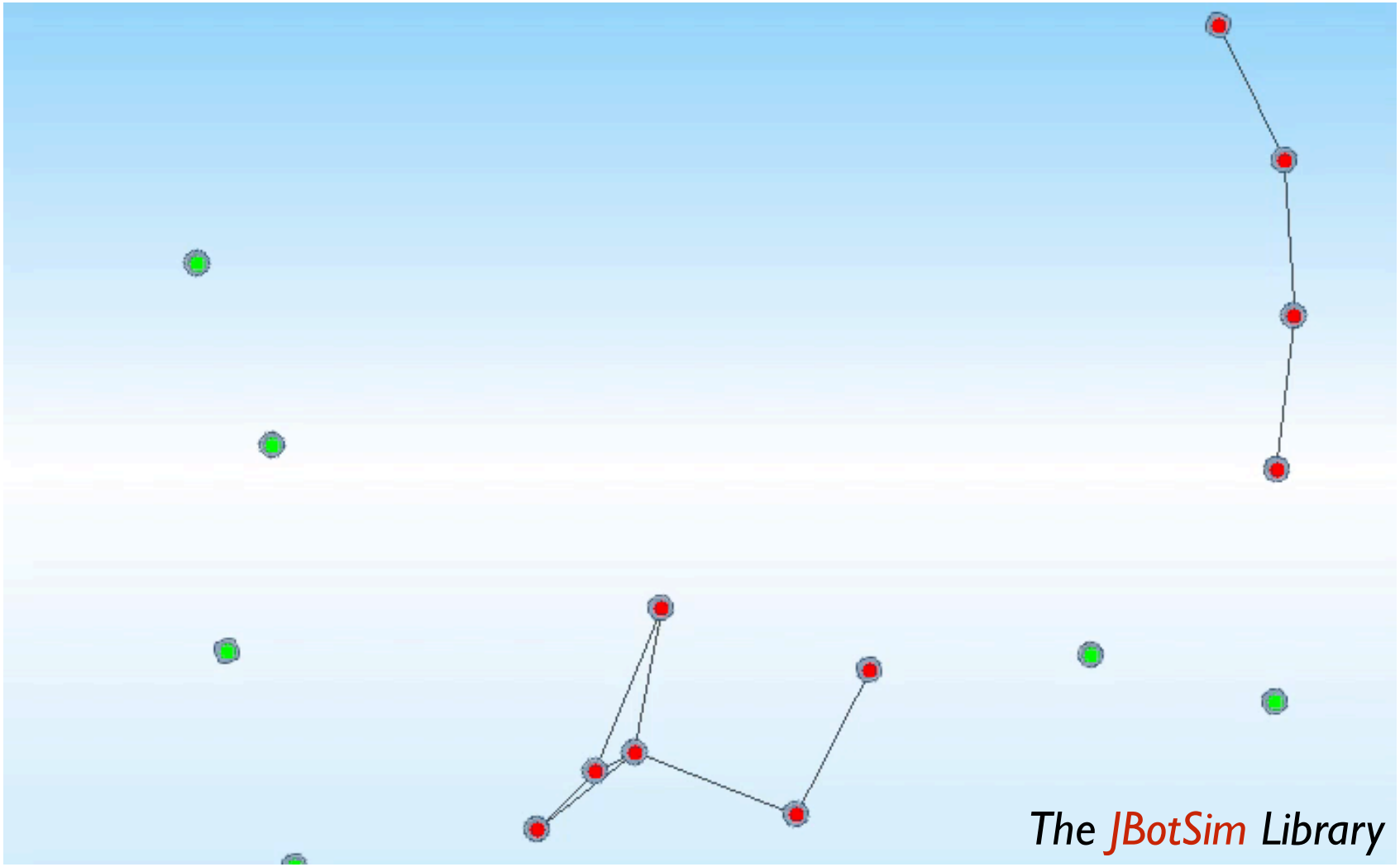
Enhancing Safety and self-sTAbilization in Time-varying distributed Environments

Franck Petit
Sorbonne University
LIP6 CNRS / INRIA

ESTATE

Enhancing Safety and self-sTAbilization in Time-varying distributed Environments

ESTATE



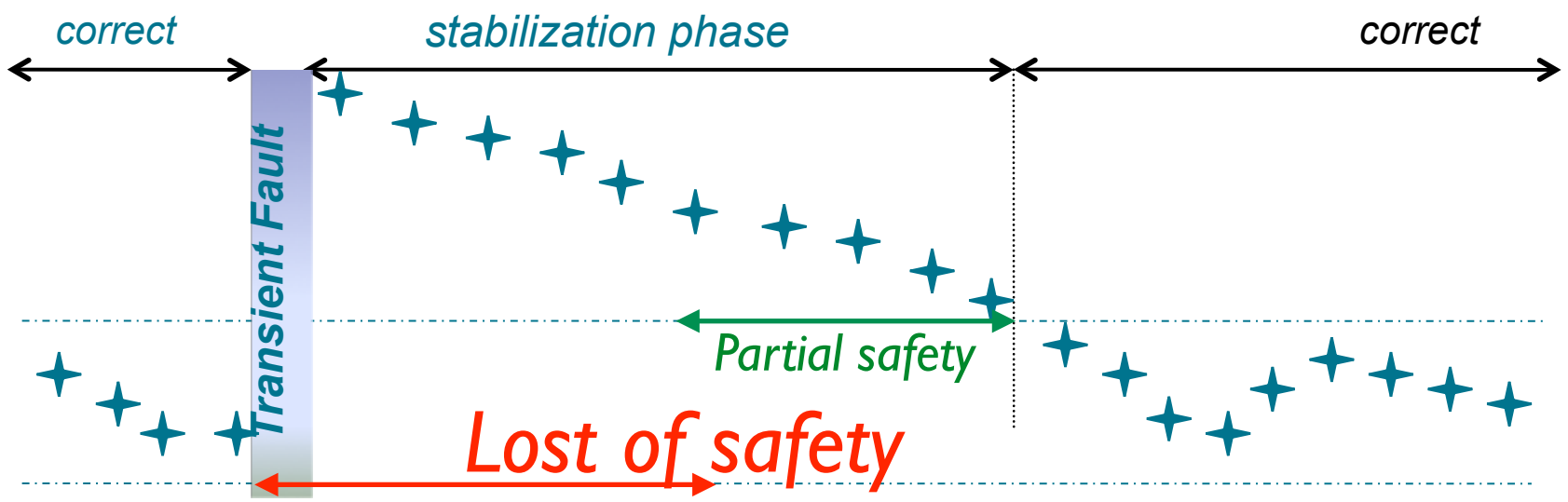
[A. Casteigts, R. Laplace, 2015]

ESTATE

Enhancing Safety and self-sTAbilization in Time-varying distributed Environments

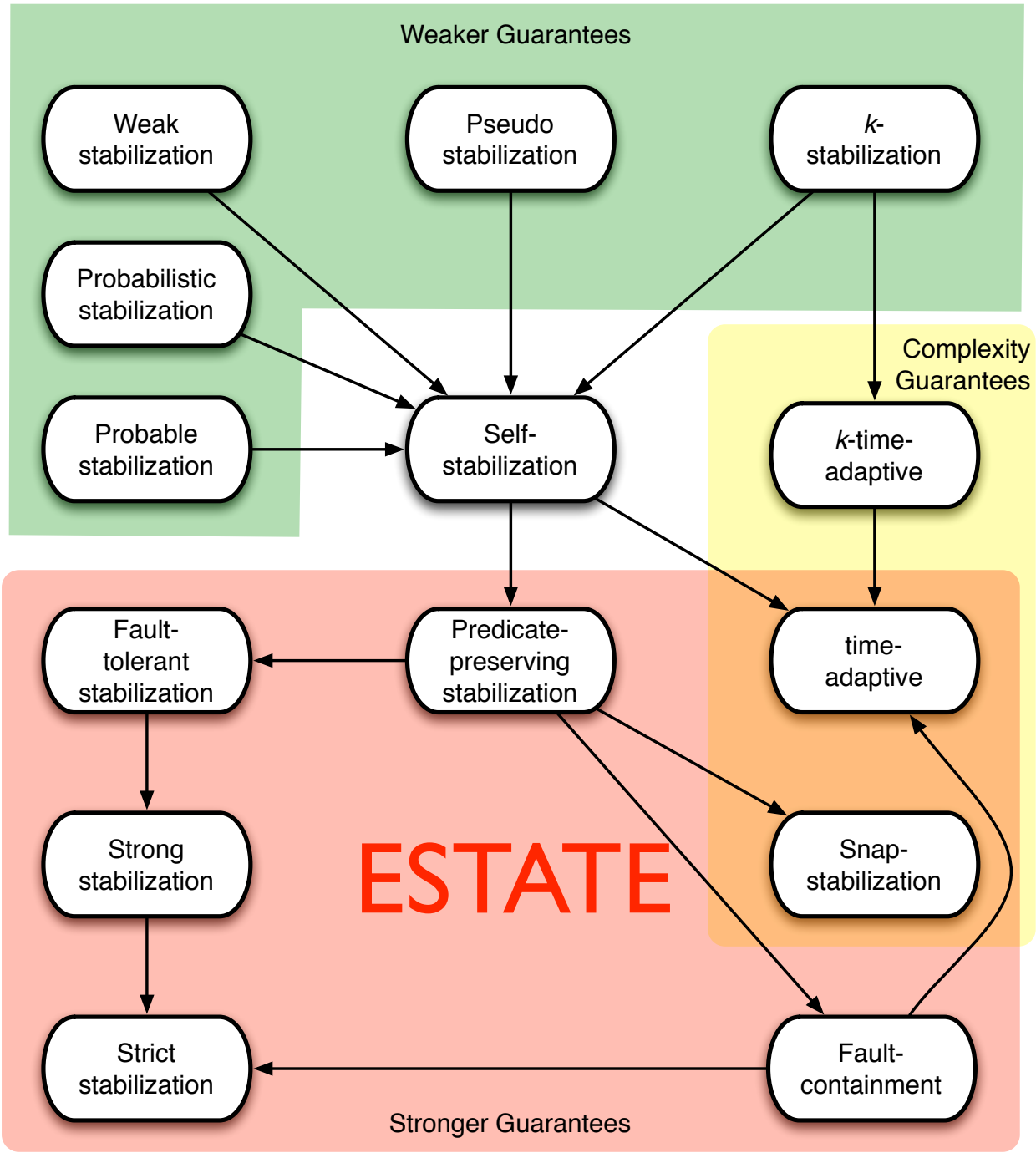
ESTATE

A self-stabilizing system, regardless of its initial state, is guaranteed to converge to the intended behavior in finite time.



