# Probabilistic Self-Stabilization

Emanuele Natale†

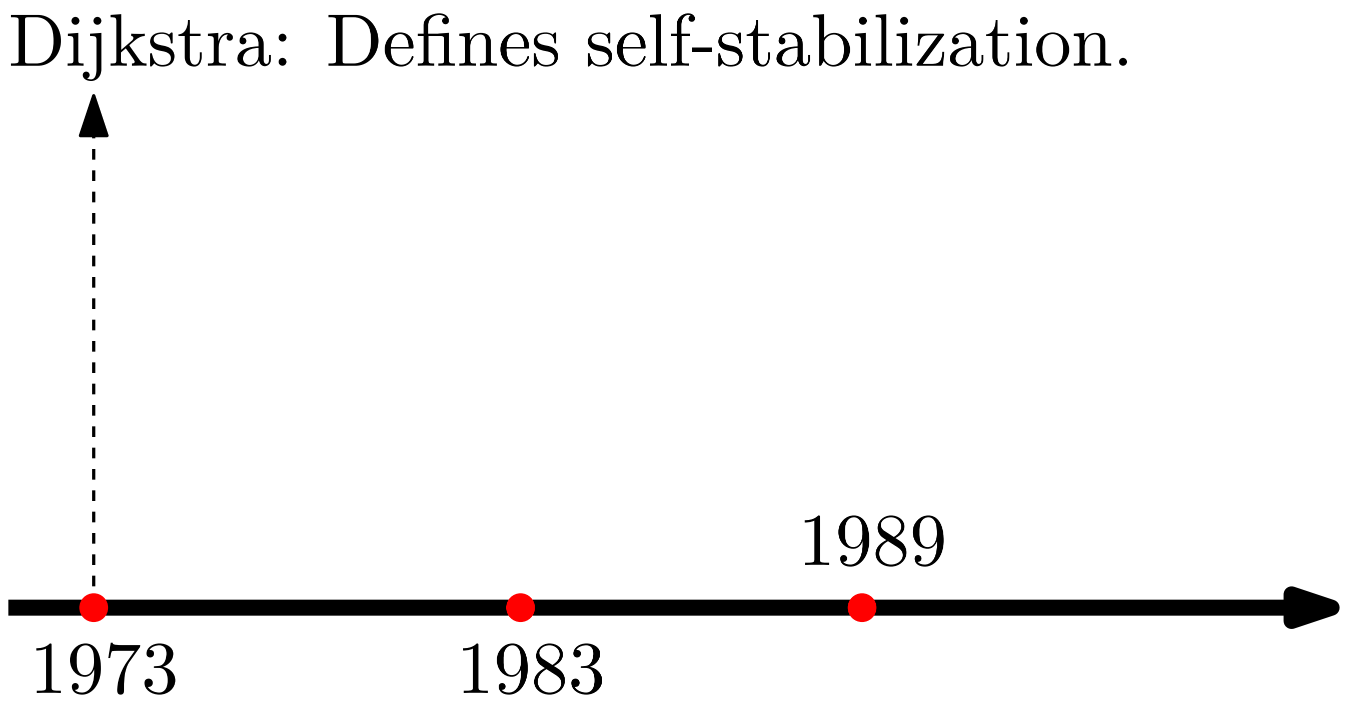
joint work with
Luca Becchetti†, Andrea Clementi*,
and Francesco Pasquale*

* UNIVERSITA' DEGLI STUDI DI ROMA TorVergata

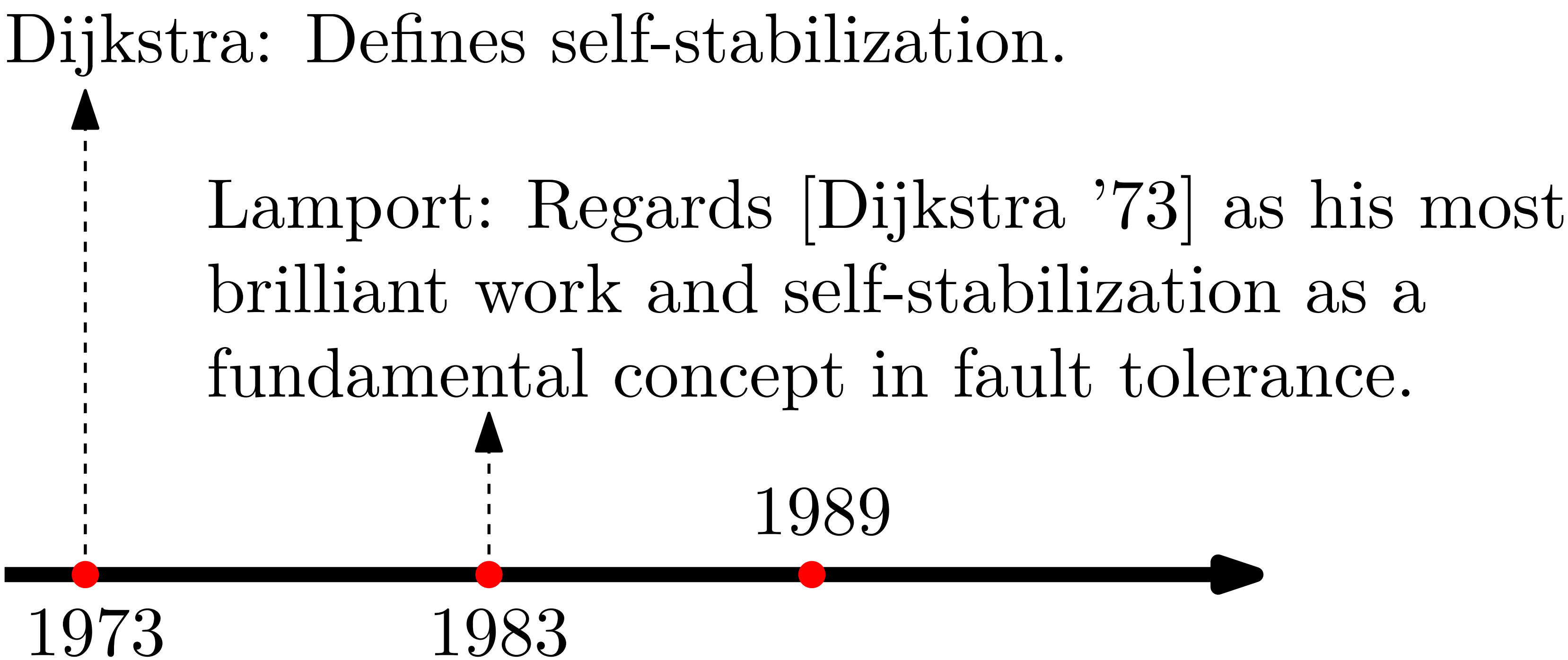
† SAPIENZA UNIVERSITÀ DI ROMA

16th Italian Conference on Theoretical Computer Science
Firenze, 9 – 11 September 2015

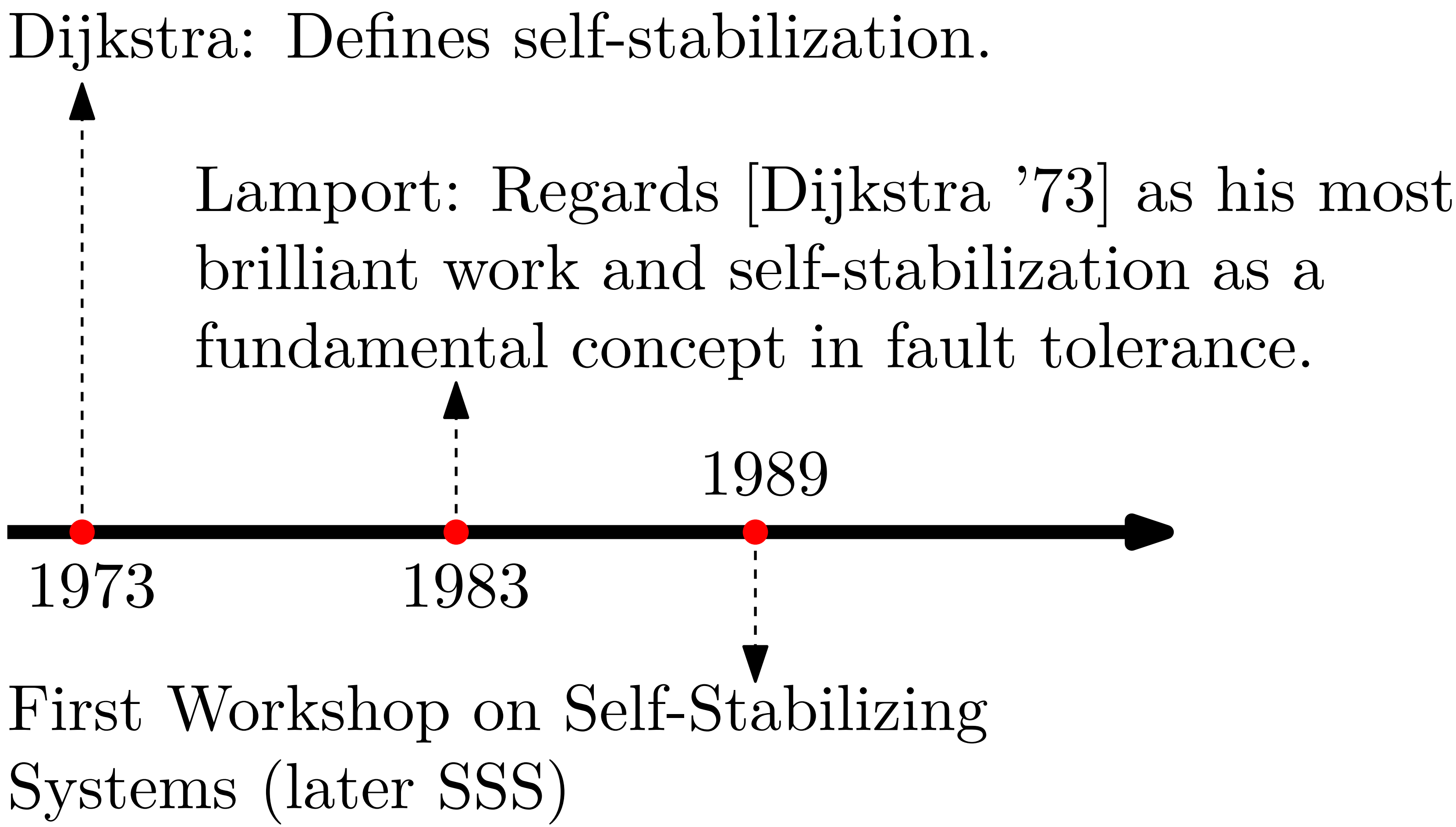
# (Brief) Timeline of Self-Stabilization