Enhancing Safety and Self-stabilization

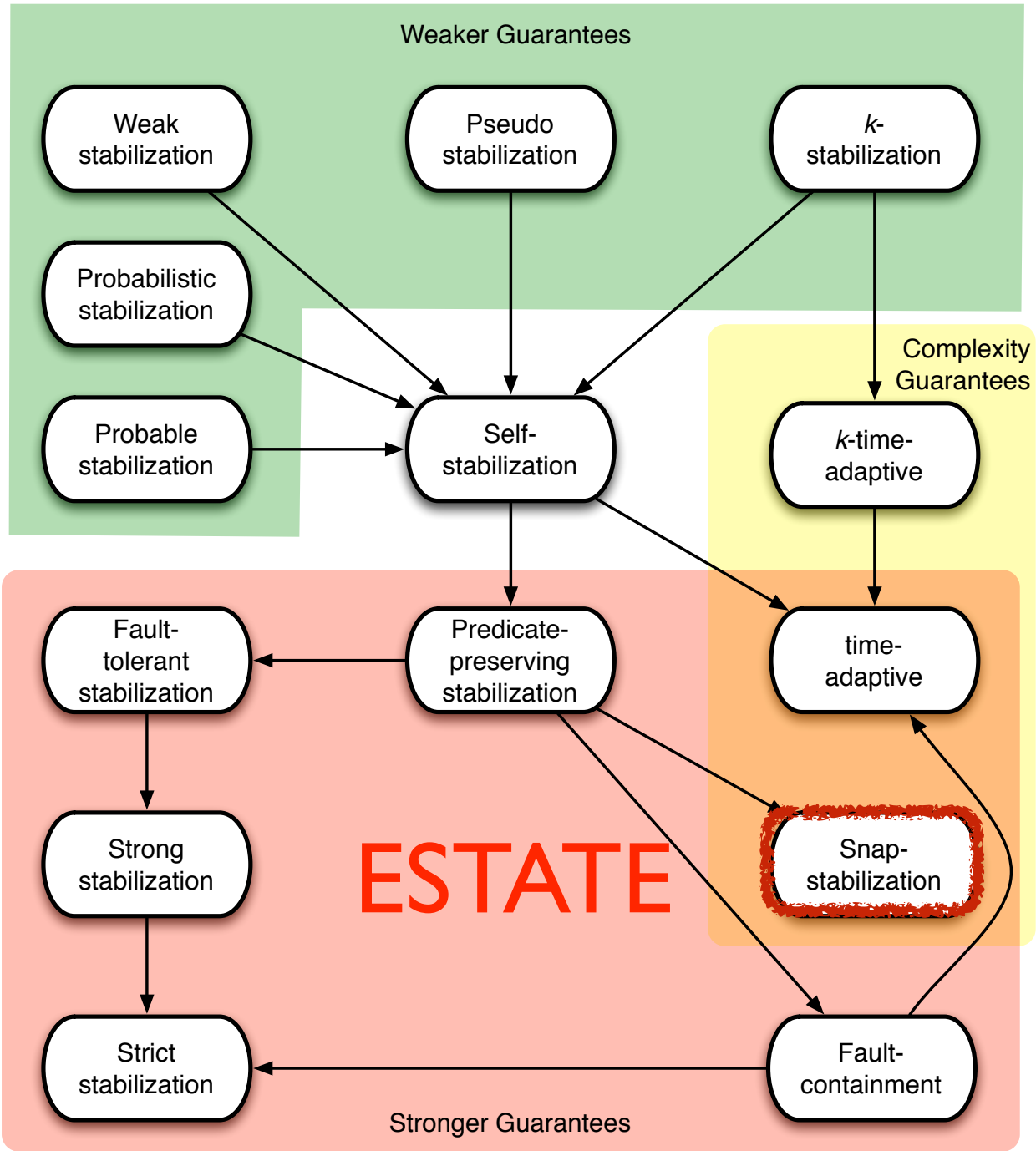
ESTATE



ESTATE

- Organization of 6 Events
  - Joint Workshop of ANR Projects DUCAT and ESTATE, Saint-Valery, March 15-18, 2022
  - Joint Workshop of ANR Projects DESCARTES and ESTATE, Fontainebleau, November 8-10 2021
  - Joint Workshop of ANR Projects DESCARTES and ESTATE, Saint-Valery, June 9-12 2020, **CANCELLED DUE TO COVID19 PANDEMIC**
  - Joint Workshop of ANR Projects DESCARTES and ESTATE, in conjonction with CoA 2019, Roscoff, April 1-5, 2019
  - Joint Workshop ESTATE/VERIMAG, Grenoble Aug. 31th, 2017
  - CoDyn, Second Workshop on Computing in Dynamic Networks, Vienna, October 20th, 2017, co-located with DISC'17
- 4 PhD Defenses, 2 HDR
  - PhD Defense of Jason Schoeters, Bordeaux March 29th, 2021
  - HDR Defense of Stéphane Devismes, Grenoble December, 17th, 2020
  - PhD Defense of Sébastien Bouchard, Paris September 26th, 2019
  - PhD Defense of Marjorie Bournat, Paris June 27th, 2019
  - HDR Defense of Arnaud Casteigts, Bordeaux June, 4th, 2018
  - PhD Defense of Anaïs Durand, Grenoble Sept. 1st, 2017
- 88 publications (43 conf. inter., 29 journaux, 16 conf. nat.)

ESTATE



Snap-Stabilizing Distributed System

Distributed System

- *It is difficult to get two computer scientists to agree on what a distributed system is.*

  [A. Tanenbaum, R. Van Renesse, ACM Computing Surveys, 1985]

- *…distributed algorithm, distributed program, networks…*

  [N. Lynch, J. Welch, 2004]

- ~~*Distributed system*~~ *Distributed Computing*:

  A distributed system is a system whose components are located on different networked computers, which communicate and coordinate their actions by passing messages to one another from any system.

  [Wikipedia, 2022]

Distributed System

- *It is difficult to get two computer scientists to agree on what a distributed system is.*

  [A. Tanenbaum, R. Van Renesse, ACM Computing Surveys, 1985]

- *A distributed system is a collection of individual computing devices that can communicate with each other.*

  [H. Attiya, J. Welch, 2004]

- *A distributed system is made up of a collection of distributed computing units, each one abstracted through the notion of process. The processes are assumed to cooperate on a common goal, which means that they exchange information in way or another.*

  [M. Raynal, 2018]

In a vacuum?

Distributed System

- *It is difficult to get two computer scientists to agree on what a distributed system is.*

  [A. Tanenbaum, R. Van Renesse, ACM Computing Surveys, 1985]

- […] *we use the term « distributed system » to mean a distributed operating system.* […]
  A distributed (operating) system is one that looks to its users like an ordinary centralized (operating) system but runs on multiple, independent central processing units (CPUs). The key concept here is transparency. In other words, the use of multiple processors should be invisible (transparent) to the user.

  [A. Tanenbaum, R. Van Renesse, ACM Computing Surveys, 1986]

Masking Fault-Tolerance?

Distributed System

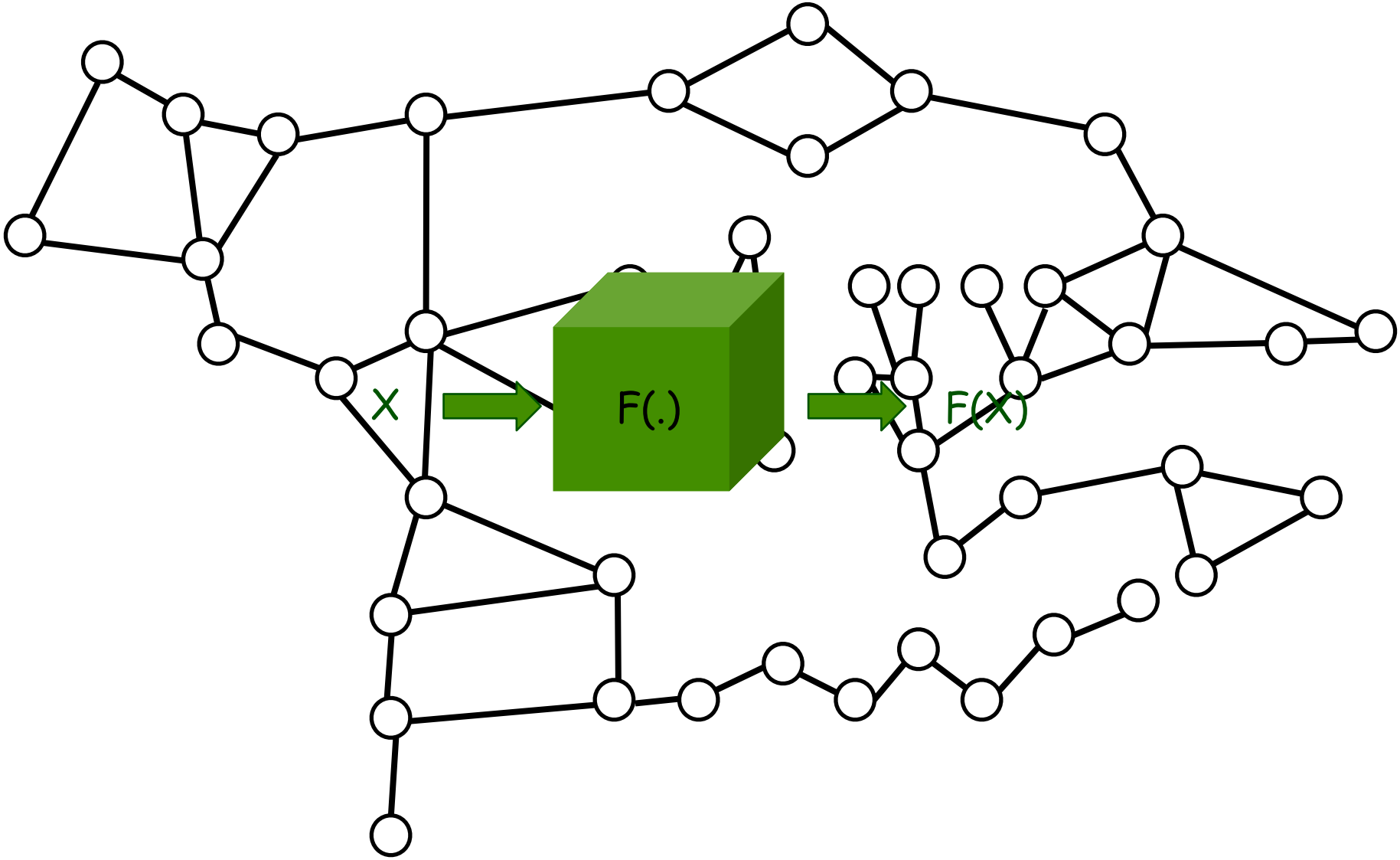
- *It is difficult to get two computer scientists to agree on what a distributed system is.*

  [A. Tanenbaum, R. Van Renesse, ACM Computing Surveys, 1985]

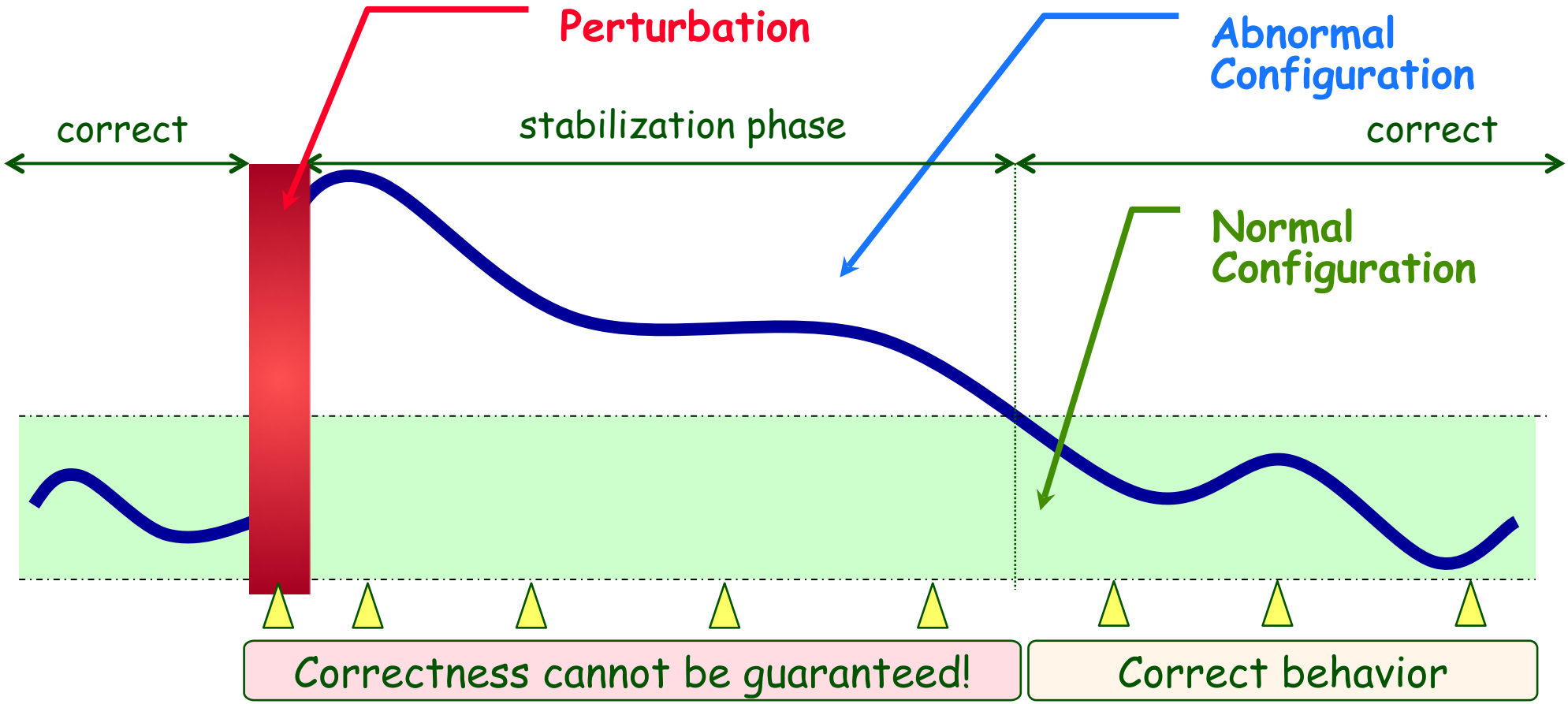
- *[…] one safe approach is to consider first what a centralized system is. […]*
  The most desirable situation in a distributed system is to be able to supply the user with a centralized view of the system, i.e., to make the distributed nature of the system transparent to the user, and let it act as though it is only use of the system, and the system is composed of a single entity […].
  [D. Peleg, 2000]

Distributed System

From the user's point of view…



Self-Stabilizing Distributed System

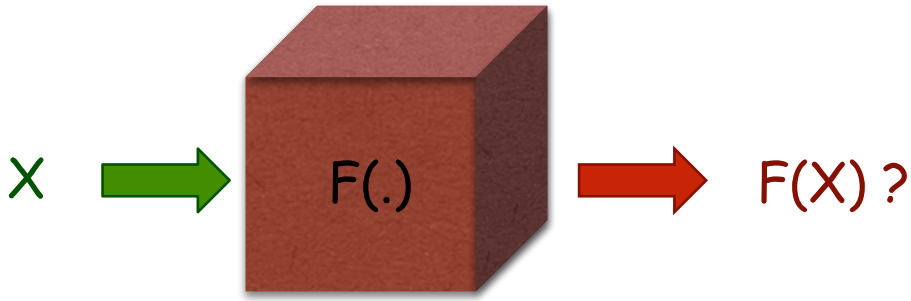


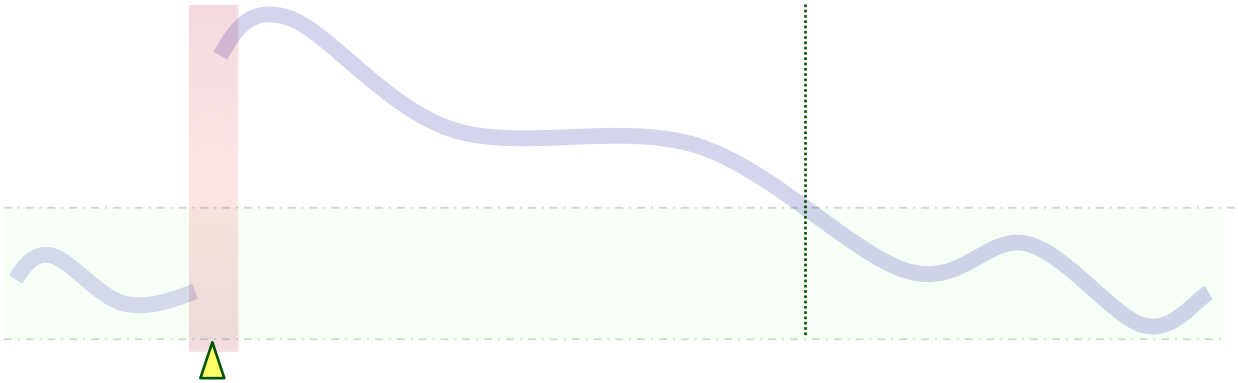
A self-stabilizing system, regardless of the initial state of the processors, is guaranteed to converge to the intended behavior in finite time.

Dijkstra 1974

Self-Stabilizing Distributed System

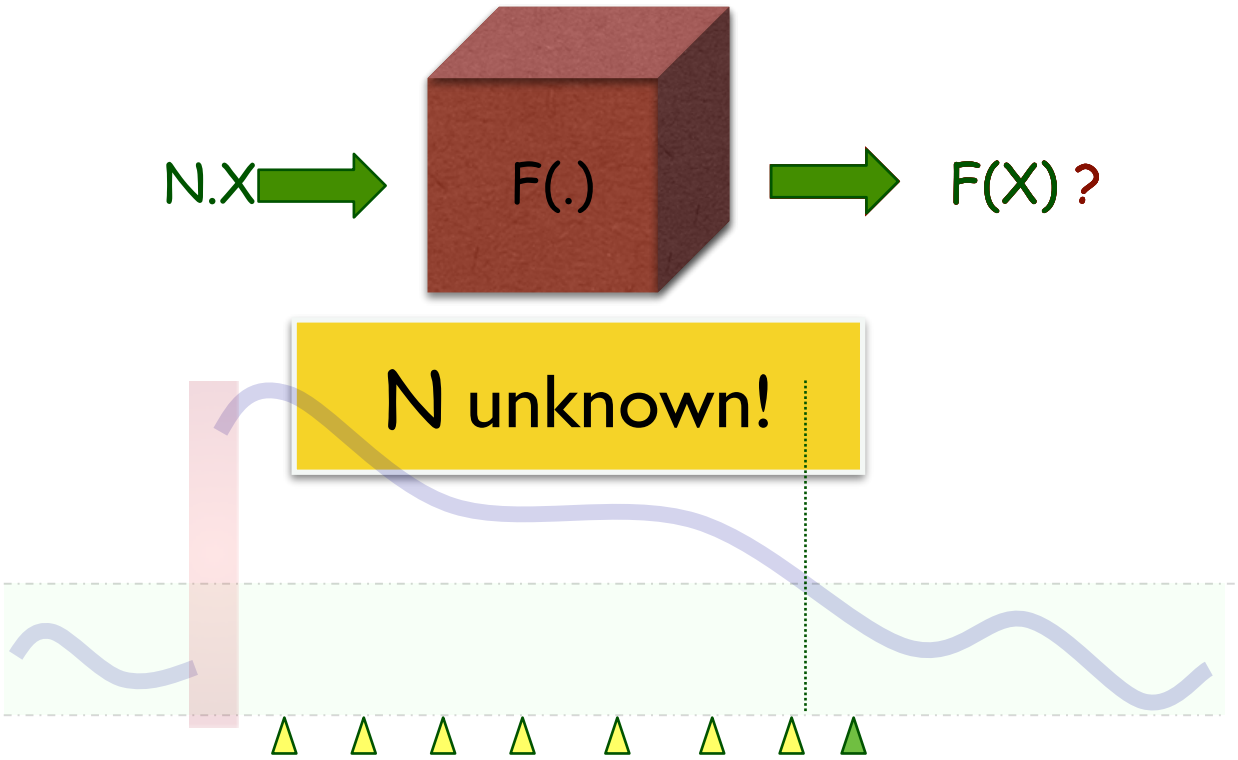
From the user's point of view…





Self-Stabilizing Distributed System

From the user's point of view…

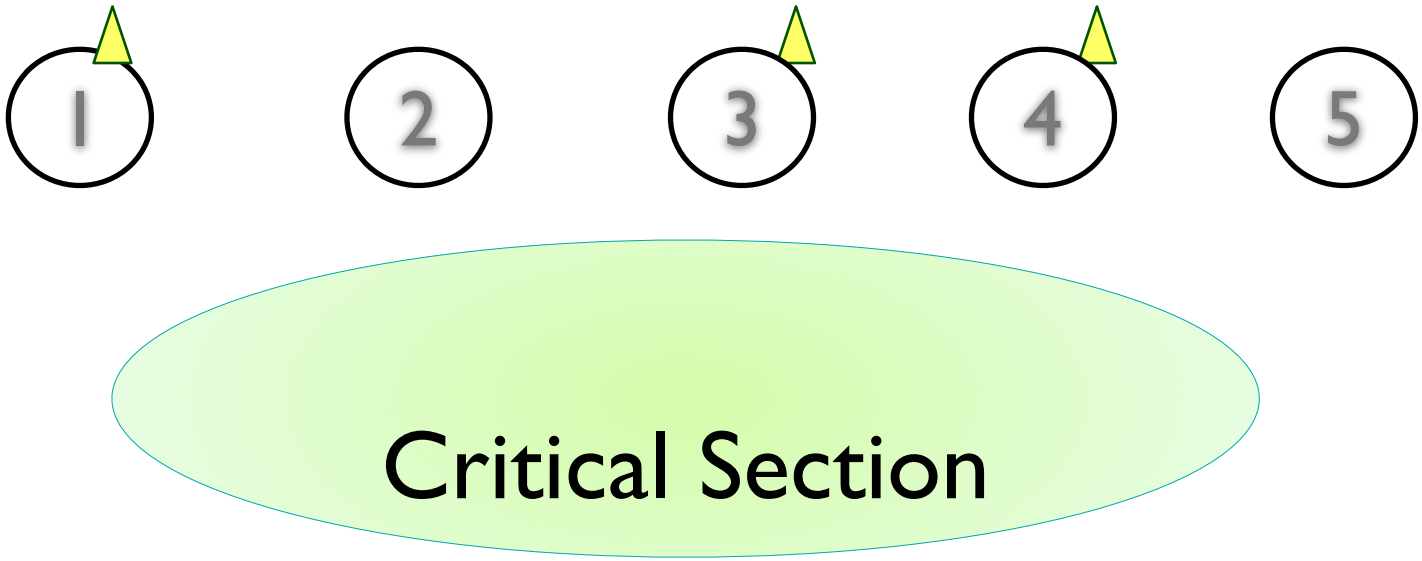


Self-Stabilization

Rough Approach

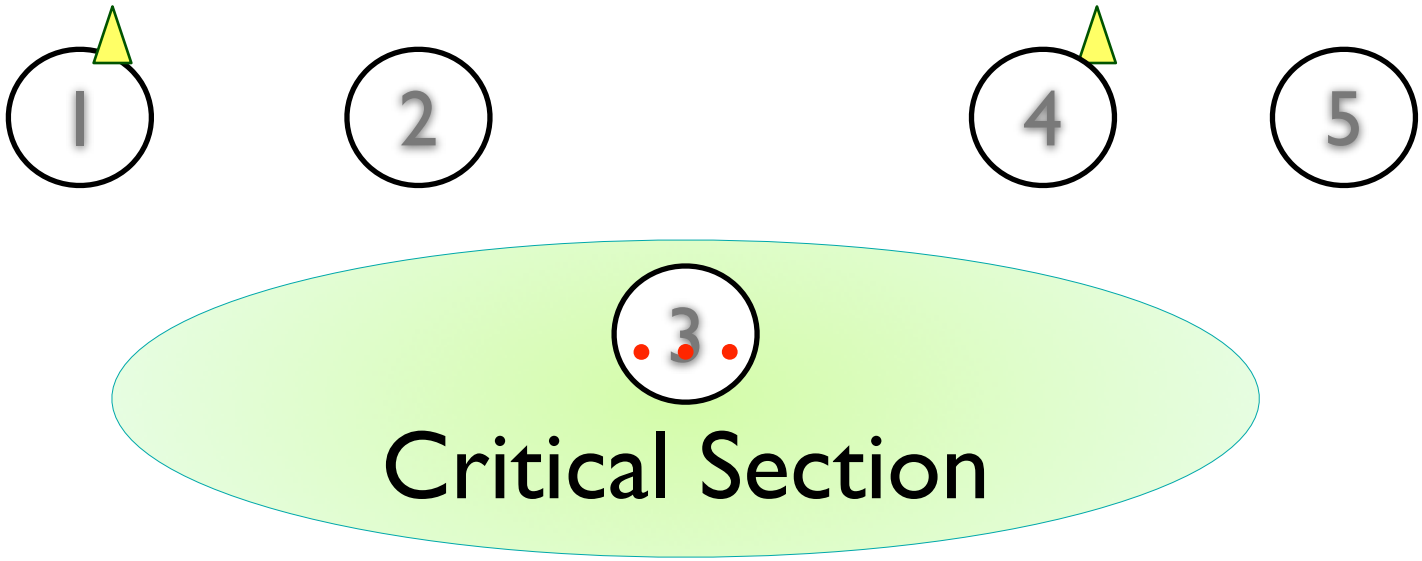
Mutual Exclusion

- Safety: No two processes execute the critical section simultaneously.
- Liveness: Upon a request, a process enters the critical section in finite time.



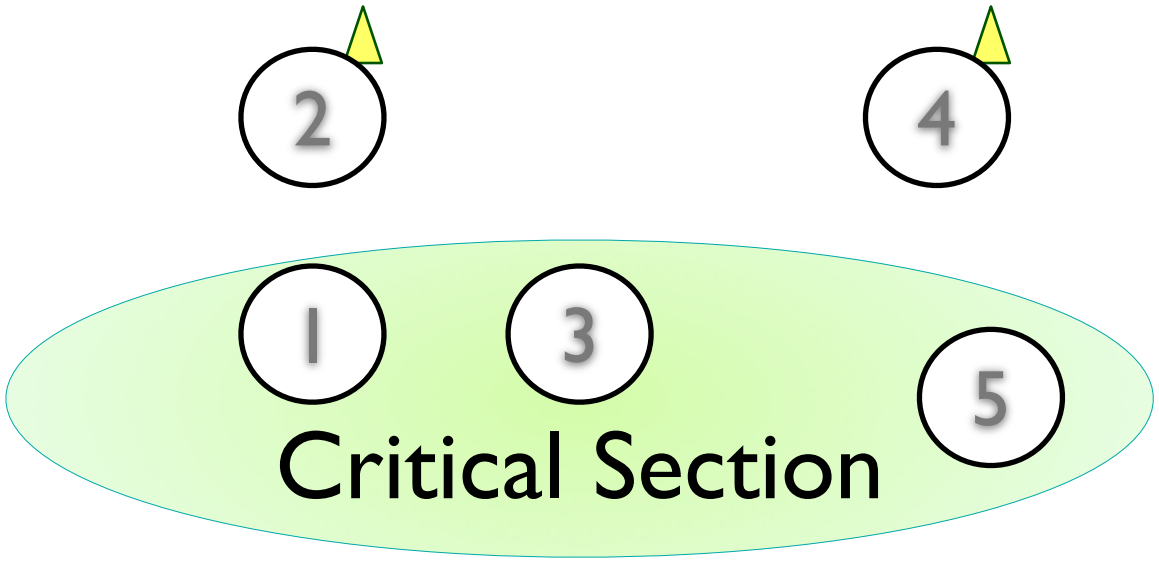
Mutual Exclusion

- Safety: No two processes execute the critical section simultaneously.
- Liveness: Upon a request, a process enters the critical section in finite time.