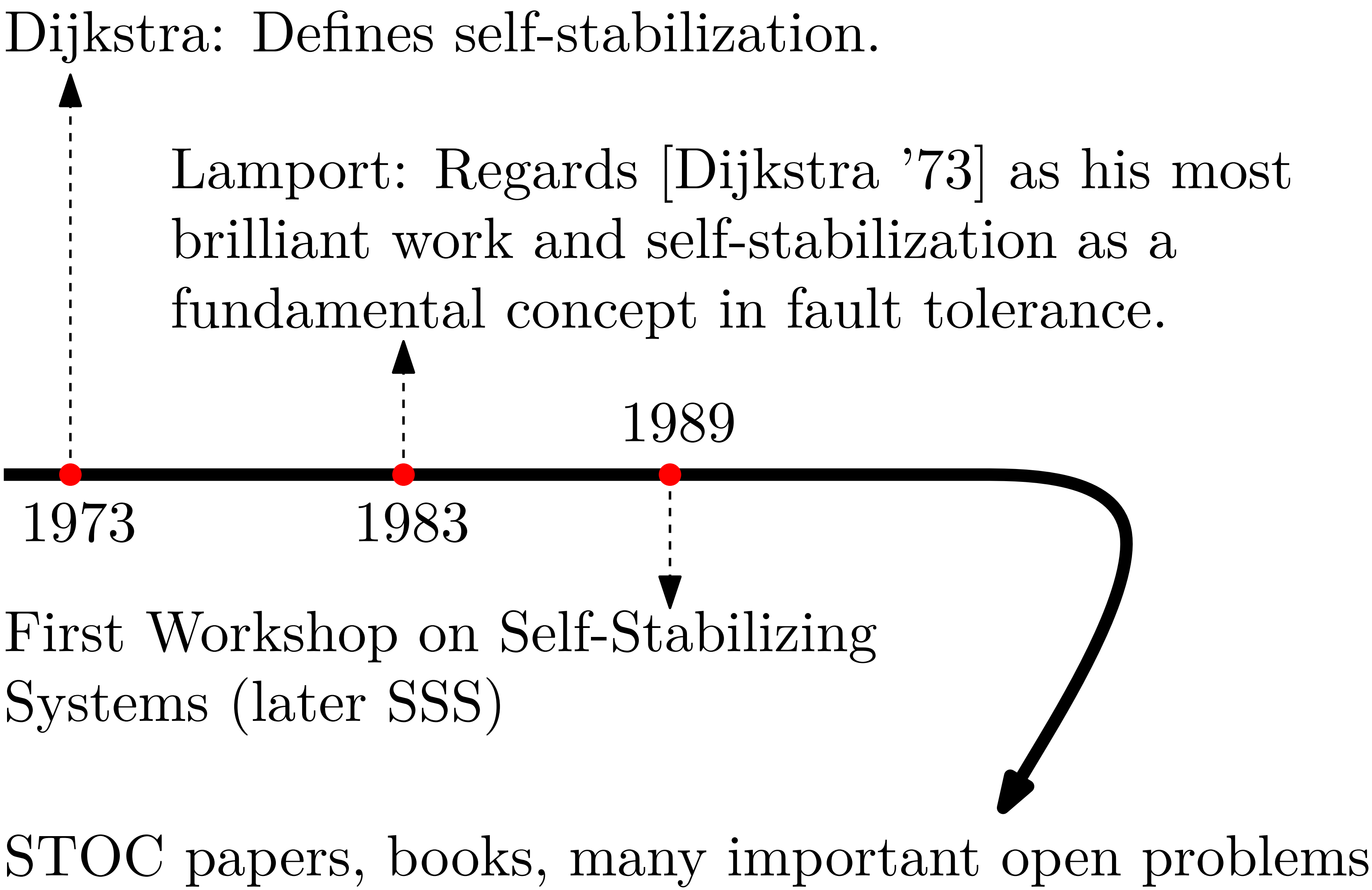
# (Brief) Timeline of Self-Stabilization



# (Brief) Timeline of Self-Stabilization



# (Brief) Timeline of Self-Stabilization



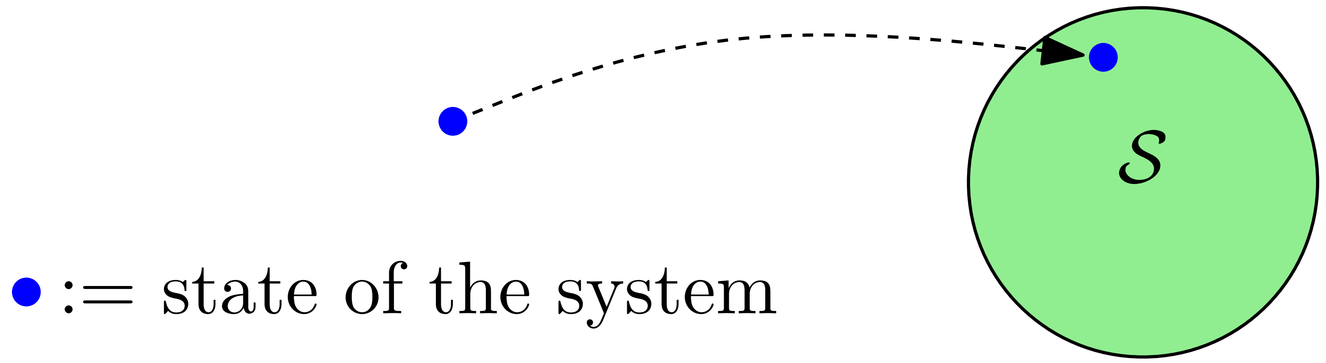
# Self Stabilization

$\mathcal{S} :=$ {"correct states of the system" }.

# Self Stabilization

$\mathcal{S}$ := {"correct states of the system" }.

**Convergence.** Starting from any state, the system eventually reaches a state in $\mathcal{S}$.

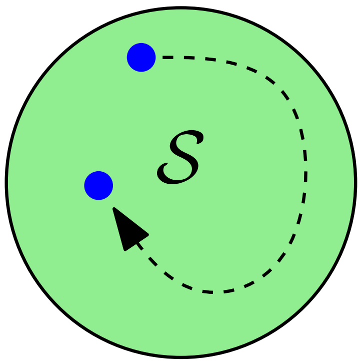


# Self Stabilization

$\mathcal{S}$ := {"correct states of the system" }.

**Convergence.** Starting from any state, the system eventually reaches a state in $\mathcal{S}$.

**Closure**. If the system is in a state in $\mathcal{S}$, it stays in $\mathcal{S}$, provided no fault happens.




● := state of the system

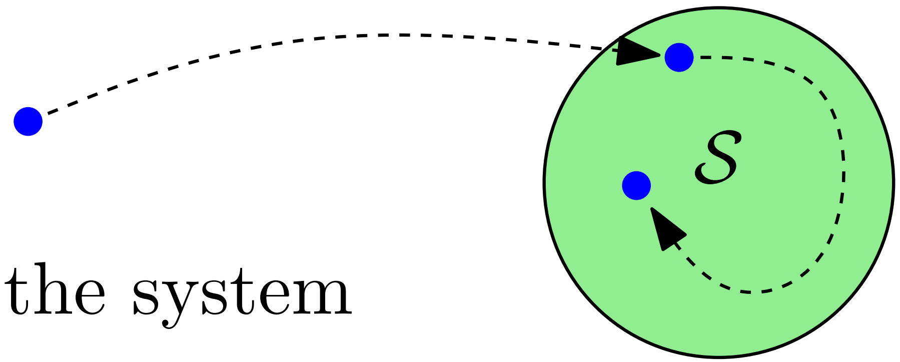
# Self Stabilization

$\mathcal{S}$ := {"correct states of the system" }.

**Convergence.** Starting from any state, the system eventually reaches a state in $\mathcal{S}$.

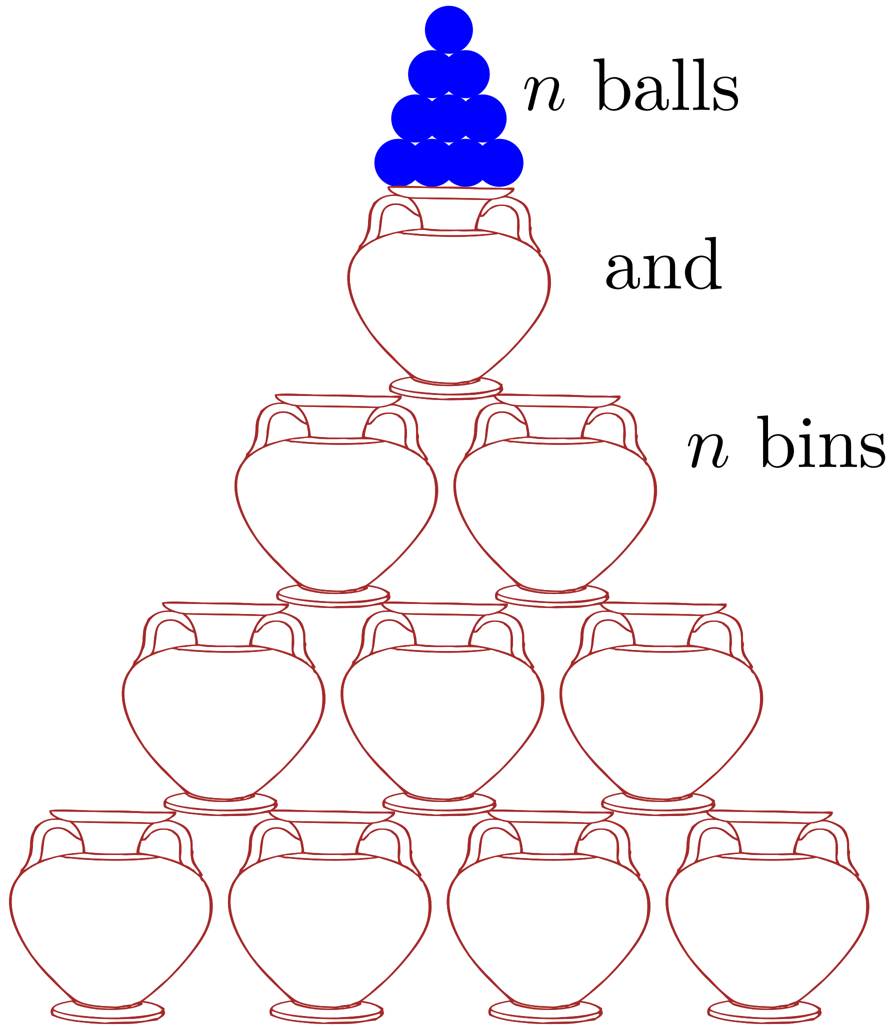
**Closure.** If the system is in a state in $\mathcal{S}$, it stays in $\mathcal{S}$, provided no fault happens.