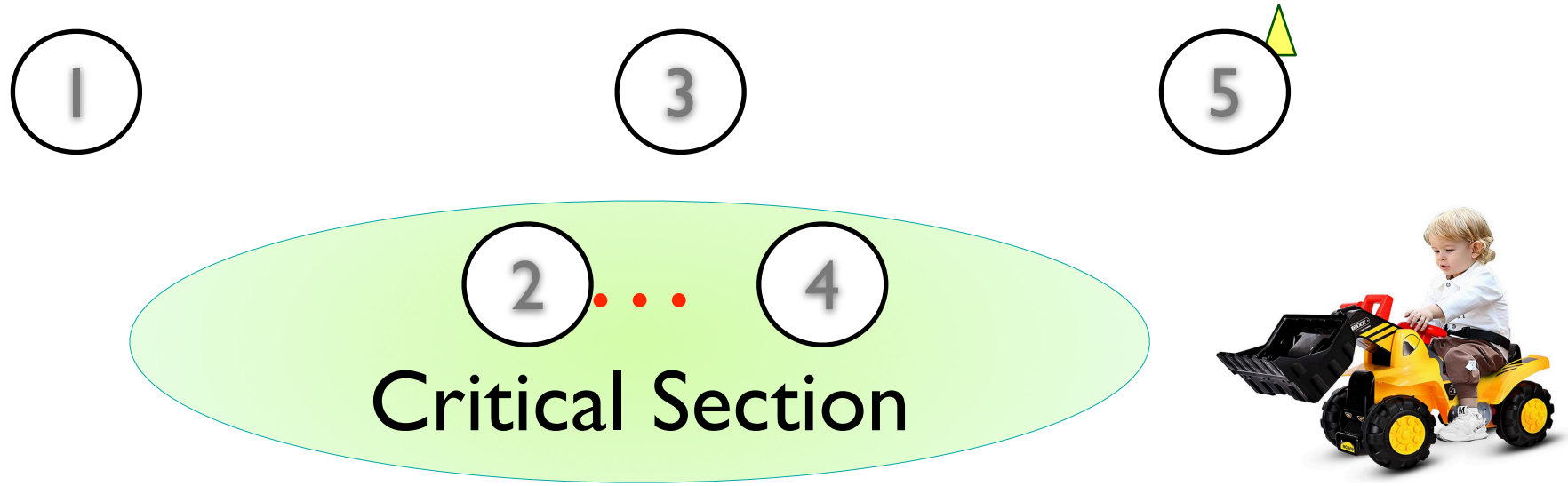
Self-Stabilizing Mutual Exclusion

- Safety: Eventually, no two processes execute the critical section simultaneously.
- Liveness: Upon a request, a process enters the critical section in finite time.

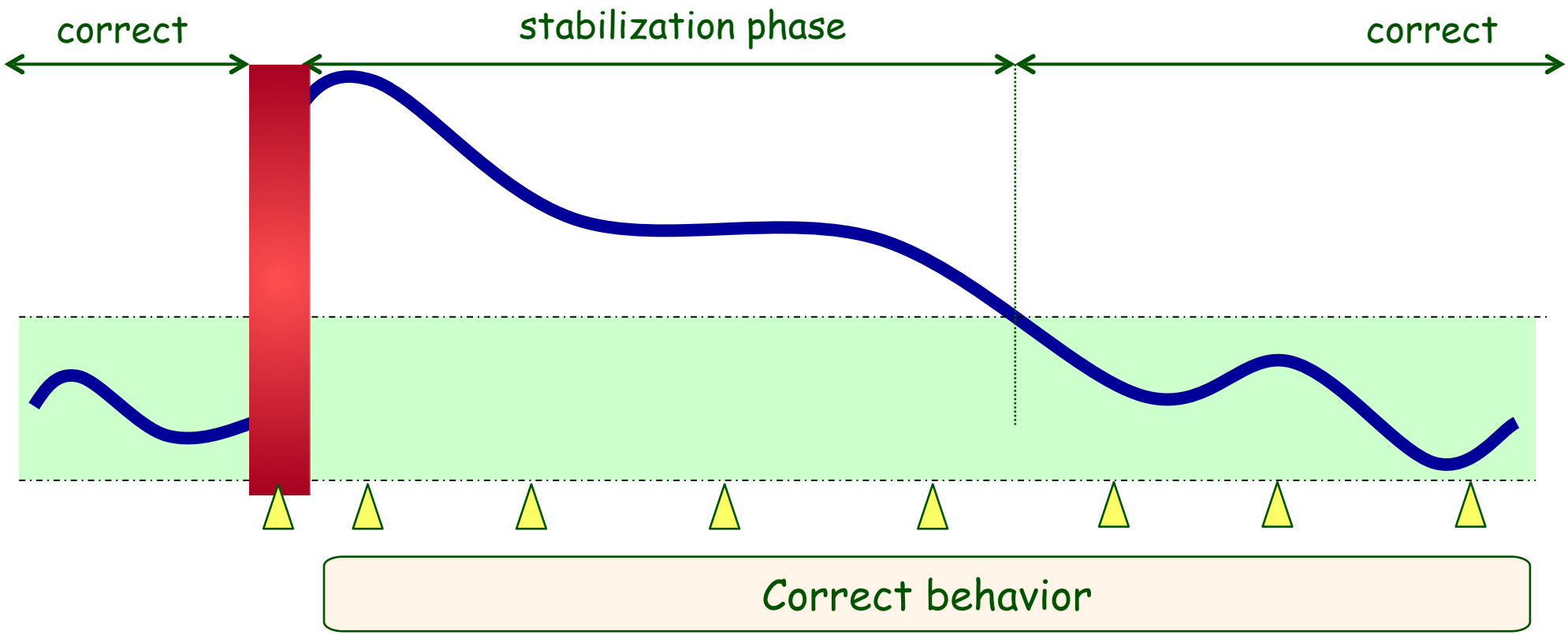


Self-Stabilizing Mutual Exclusion

- Safety: Eventually, no two processes execute the critical section simultaneously.
- Liveness: Upon a request, a process enters the critical section in finite time.



Snap-Stabilizing Distributed System

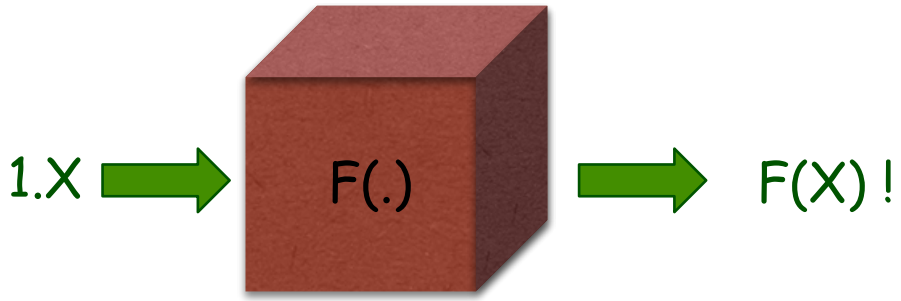


A snap-stabilizing system, regardless of the initial state of the processors, is guaranteed to converge to the intended behavior in 0 time.

Bui, Datta, Petit, and Villain 1999

Snap-Stabilizing Distributed System

From the user's point of view…

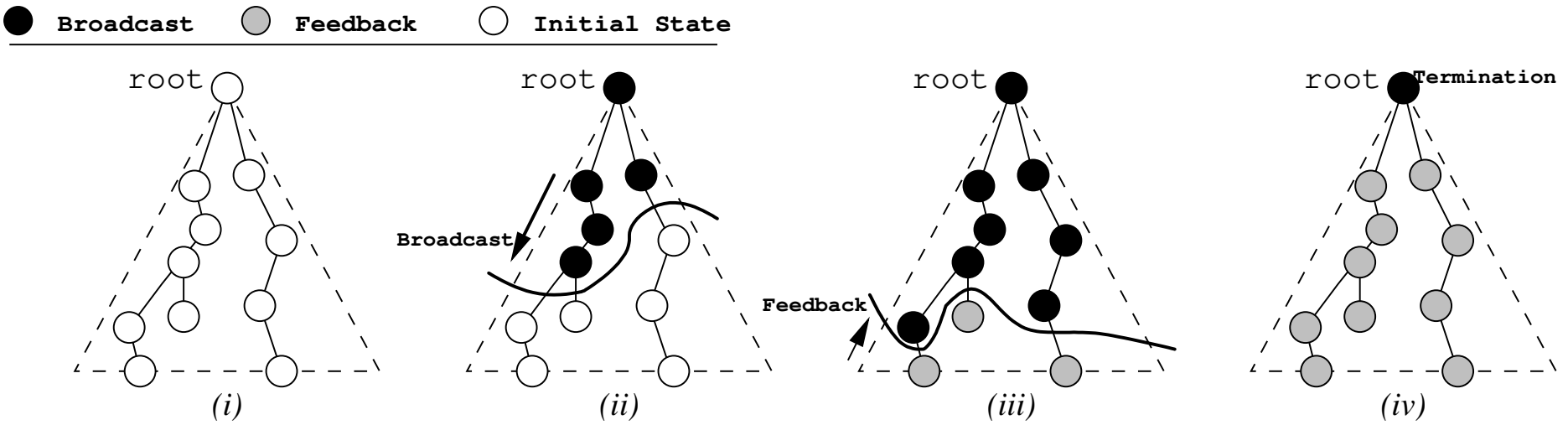


Snap-Stabilization

Cautious Approach

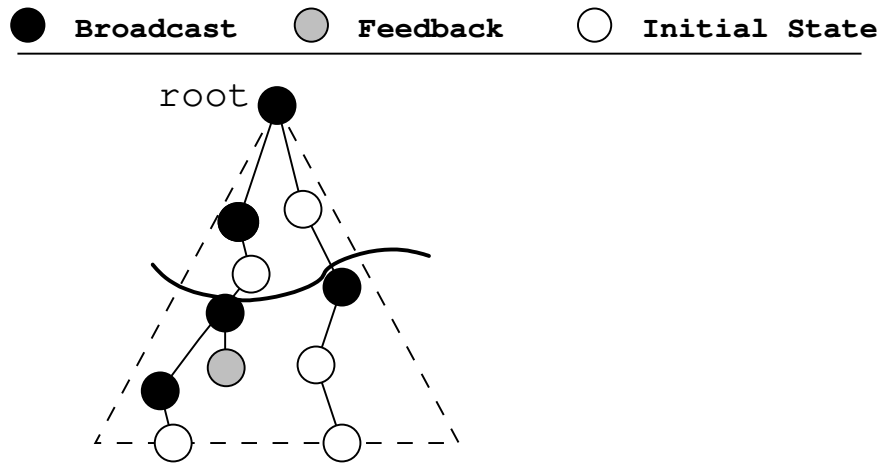
PIF

Propagation of Information with Feedback

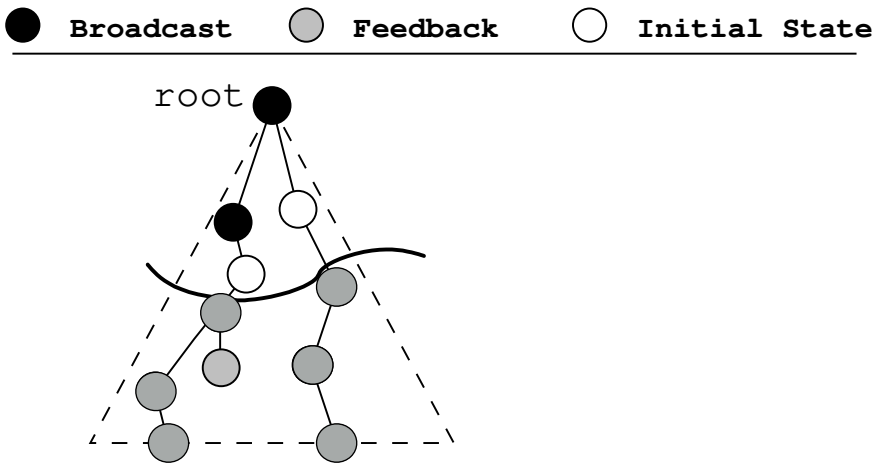


[*A Segall. IEEE Transactions on Information Theory, 1983*]

Self-Stabilizing PIF

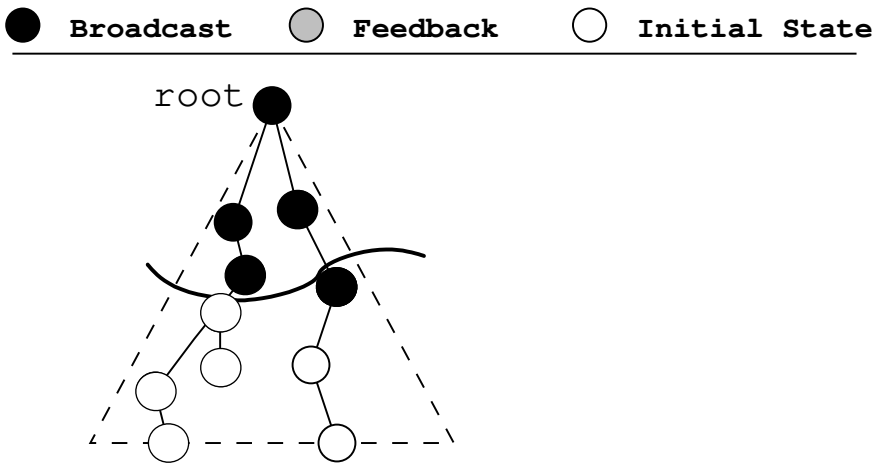


Snap-Stabilizing PIF



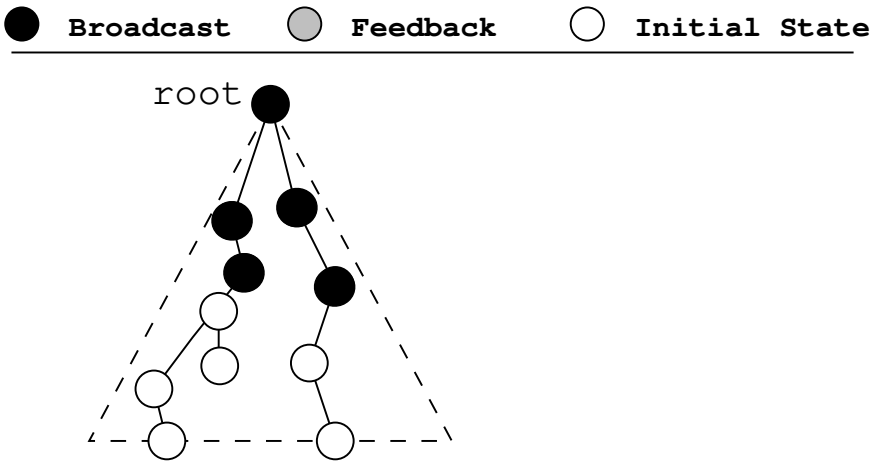
[*Bui, Datta, Petit, Villain. Distributed Computing* 2007]