A system is self-stabilizing iff guarantees convergence and closure w.r.t. $\mathcal{S}$.



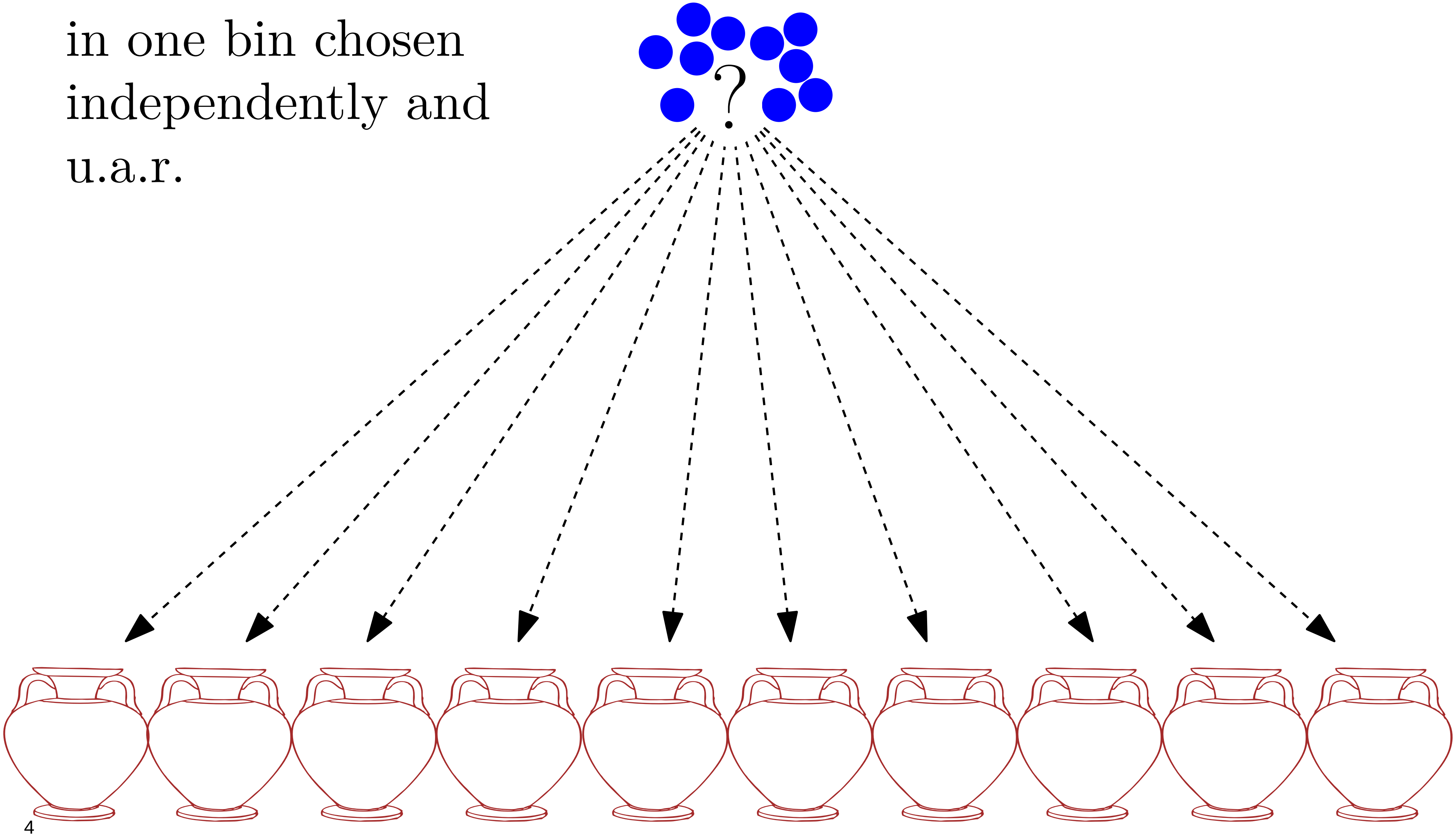

● := state of the system

# Balls-into-Bins



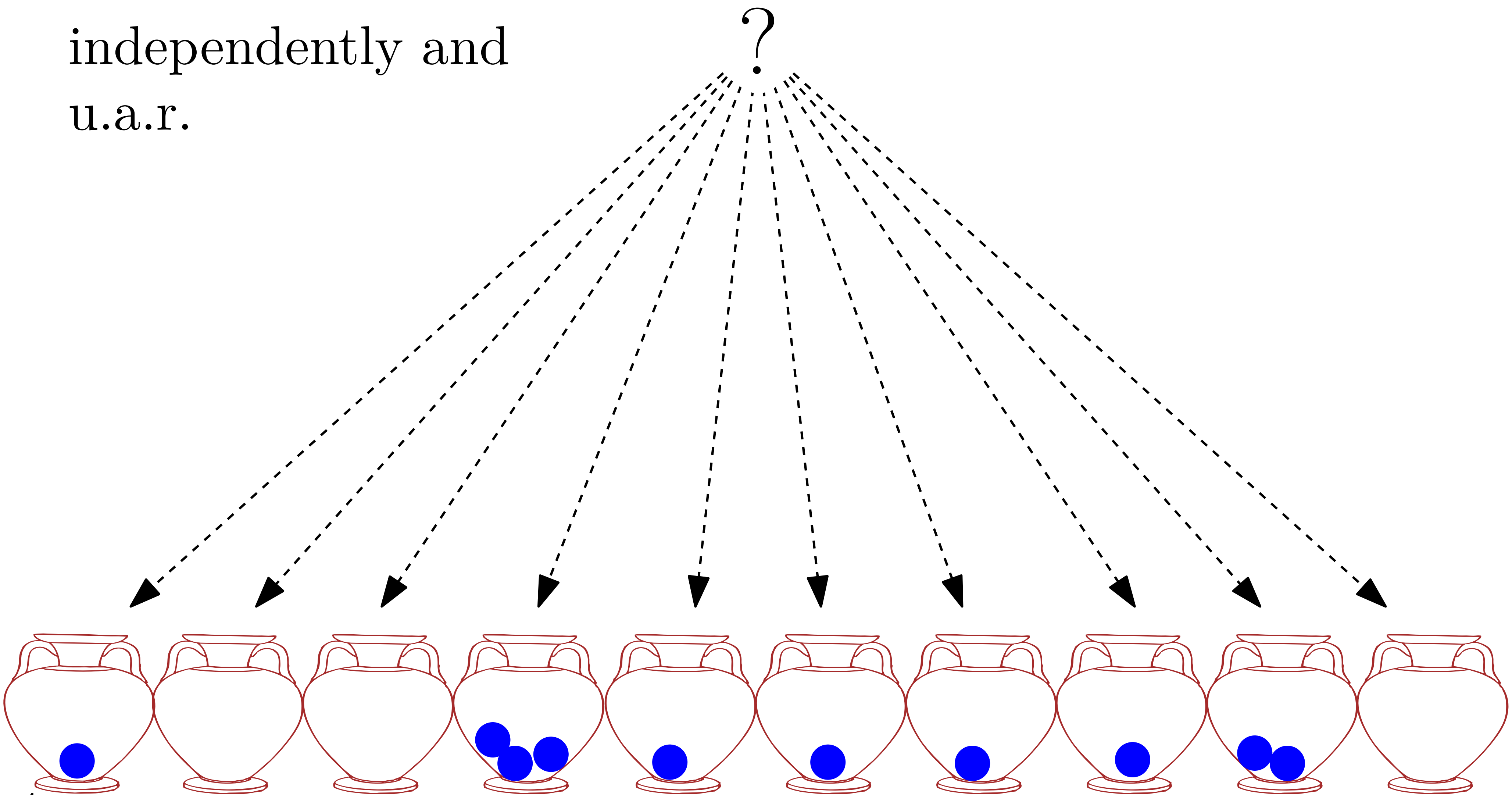
# Balls-into-Bins

Each ball is thrown in one bin chosen independently and u.a.r.



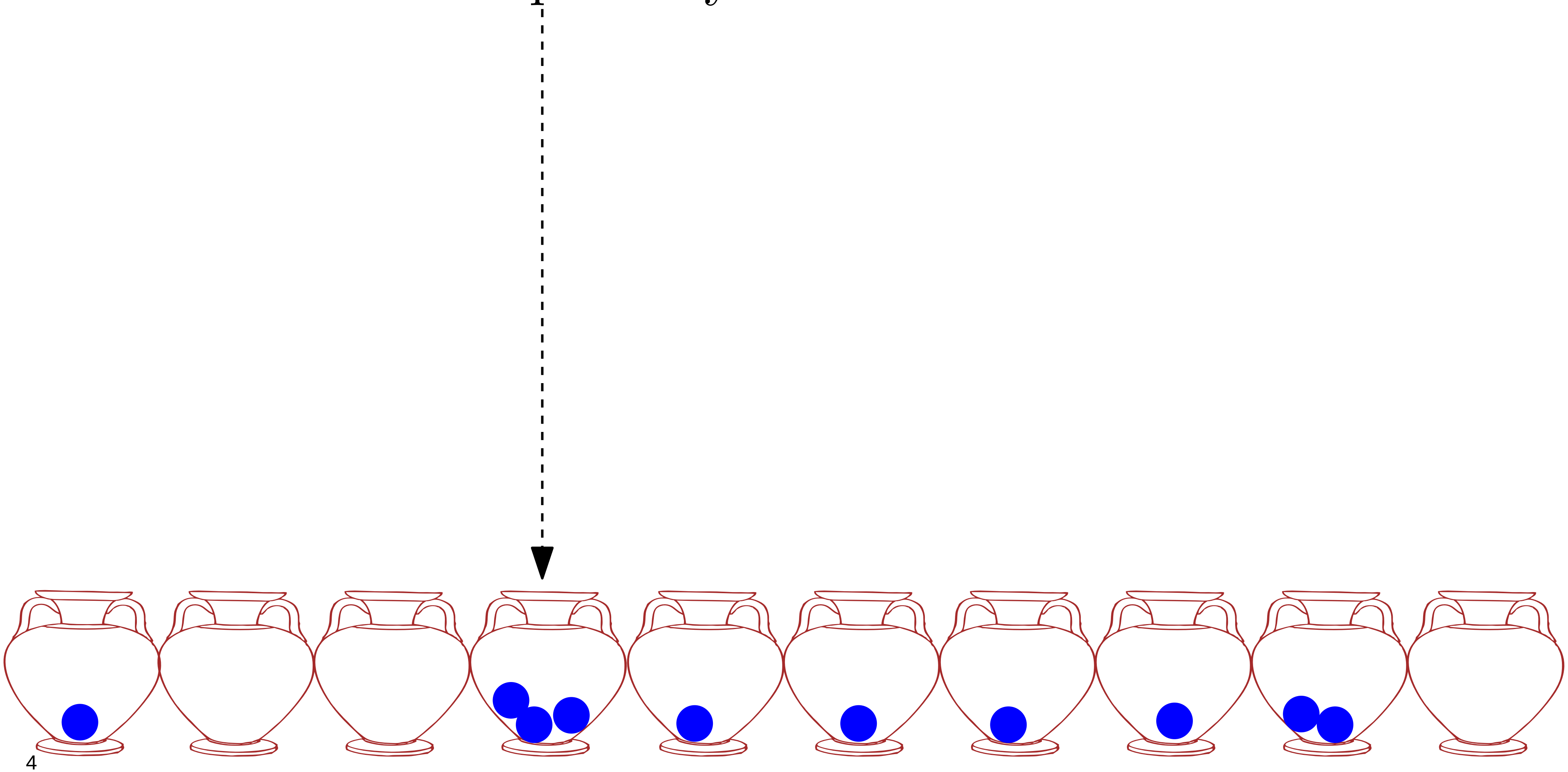
# Balls-into-Bins

Each ball is thrown in one bin chosen independently and u.a.r.