Snap-Stabilizing PIF



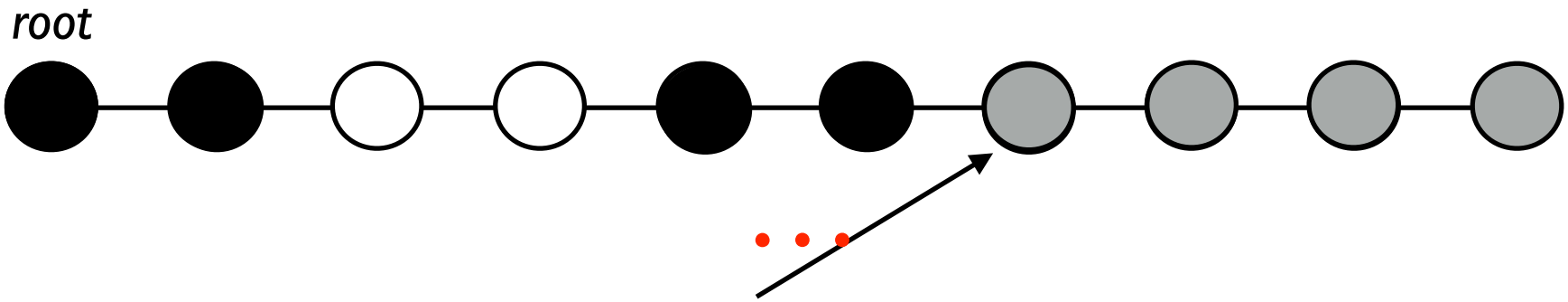
[*Bui, Datta, Petit, Villain. Distributed Computing* 2007]

Snap-Stabilizing PIF



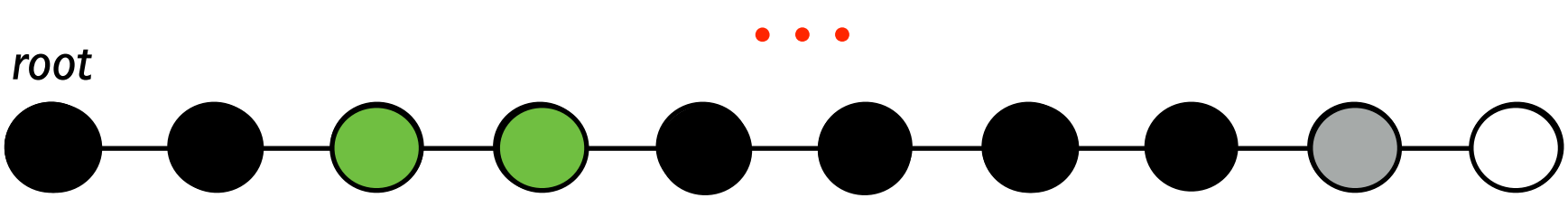
[*Bui, Datta, Petit, Villain. Distributed Computing* 2007]

Self-Stabilizing PIF in Trees with No Sense of Direction



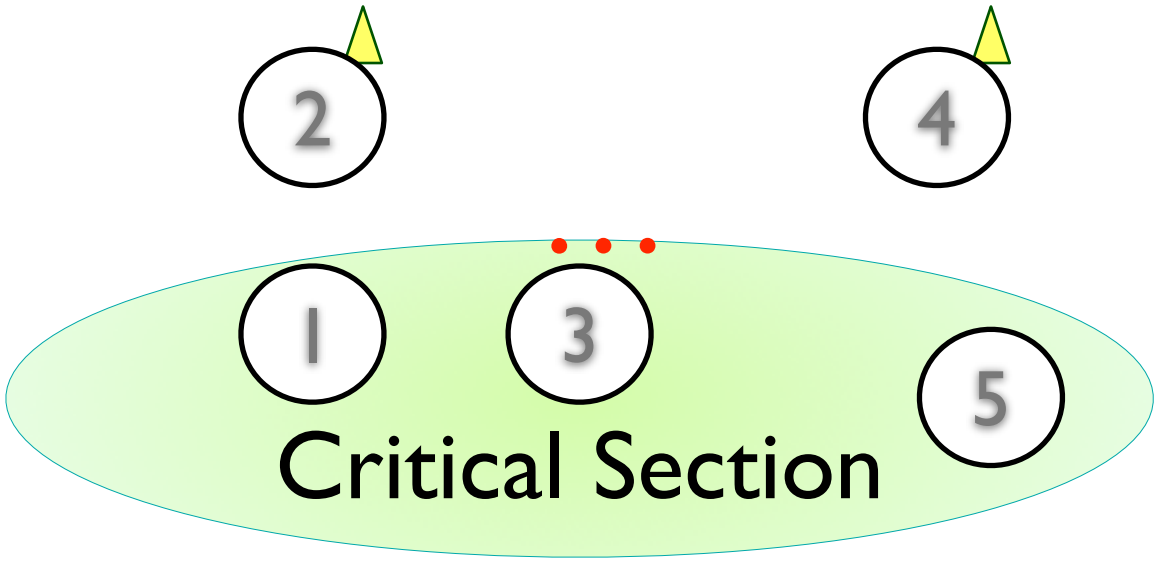
With no sense of direction, Broadcast can move toward the root

Snap-Stabilizing PIF
in
Trees with No Sense of Direction



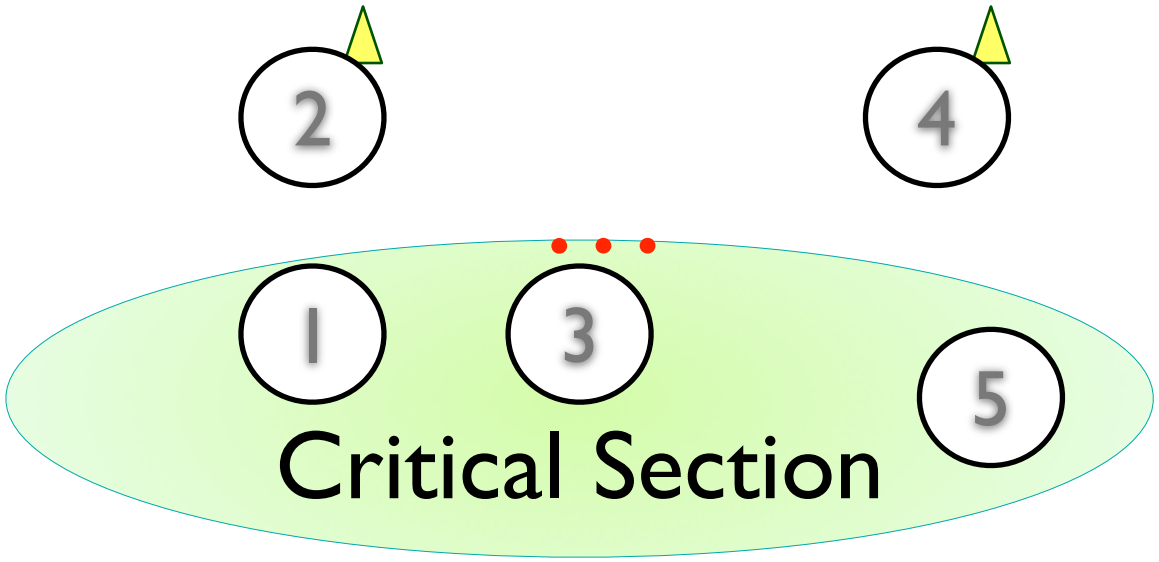
Self-Stabilizing Mutual Exclusion

- Safety: No two processes execute the critical section simultaneously.
- Liveness: Upon a request, a process enters the critical section in finite time.



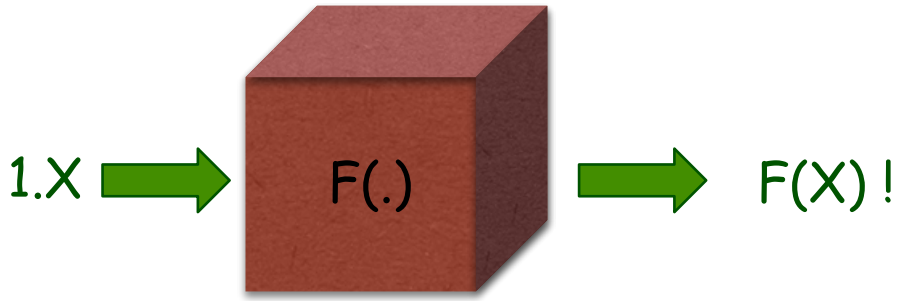
Snap-Stabilizing Mutual Exclusion

- Safety: No two processes execute the critical section simultaneously.
- Liveness: Upon a request, a process enters the critical section in finite time.



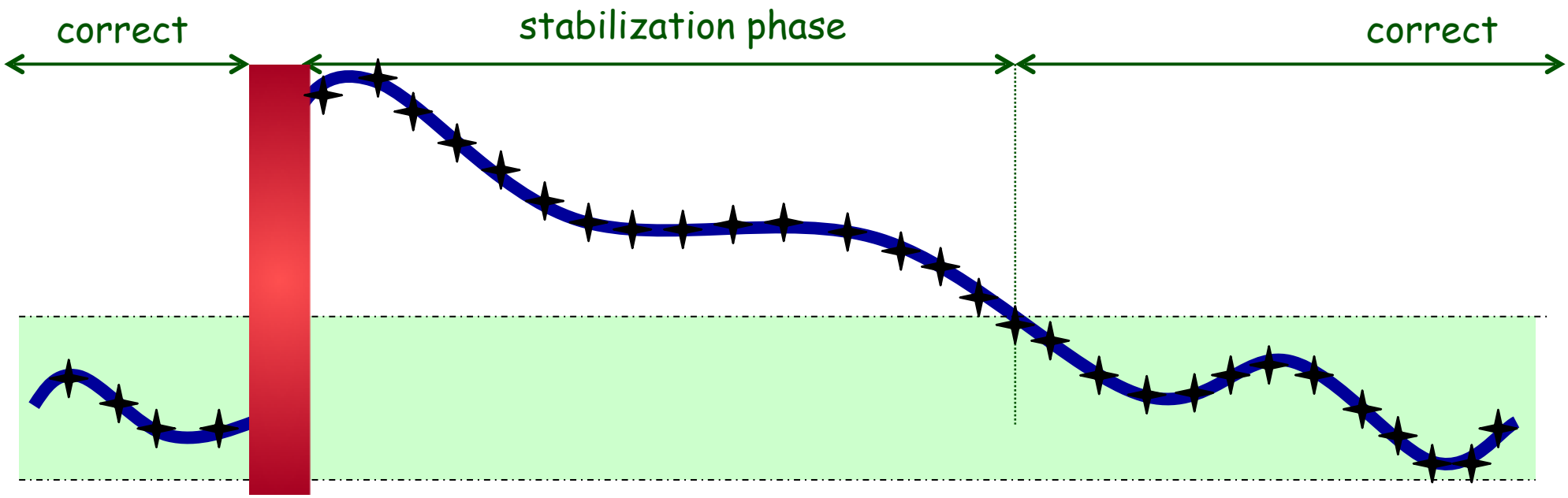
Snap-Stabilizing Distributed System

From the user's point of view…

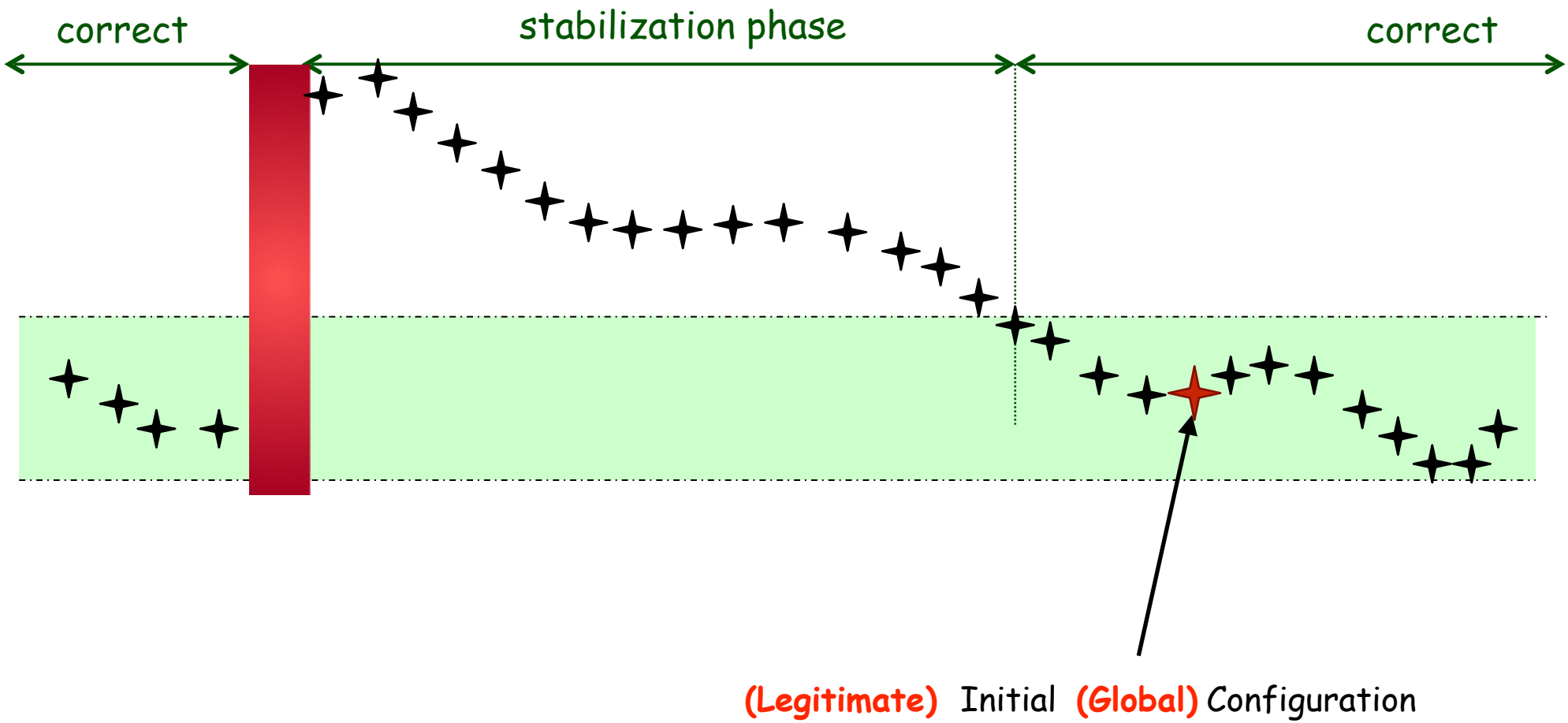


Stabilization Time equal zero

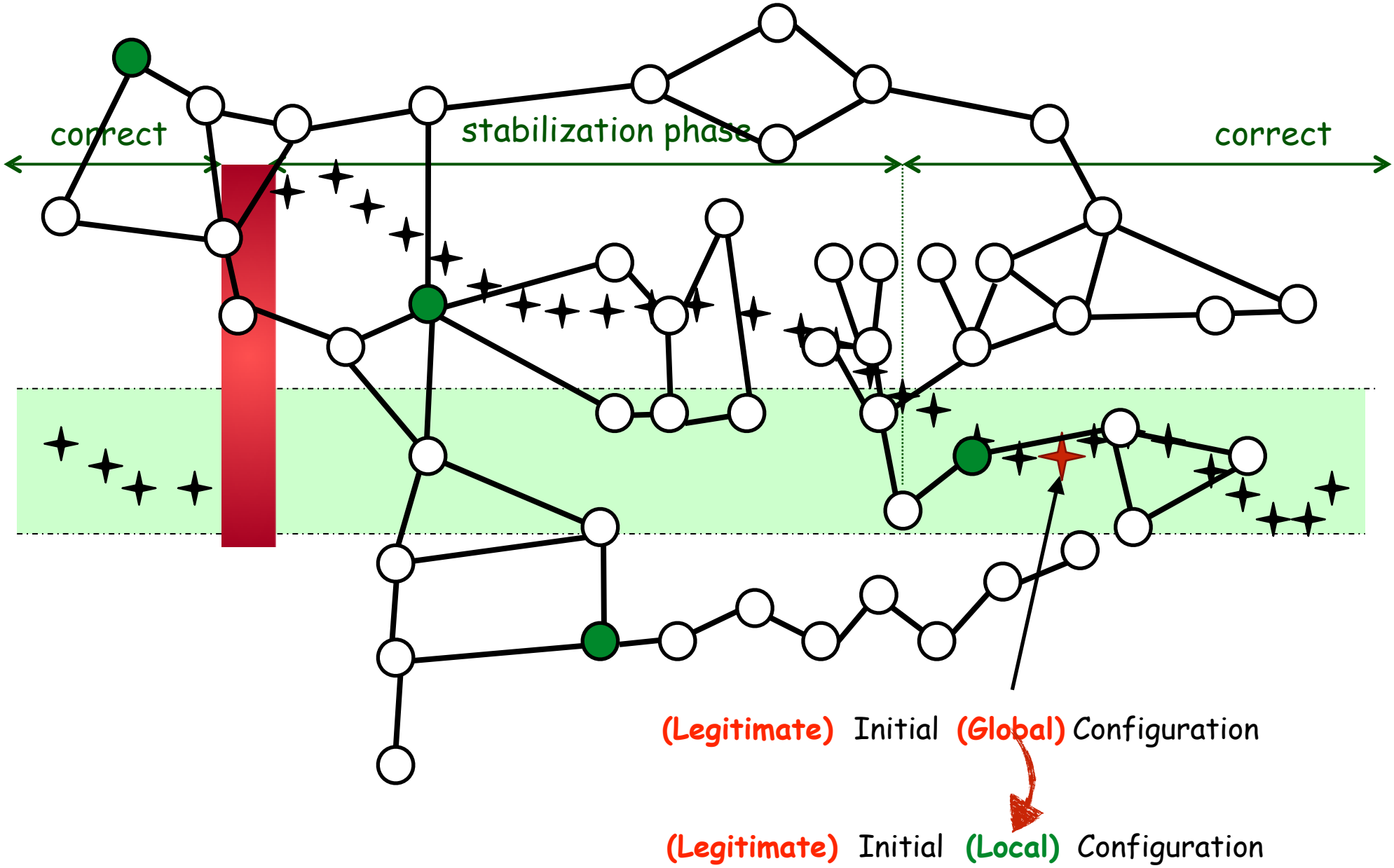
Distributed System



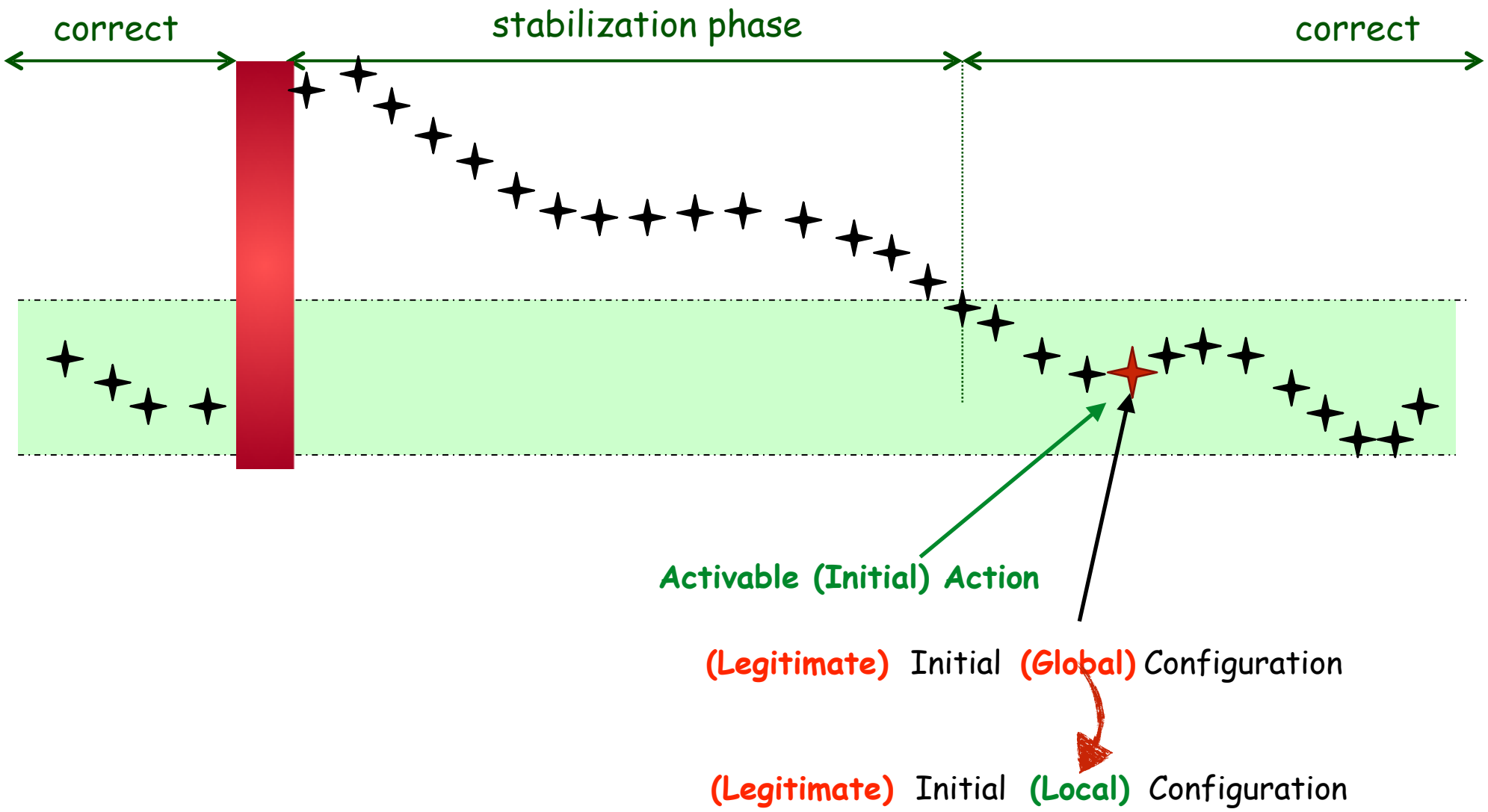
Distributed System



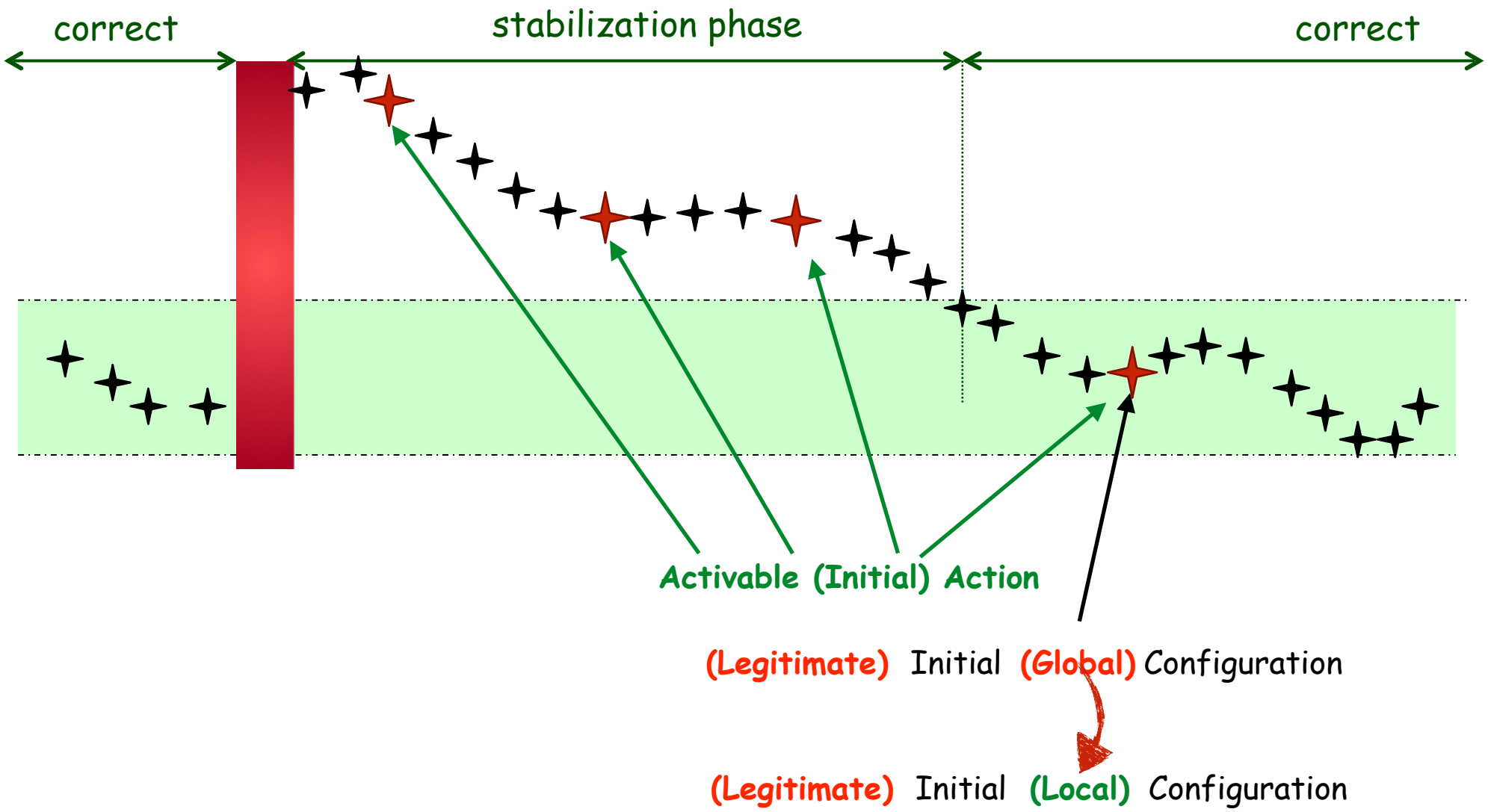
Distributed System



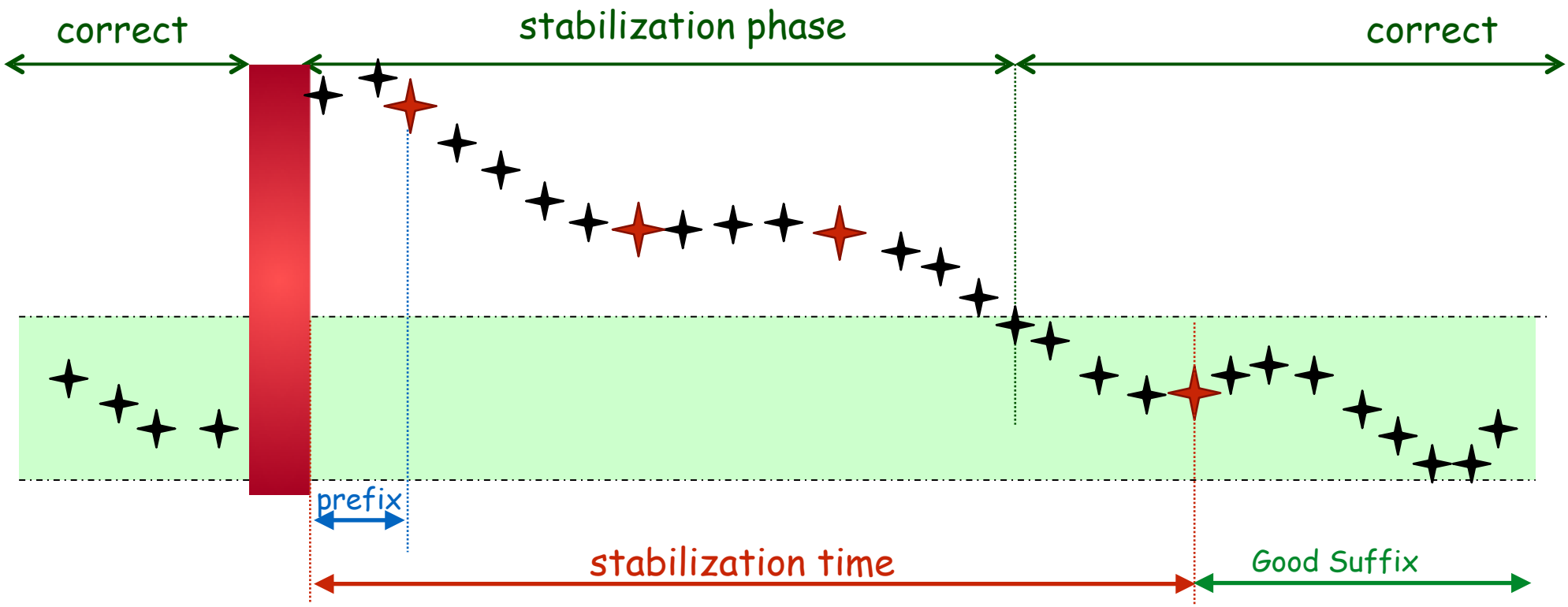
Distributed System



Distributed System

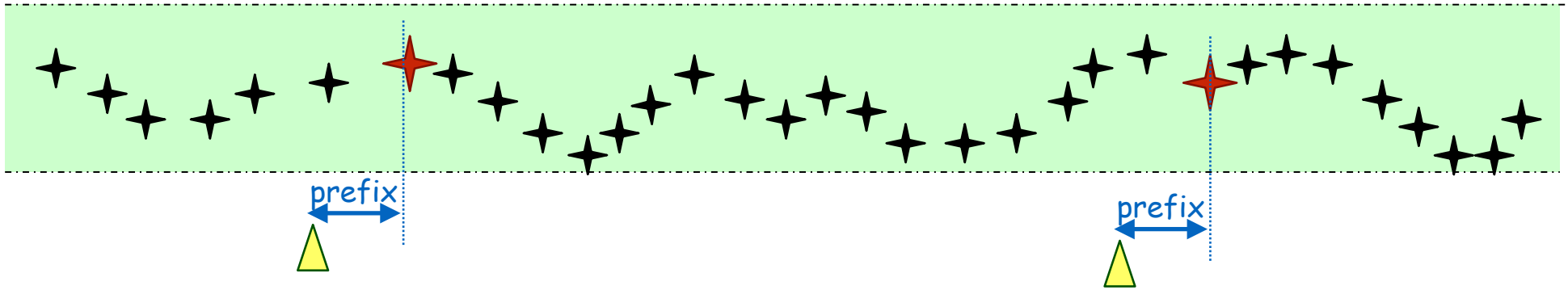


Distributed System



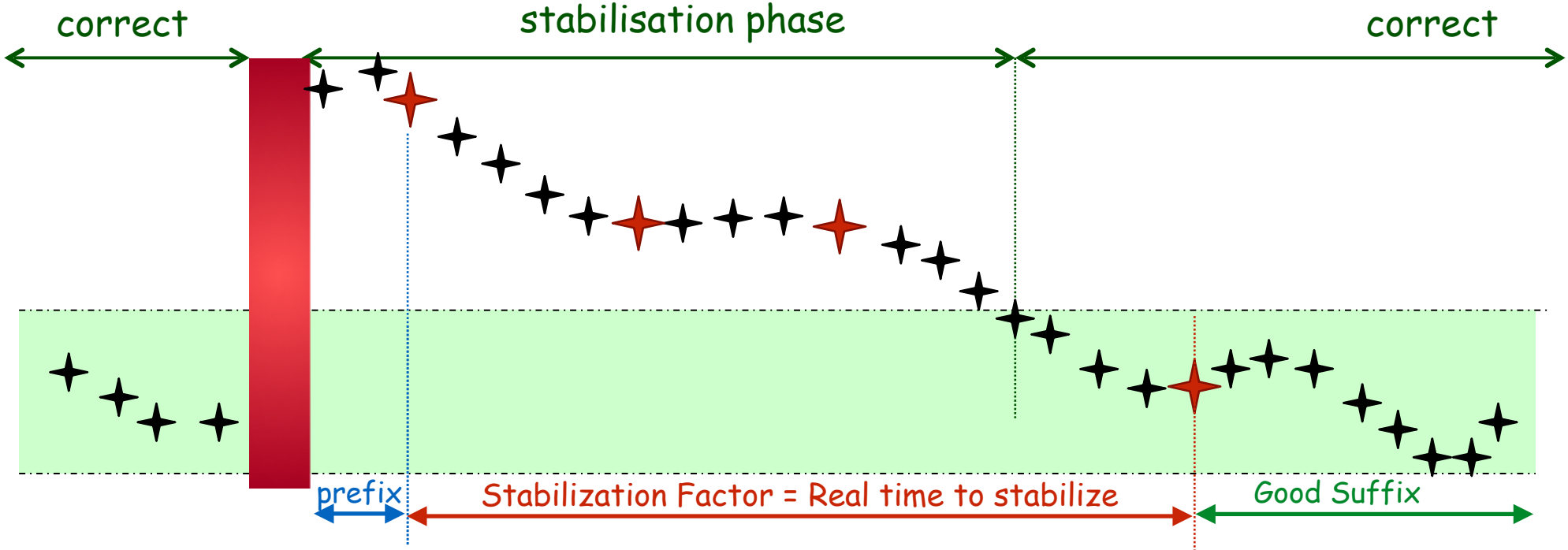
Distributed System

From the user's point of view…



Prefix : « response time » to start the algorithm (or the user request)

Self-Stabilizing Distributed System



Snap-Stabilizing Distributed System

Snap-stabilizing executions contain no stabilizing factors w.r.t. specifications

Stabilization Time is equal to 0.

Expressiveness of Snap-Stabilization

Can we provide a snap-stabilizing solution to every problem that has a self-stabilizing solution?

Expressiveness of Snap-Stabilization

- *What is the expressiveness of self-stabilization?*
  - A self-stabilizing transformer working in the message-passing model that transforms most of non-self-stabilizing algorithms (every problem that can be defined by a suffix-closed specification) into self-stabilizing ones.