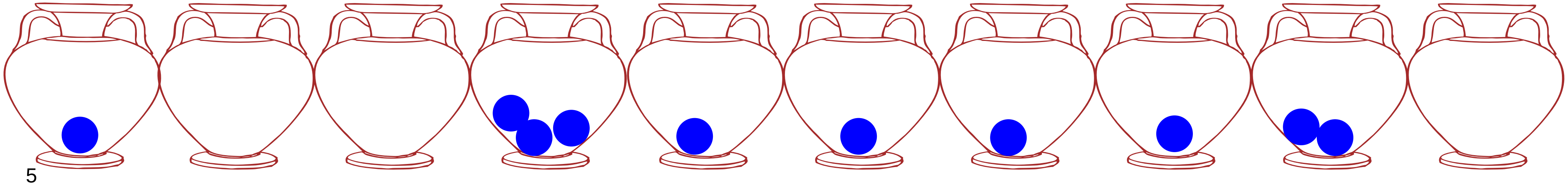
# Balls-into-Bins

*Maximum load*: maximum number of balls that end up in any bin.



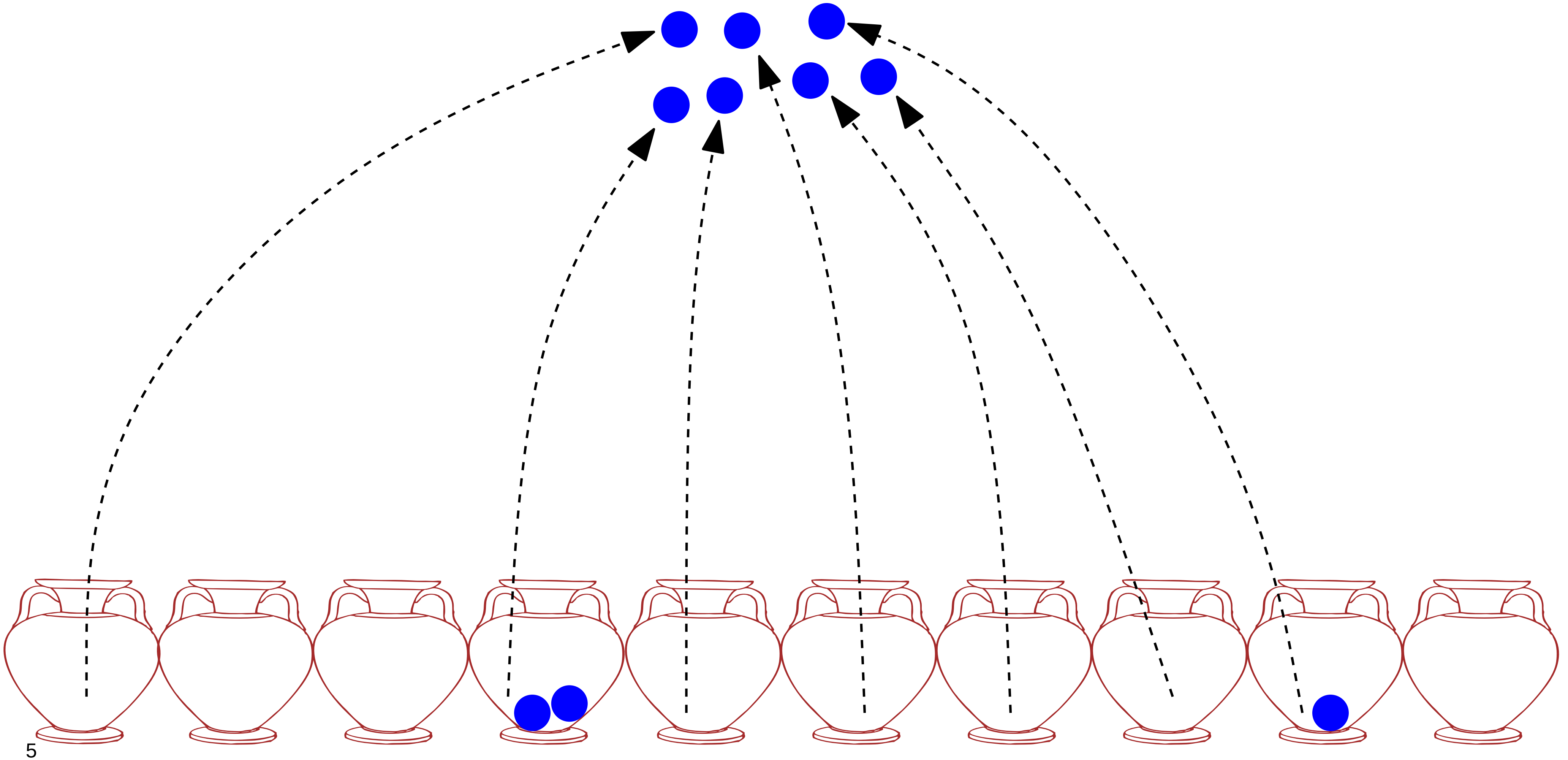
# Repeated Balls-into-Bins

At each round, pick one ball from each non-empty bin...



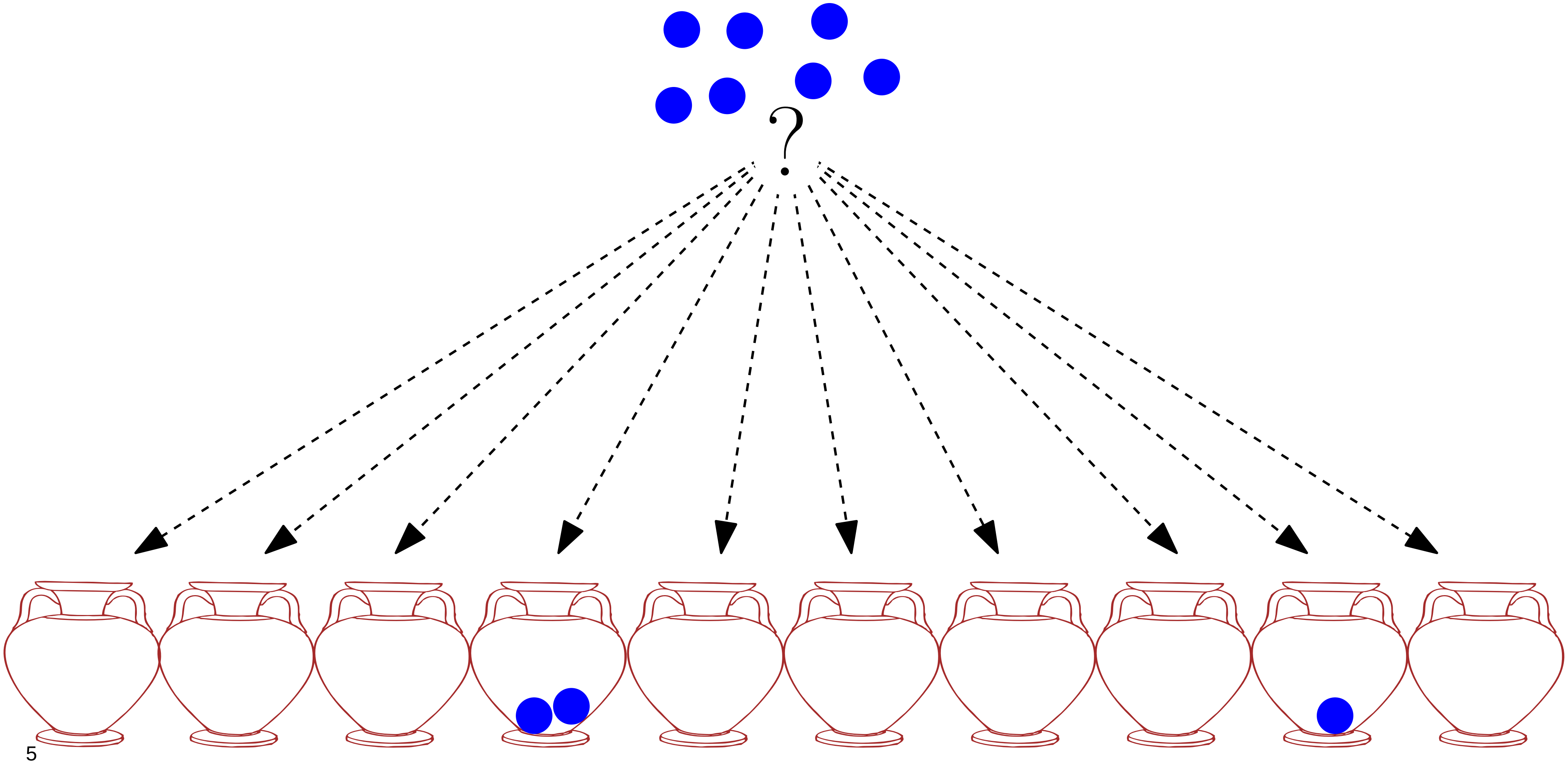
# Repeated Balls-into-Bins

At each round, pick one ball from each non-empty bin...



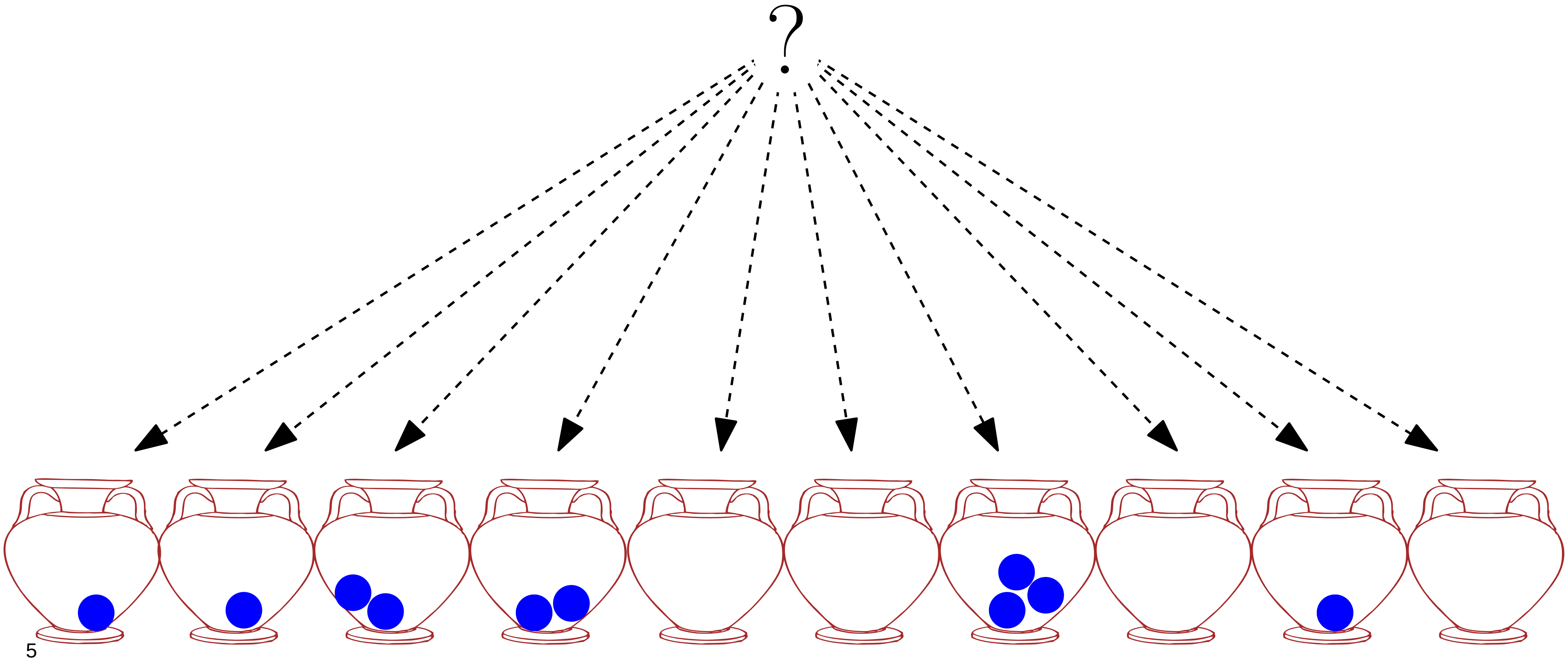
# Repeated Balls-into-Bins

At each round, pick one ball from each non-empty bin...
...and throw them again u.a.r.



# Repeated Balls-into-Bins

At each round, pick one ball from each non-empty bin...
...and throw them again u.a.r.



# Repeated Balls-into-Bins

At each round, pick one ball from each non-empty bin...
...and throw them again u.a.r.

*Max load*: max. number of balls in any bin.

# Self-stabilizing?

Let $\mathcal{S} = \{\text{config. with max load } \mathcal{O}(\log n)\}$
Is the repeated balls into bins self-stabilizing?

# Self-stabilizing?

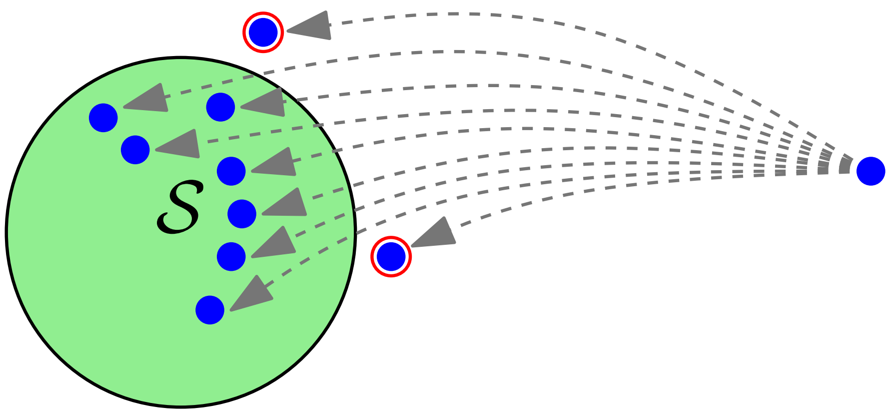
Let $\mathcal{S} = \{\text{config. with max load } \mathcal{O}(\log n)\}$
Is the repeated balls into bins self-stabilizing?

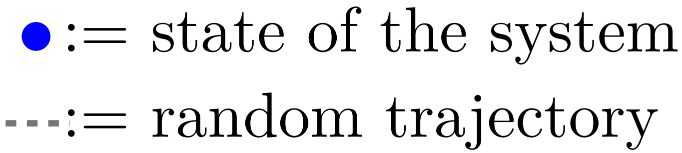
Convergence?

# Self-stabilizing?

Let $\mathcal{S} = \{\text{config. with max load } \mathcal{O}(\log n)\}$
Is the repeated balls into bins self-stabilizing?

Convergence? NO.



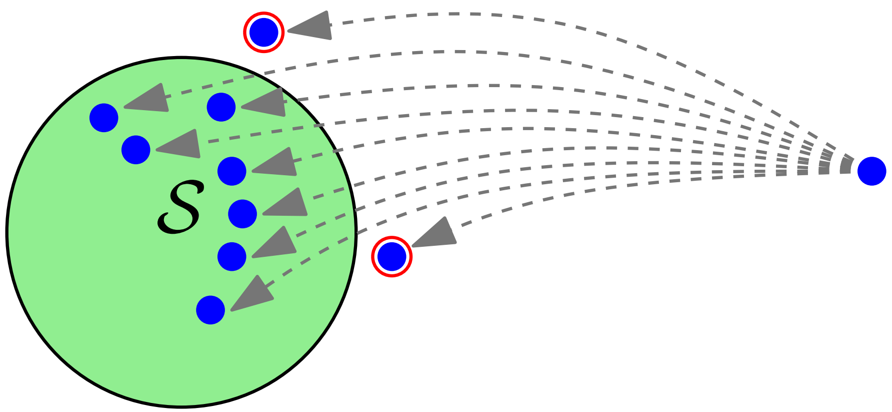



# Self-stabilizing?

Let $\mathcal{S} = \{\text{config. with max load } \mathcal{O}(\log n)\}$
Is the repeated balls into bins self-stabilizing?

Convergence? NO.

Closure?



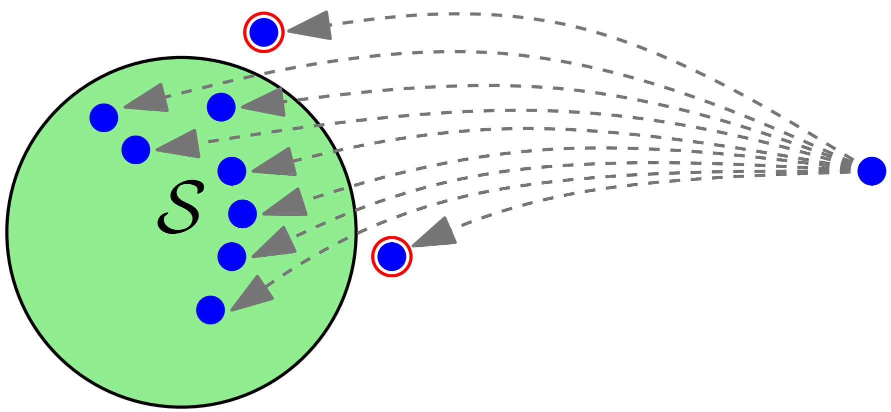

● := state of the system

--- := random trajectory

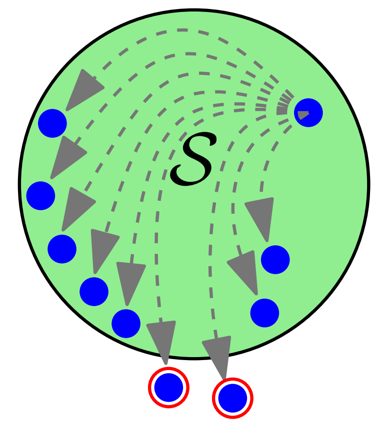
# Self-stabilizing?

Let $\mathcal{S} = \{\text{config. with max load } \mathcal{O}(\log n)\}$
Is the repeated balls into bins self-stabilizing?

Convergence? NO.




Closure? NO.

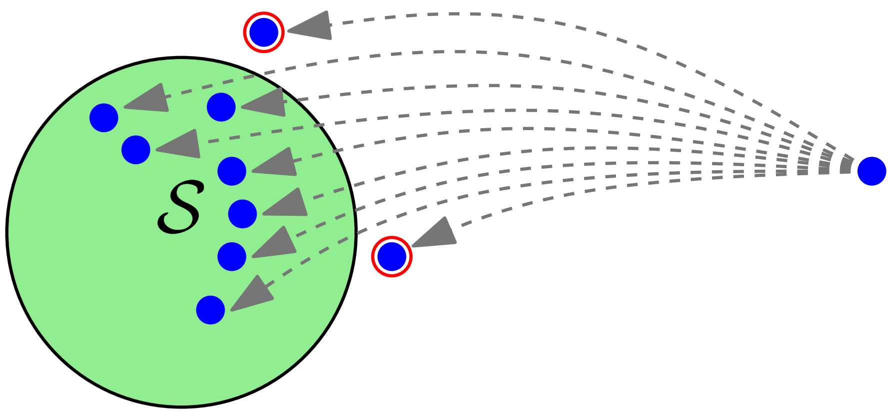



● := state of the system
--- := random trajectory

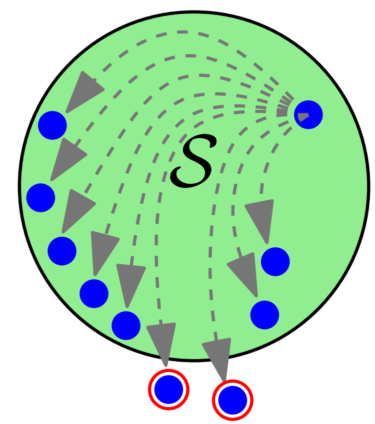
# Self-stabilizing?

Let $\mathcal{S} = \{\text{config. with max load } \mathcal{O}(\log n)\}$
Is the repeated balls into bins self-stabilizing?

Convergence? NO.