    [S. Katz and K. Perry, DC, 1993]

- *What about the expressiveness snap-stabilization?*
  - A universal transformer that provides a snap-stabilizing version of any protocol that can be self-stabilized with the transformer of [KP93] (in the locally shared memory model).
    [A. Cournier, S. Devismes, A. K. Datta, S. Devismes, F. Petit, and V. Villain, TCS, 2016]

PIF

- Propagation of Information with Feedback in General Graphs

- **Distributed-Control Problems**
  - Broadcast, Routing, Synchronization, Protocol, Leader Election, Resource Sharing and Allocation, Graph Algorithms, Termination Detection, Deadlock Detection, Reset, Distributed Ranking, Distributed Sorting...

Self-Stabilizing Compiler

- GOAL
  - «Universal» Tool to Transform (Compile) any non self-stabilizing distributed algorithm into a self-stabilizing one [Katz and Perry 1993]

Snap-Stabilizing Compiler

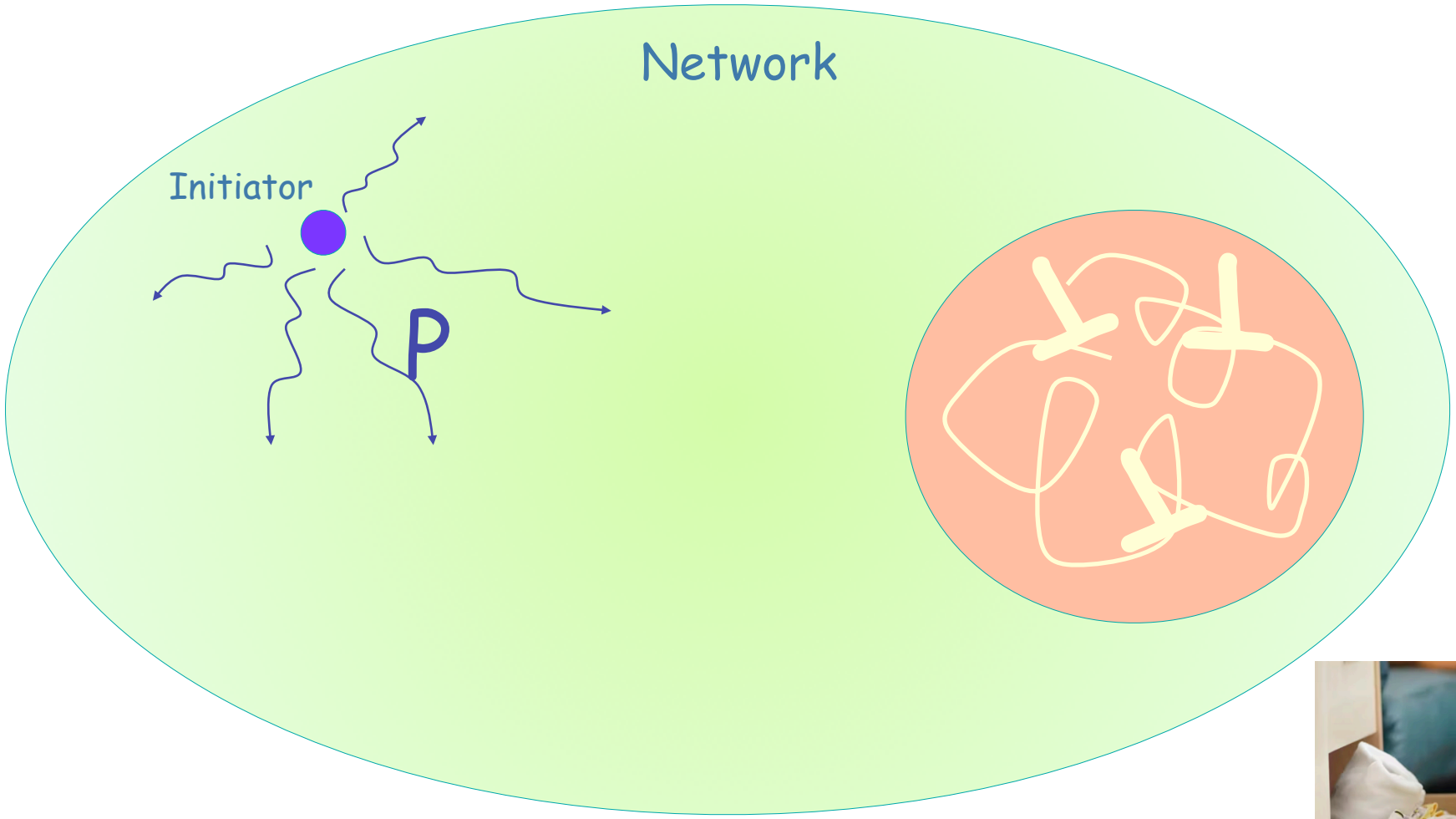
- GOAL
  - «Universal» Tool to Transform (Compile) any non self-stabilizing distributed algorithm into a snap-stabilizing one

Snap-Stabilizing Compiler

- IDEA
  - Snap-stabilizing leader test
  - Snap-stabilizing reset
  - Snap-stabilizing snapshot
  - Snap-stabilizing termination detection