Closure? NO.




● := state of the system
- - - := random trajectory

...but almost!

# “ ” Self Stabilization

- **pseudo self-stabilization**: the system is allowed to deviate from legitimate states for a finite amount of time;
- $k$**-self-stabilization**: all allowed initial states are those from which a legitimate state of the system can be reached by changing the state of at most $k$ agents;
- **probabilistic self-stabilization**: randomized strategies for self-stabilization are allowed;
- **weak self-stabilization**: only requires the existence of an execution that eventually converges to a legitimate state.
- **randomized self-stabilization**: the expected number of rounds needed to reach a correct state is bounded by some constant $k$.

# “ ” Self Stabilization

- **pseudo self-stabilization**: the system is allowed to deviate from legitimate states for a finite amount of time;
- ***k*-self-stabilization**: all allowed initial states are those from which a legitimate state of the system can be reached by changing the state of at most $k$ agents;
- **probabilistic self-stabilization**: randomized strategies for self-stabilization are allowed;
- **weak self-stabilization**: only requires the existence of an execution that eventually converges to a legitimate state.
- **randomized self-stabilization**: the expected number of rounds needed to reach a correct state is bounded by some constant $k$.

- **self-stabilization w.h.p.**: convergence and closure are guaranteed only with high probability (fails with prob $n^{-\Theta(1)}$).

# The Infamous Stochastic Dependence

Stochastic dependence in balls-into-bins:
negative association,
Poisson approximation...

# The Infamous Stochastic Dependence

Stochastic dependence in balls-into-bins:
negative association,
Poisson approximation...

Stochastic dependence in repeated balls-into-bins:

...?

# The Infamous Stochastic Dependence

Stochastic dependence in balls-into-bins:
negative association,
Poisson approximation...

Stochastic dependence in repeated balls-into-bins:

...?

A *coupling w.h.p.*: the tetris process

# The Infamous Stochastic Dependence

Stochastic dependence in balls-into-bins:
negative association,
Poisson approximation...

Stochastic dependence in repeated balls-into-bins:

...?

A *coupling w.h.p.*: the tetris process

$M_t^{(RBB)}$ := time $t$ max. load in repeated b.i.b.
$M_t^{(T)}$ := time $t$ max. load in tetris proc.

$$\Pr(M_t^{(RBB)} \geq k) \leq \Pr(M_t^{(T)} \geq k) + t \cdot e^{-\Theta(n)}$$

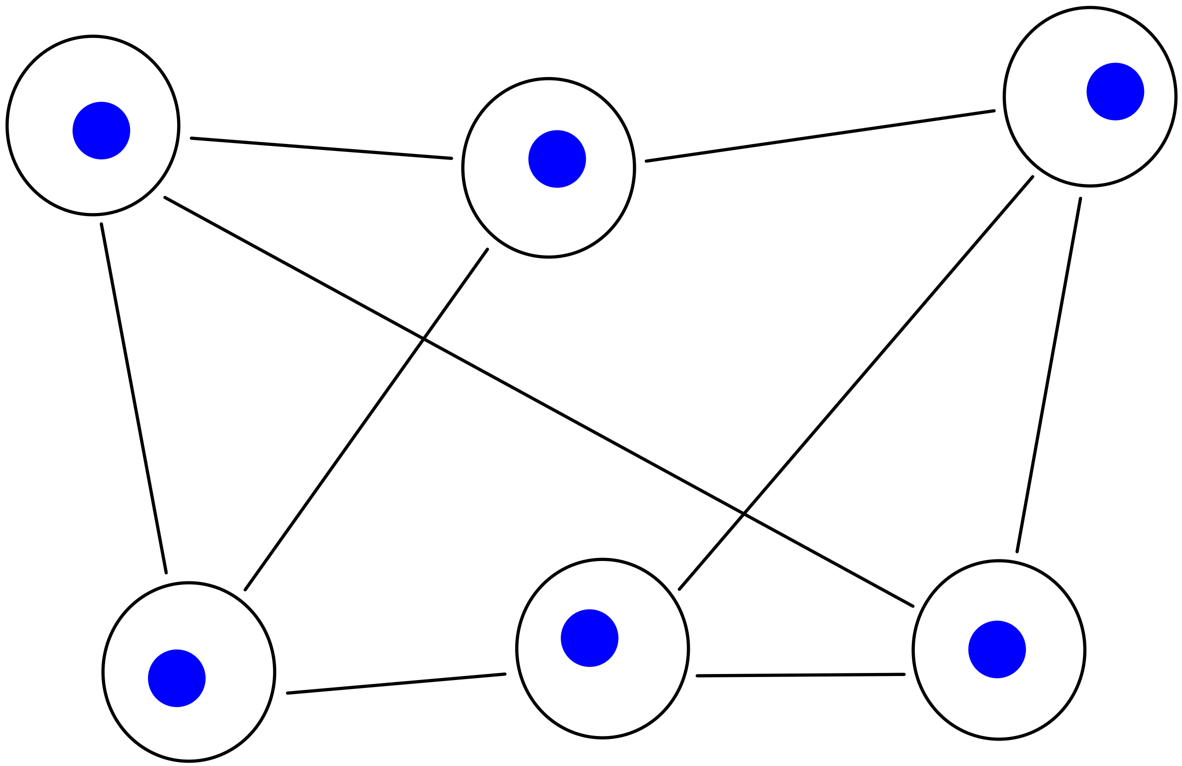
# Our Contribution [ACM SPAA '15]

From any configuration, in $\mathcal{O}(n)$ rounds the repeated balls-into-bins process reaches a conf. with max load $\mathcal{O}(\log n)$ w.h.p. and, from any conf. with max load $\mathcal{O}(\log n)$, the max load keeps $\mathcal{O}(\log n)$ for poly$(n)$ rounds w.h.p.

# B.i.B. & $\mathcal{GOSSIP}$ Random Walks

Goal: keep max load below $\mathcal{O}(\log n)$.

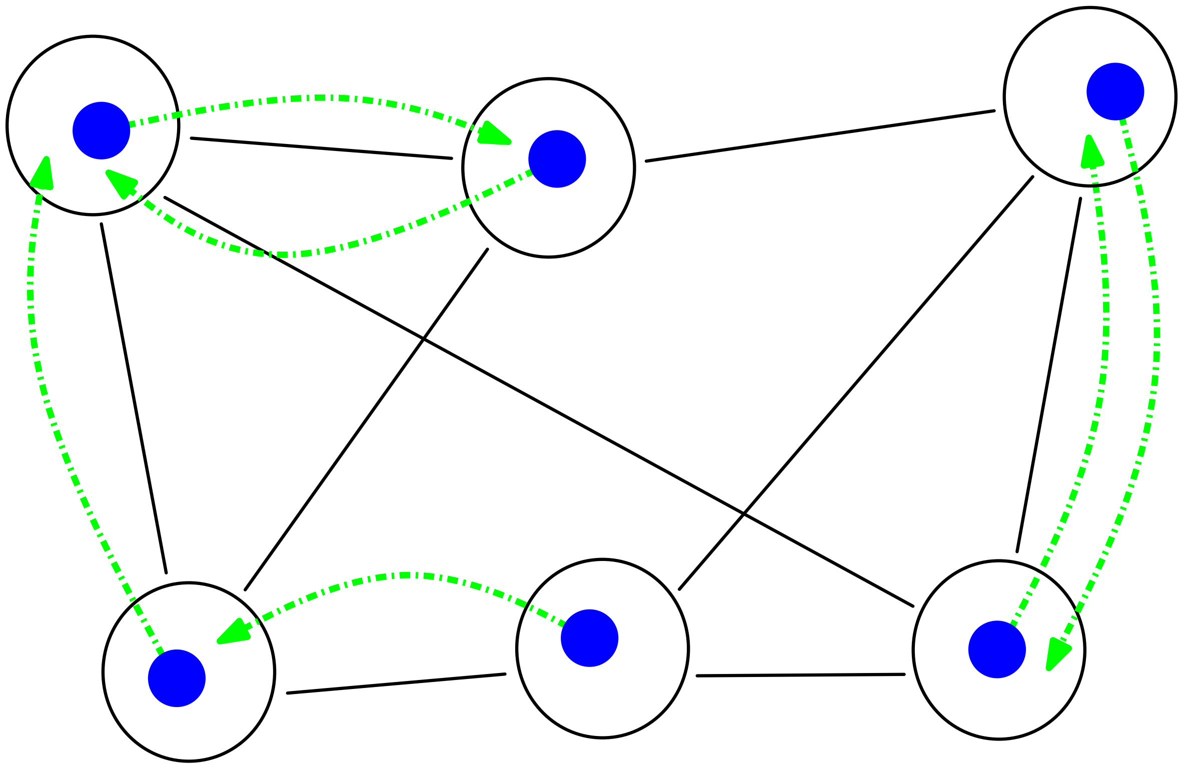
max # of tokens on each node



# B.i.B. & $\mathcal{GOSSIP}$ Random Walks

Goal: keep max load below $\mathcal{O}(\log n)$.

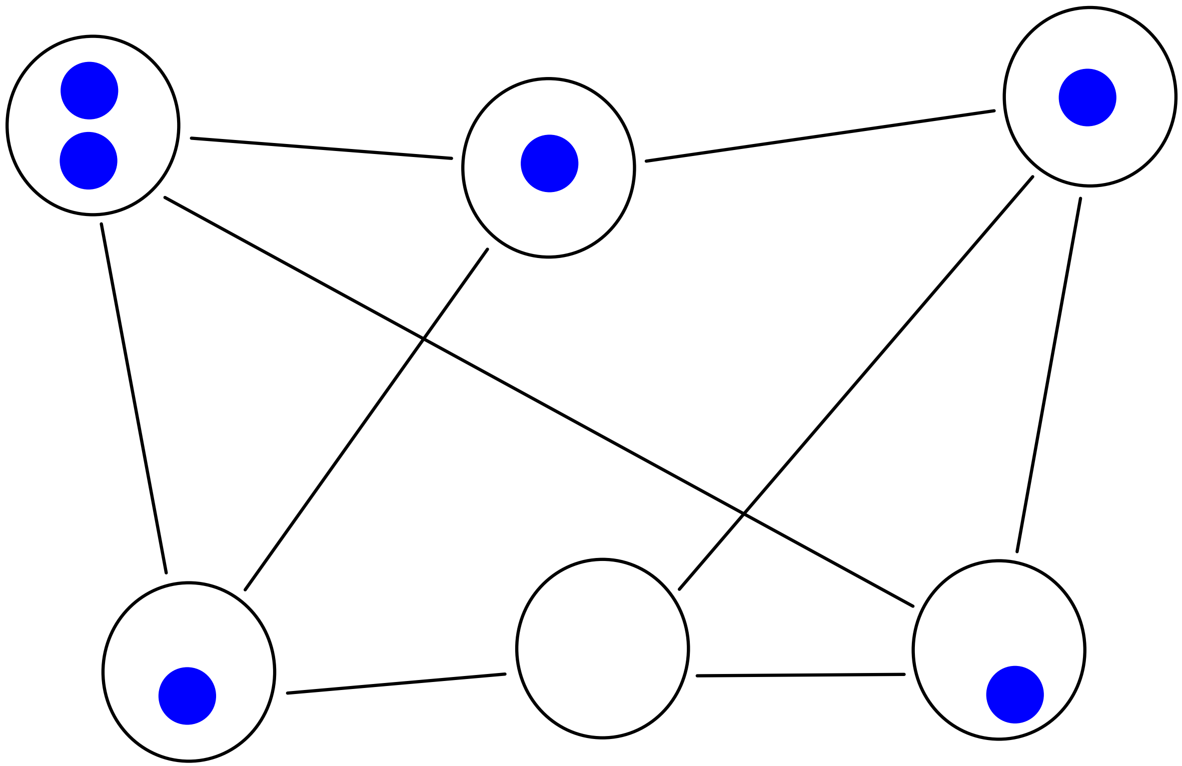
Simple random walks: max load $\mathcal{O}(\log n)$ w.h.p.



# B.i.B. & $\mathcal{GOSSIP}$ Random Walks

Goal: keep max load below $\mathcal{O}(\log n)$.

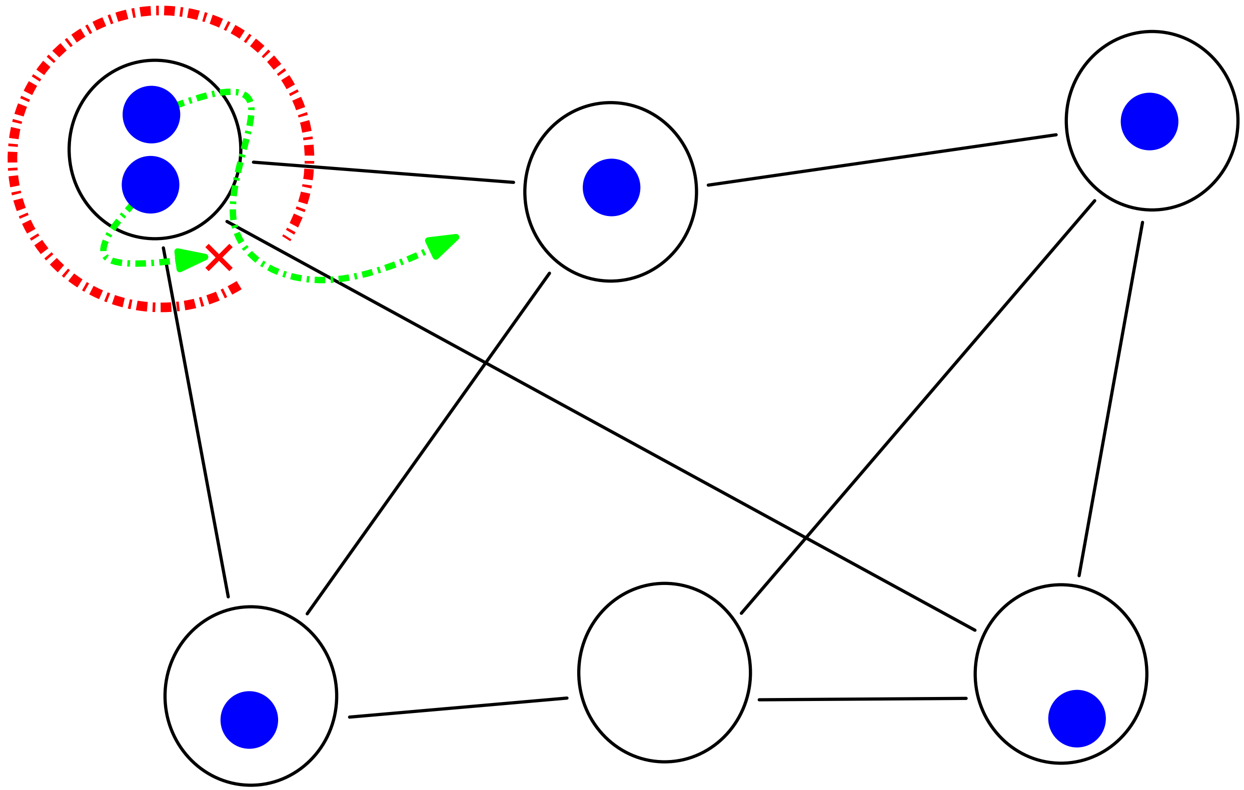
Simple random walks: max load $\mathcal{O}(\log n)$ w.h.p.



# B.i.B. & $\mathcal{GOSSIP}$ Random Walks

Goal: keep max load below $\mathcal{O}(\log n)$.

Simple random walks: max load $\mathcal{O}(\log n)$ w.h.p.

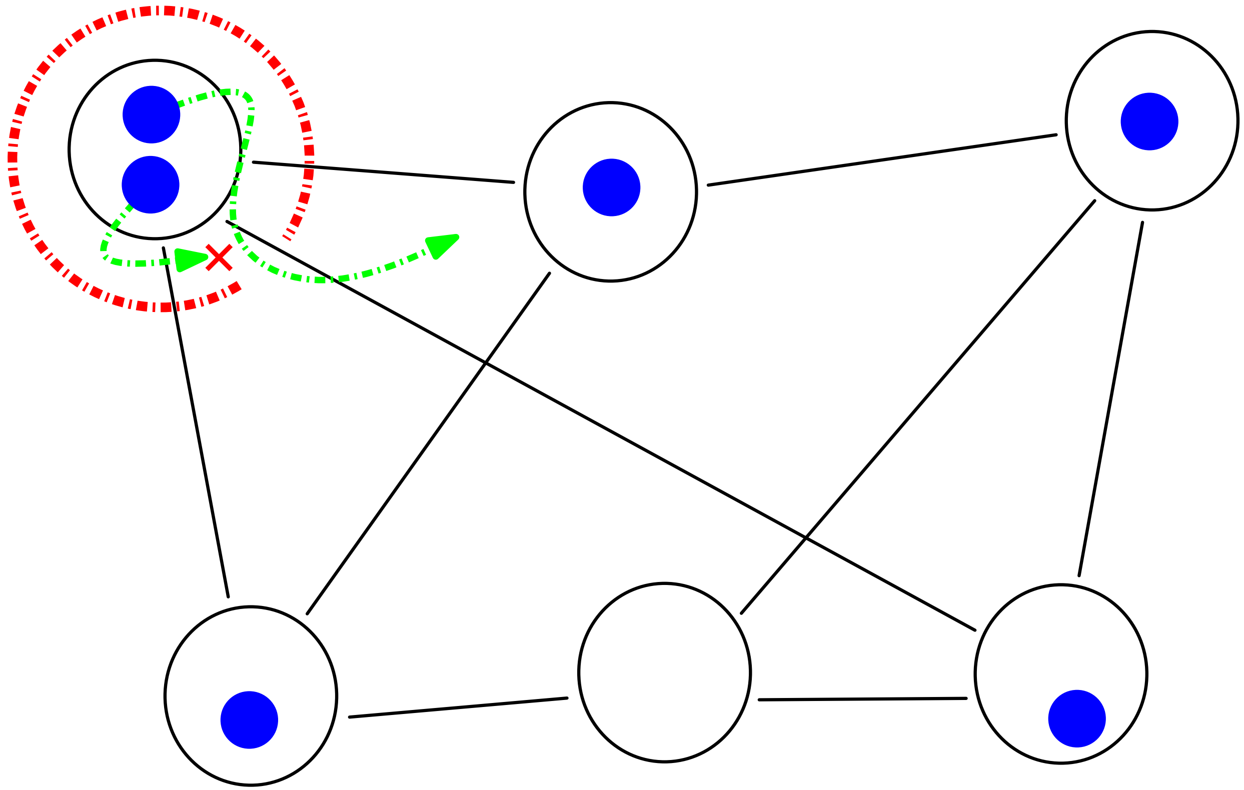


$\mathcal{GOSSIP}$ model [Censor-Hillel et al. '12]: only one token moves from each node (limited communication).

# B.i.B. & $\mathcal{GOSSIP}$ Random Walks

Goal: keep max load below $\mathcal{O}(\log n)$.

Simple random walks: max load $\mathcal{O}(\log n)$ w.h.p.



$\mathcal{GOSSIP}$ model [Censor-Hillel et al. '12]: only one token moves from each node (limited communication).

Max load of $\mathcal{GOSSIP}$ random walks: $\mathcal{O}(\log n)$?

# Some Related Work

Information exchange in phone-call model [Berenbrink et al. 2010, Elsässer et al. 2015]: analysis for polylog($n$) rounds.

# Some Related Work

Information exchange in phone-call model [Berenbrink et al. 2010, Elsässer et al. 2015]: analysis for polylog($n$) rounds.

Mixing time on regular expanders [Becchetti et al. 2015]: maximum load $\sqrt{t}$ ($t$ rounds).