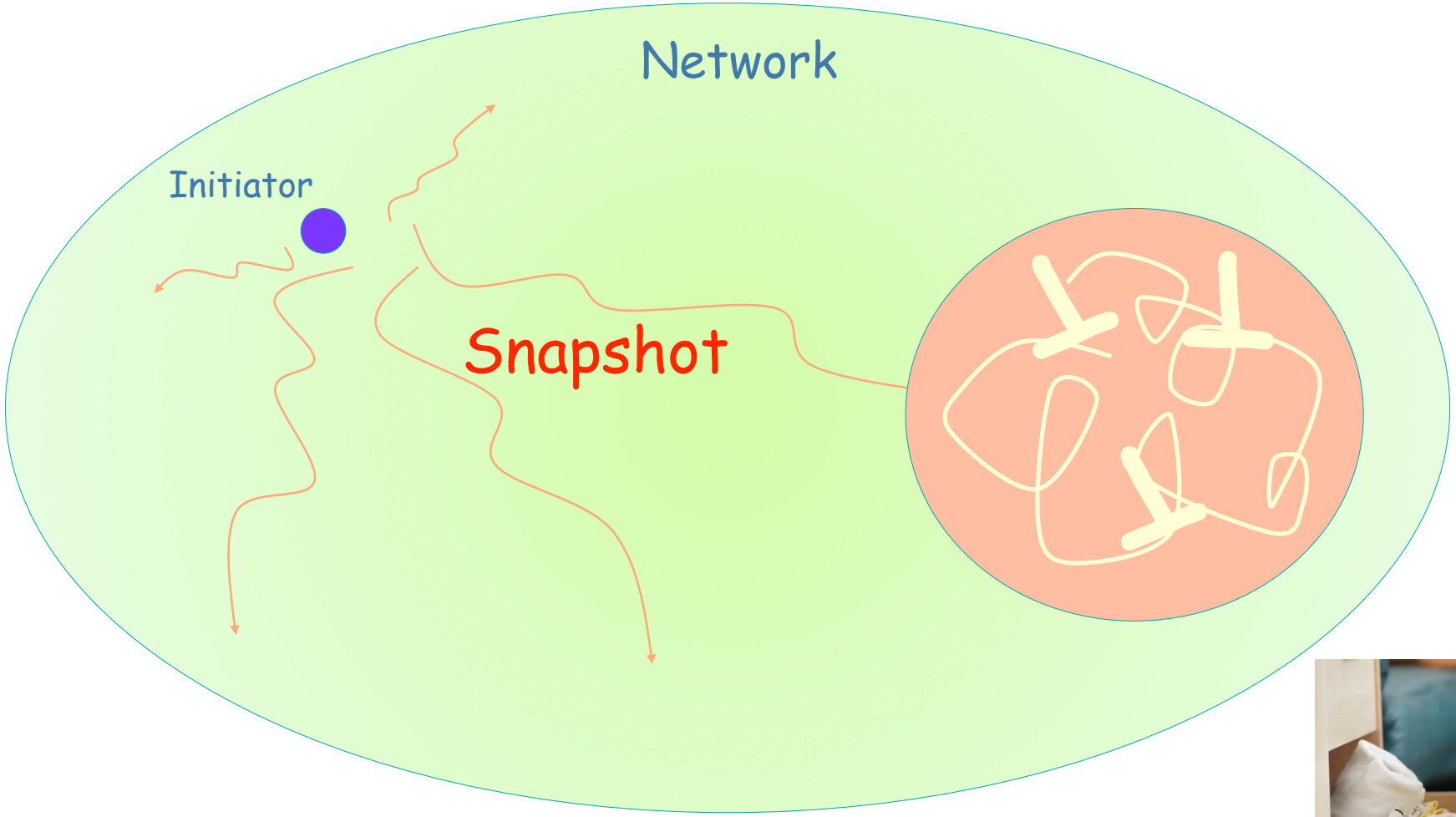
Snap-stabilizing PIF

Snap-Stabilizing Compiler



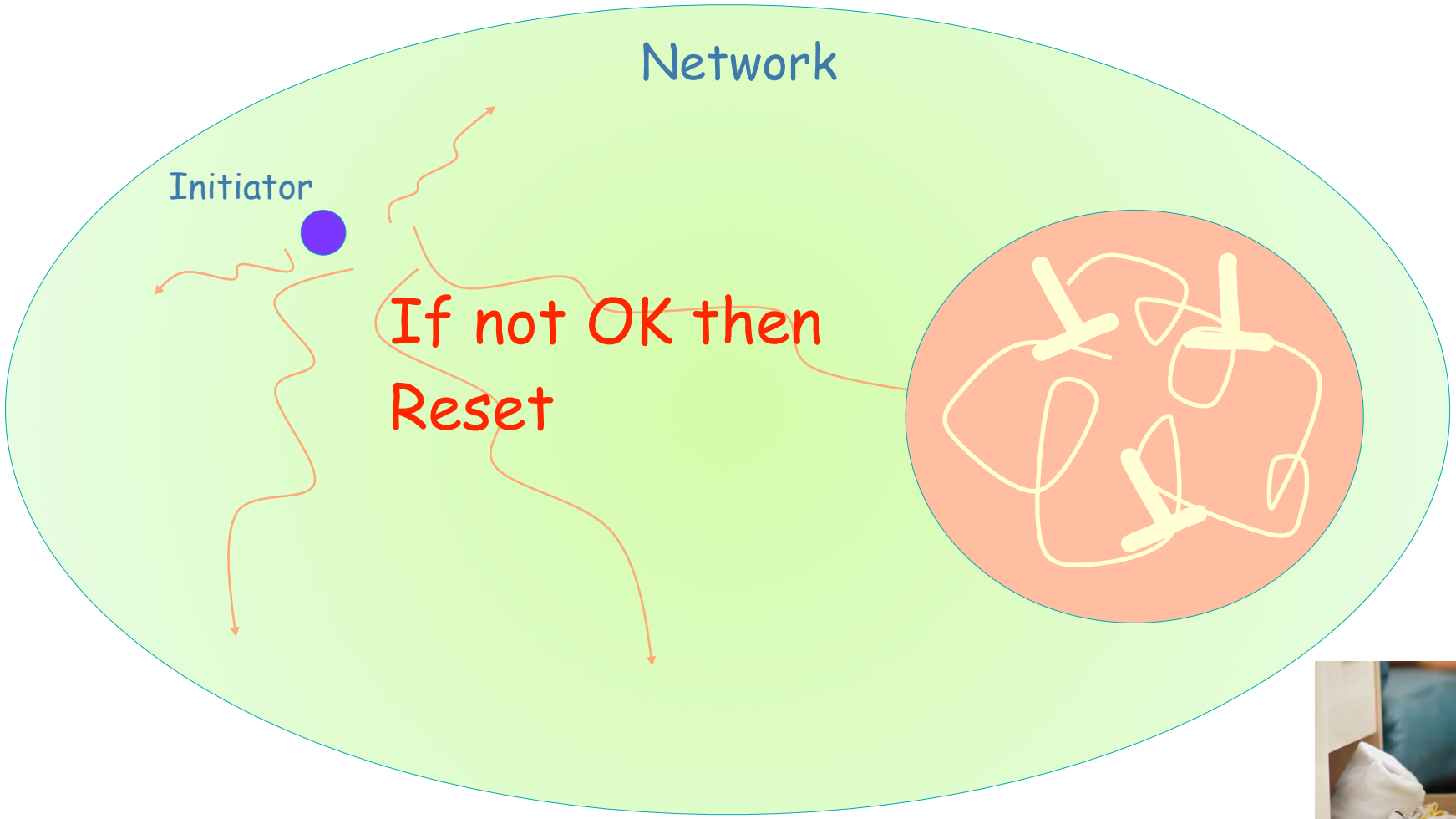
With a Single Initiator

Snap-Stabilizing Compiler



With a Single Initiator

Snap-Stabilizing Compiler



With a Single Initiator

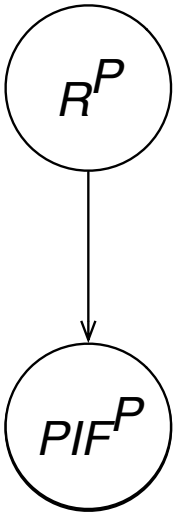
Snap-Stabilizing Compiler

1. *Each step of P (which is not even self-stabilizing) is scheduled using a snap-stabilizing PIF wave.*

With a Single Initiator

Snap-Stabilizing Compiler

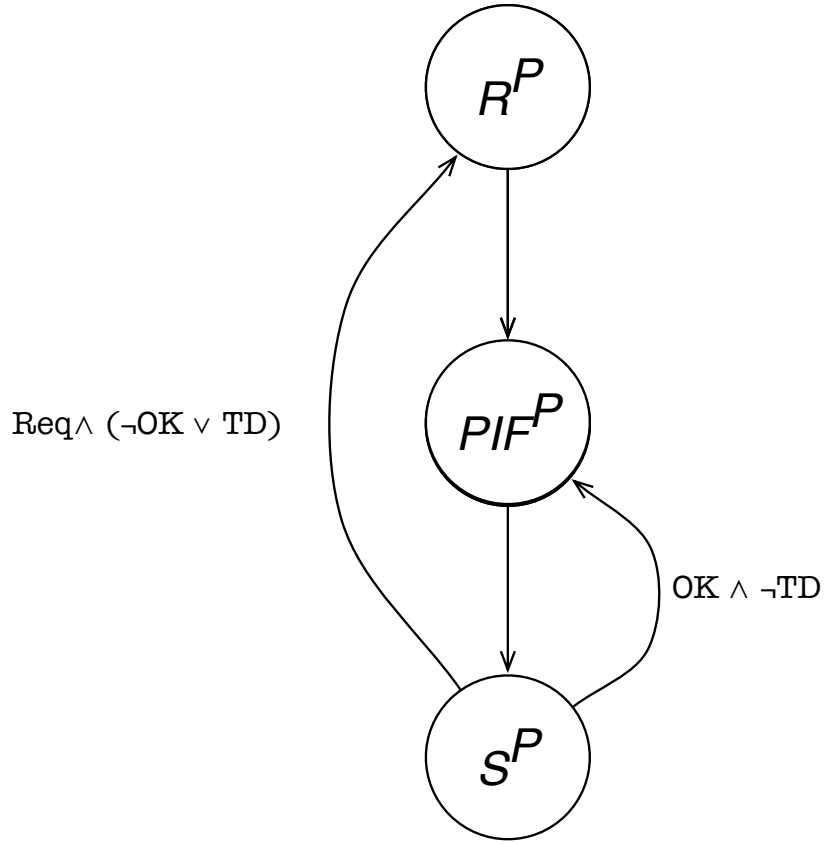
1. *Each step of P (which is not even self-stabilizing) is scheduled using a snap-stabilizing PIF wave.*
2. *A simple attempt to transform T into a snap-stabilizing protocol is to reset the network using R^P before starting P.*



With a Single Initiator

Snap-Stabilizing Compiler

1. *Each step of P (which is not even self-stabilizing) is scheduled using a snap-stabilizing PIF wave.*
2. *A simple attempt to transform T into a snap-stabilizing protocol is to reset the network using R^P before starting P.*
3. *A snap-stabilizing snapshot S^P protocol is associated to each step of P to detect if either (1) the configuration is normal w.r.t. to the specifications of P, (2) P terminated, or (3) none of this two cases.*



With a Single Initiator

Snap-Stabilizing Compiler

- Same principle
  - Snap-stabilizing leader test
  - Snap-stabilizing leader election
  - Snap-stabilizing reset
  - Snap-stabilizing snapshot
  - Snap-stabilizing termination detection

With a Multiple Initiator Protocol

Expressiveness of Snap-Stabilization

- A universal transformer that provides a snap-stabilizing version of any protocol that can be self-stabilized with the transformer of [KP93] (in the locally shared memory model).

  [Cournier, Datta, Petit, Villain. Enabling snap-stabilization, 2003.]

  [Cournier, Devismes, Villain. From Self- to Snap- Stabilization. 2006]

  [A. Cournier, S. Devismes, A. K. Datta, S. Devismes, F. Petit, and V. Villain, TCS, 2016]

- Snap-Stabilization in Message-Passing System : Snap-stabilization requires bounded-capacity channels.

  [S. Delaët, S. Devismes, M. Nesterenko, S. Tixeuil, JPDC 2010]

Probabilistic Snap-stabilization in Anonymous Networks

[K. Altisen, S. Devismes, TCS 2017]

- Key Idea:
  - Relaxing snap-stabilization without altering its strong safety guarantees
  - to address anonymous networks
- Weakened form of snap-stabilization

- 2 probabilistic snap-stabilizing protocols for the *guaranteed service leader election* for *anonymous* networks in the atomic-state model (assuming the knowledge of B, $B<n\leq 2B$)

Snap-Stabilization in Anonymous Distributed Systems

- Snap-Stabilizing Waves in Anonymous Networks.

  [C. Boulinier, M. Levert, F. Petit, ICDCN 2008]

  - Key Idea : PIF in anonymous distributed systems based on the unison in [C. Boulinier, F. Petit, V. Villain, PODC 2004]
  - Generic snap-stabilizing tool for anonymous networks (assuming the knowledge of D, the diameter of the network)
  - Snap-stabilizing causal atomic broadcast for anonymous distributed systems, that can be used as a pipeline of messages.

Snap-Stabilization in Anonymous Distributed Systems

- Snap-Stabilizing Waves in Anonymous Networks.

  [C. Boulinier, M. Levert, F. Petit, ICDCN 2008]

- Silent Anonymous Snap-Stabilizing Termination Detection.

  [L. Blin, C. Johnen, G. Le Bouder, F. Petit, under submission, 2022]

  - Key Idea: Detection whether an observed terminating silent self-stabilizing algorithm A has converged to a configuration that satisfies an intended predicate.
  - Based on any self-stabilizing unison in the literature.

Snap-Stabilization in Dynamic Distributed Systems?