# Some Related Work

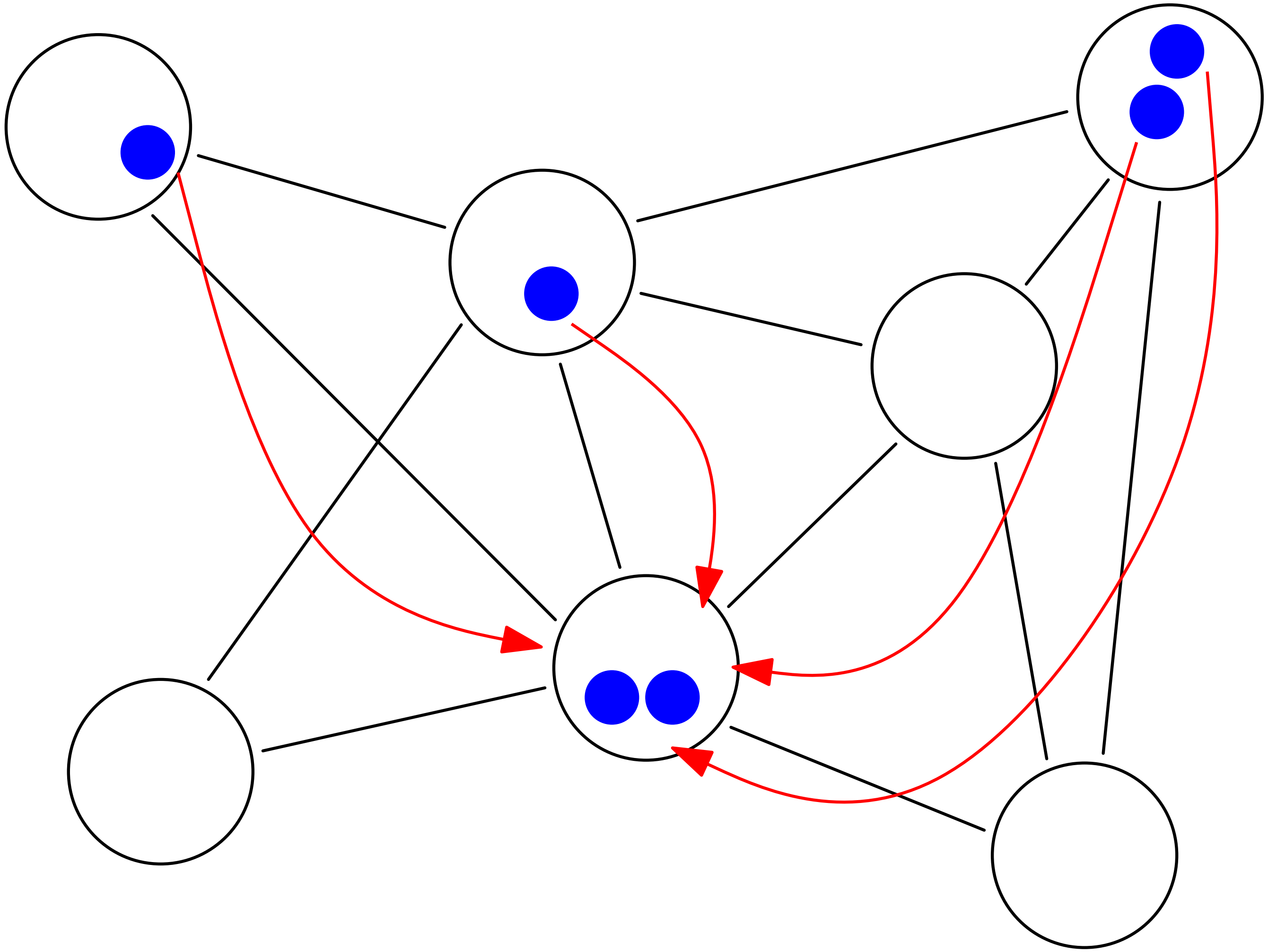
Information exchange in phone-call model [Berenbrink et al. 2010, Elsässer et al. 2015]: analysis for polylog($n$) rounds.

Mixing time on regular expanders [Becchetti et al. 2015]: maximum load $\sqrt{t}$ ($t$ rounds).

Closed Jackson networks in queueing theory: asynchronous version of $\mathcal{GOSSIP}$ r.w.s (admits closed form solution).

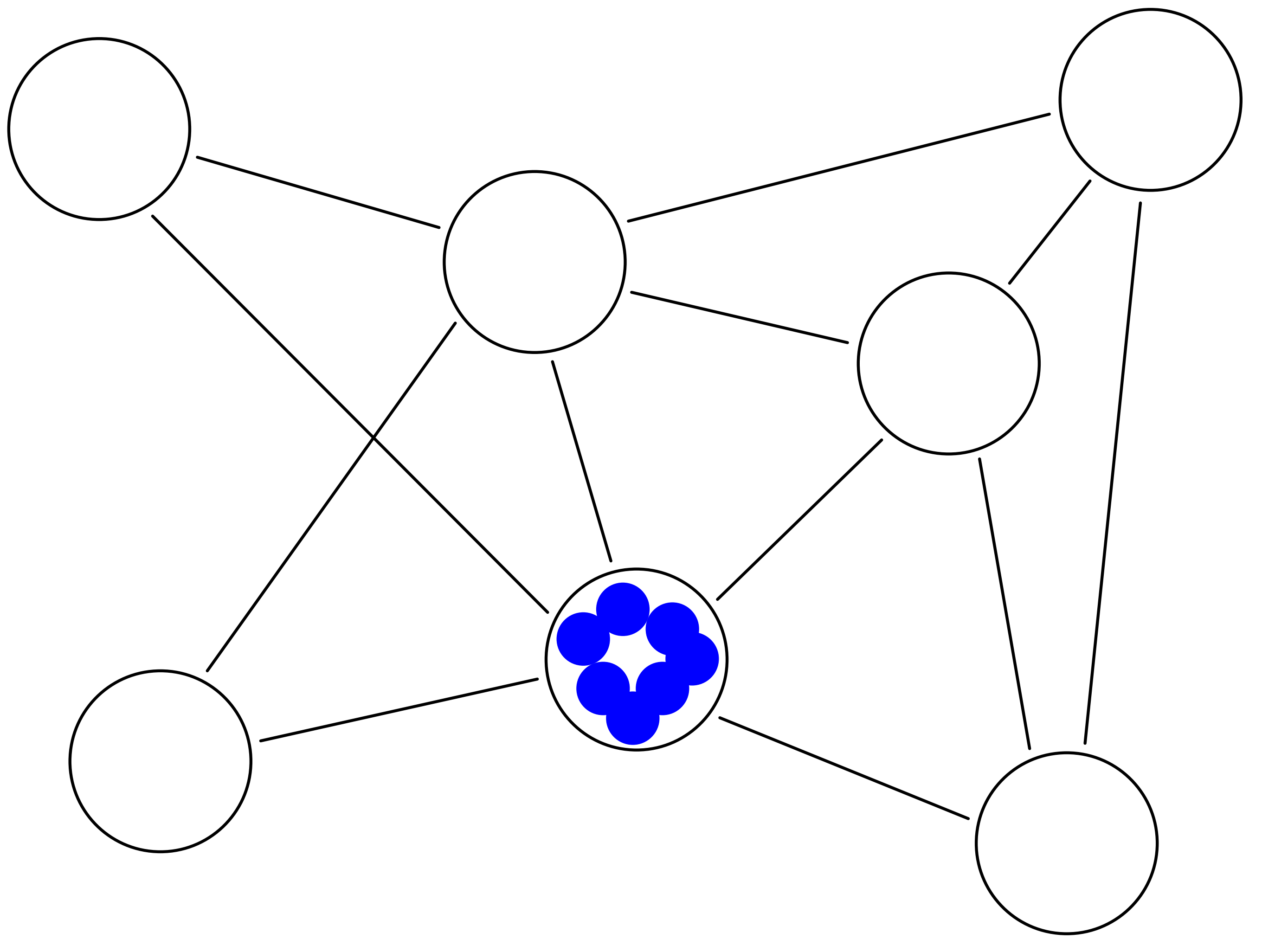
# Adversarial Model

Every $\Omega(n)$ rounds: the adversary move the tokens
(cfr Adversarial Queuing Theory [Borodin et al., '01])



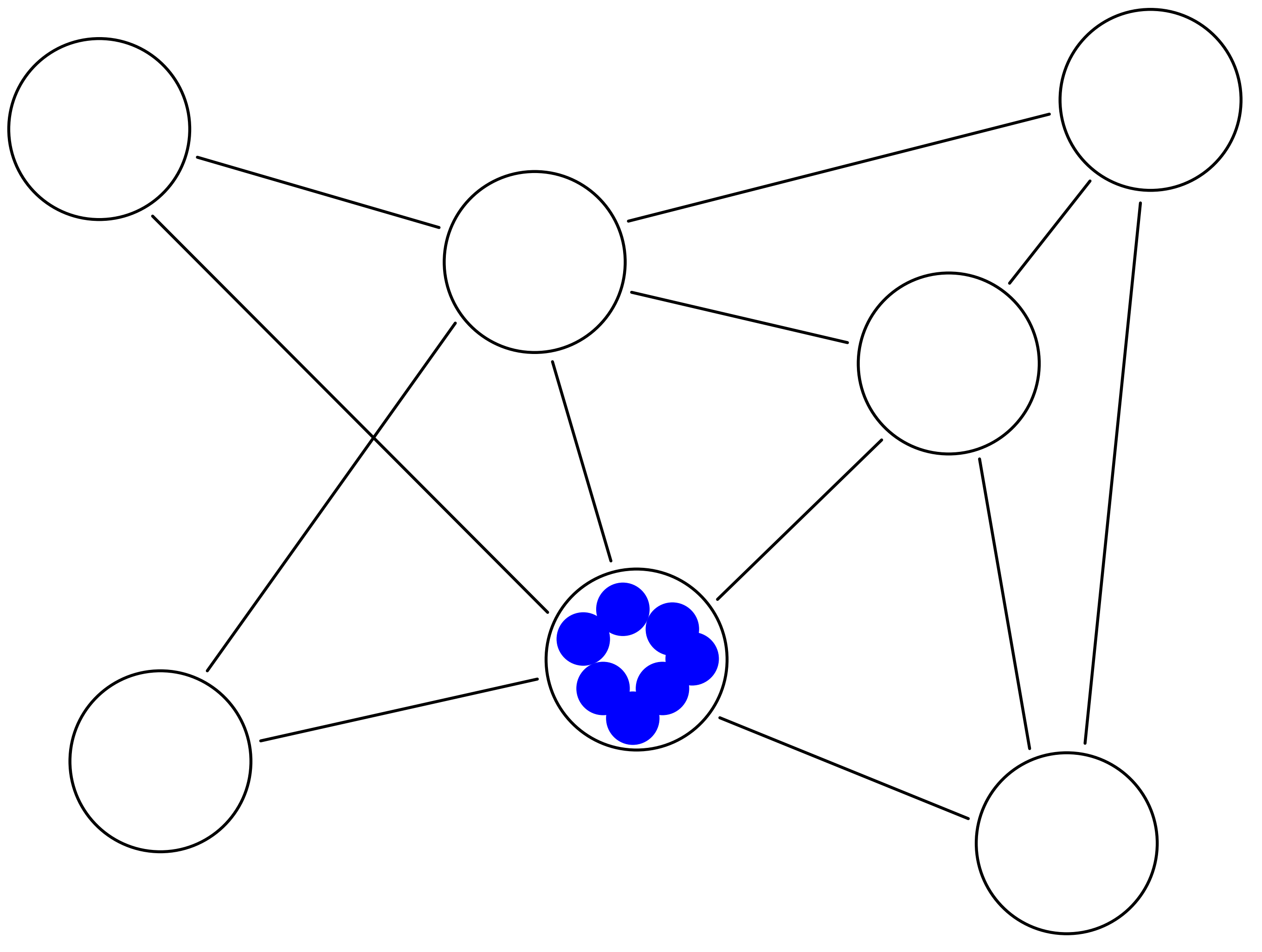
# Adversarial Model

Every $\Omega(n)$ rounds: the adversary move the tokens
(cfr Adversarial Queuing Theory [Borodin et al., '01])



# Adversarial Model

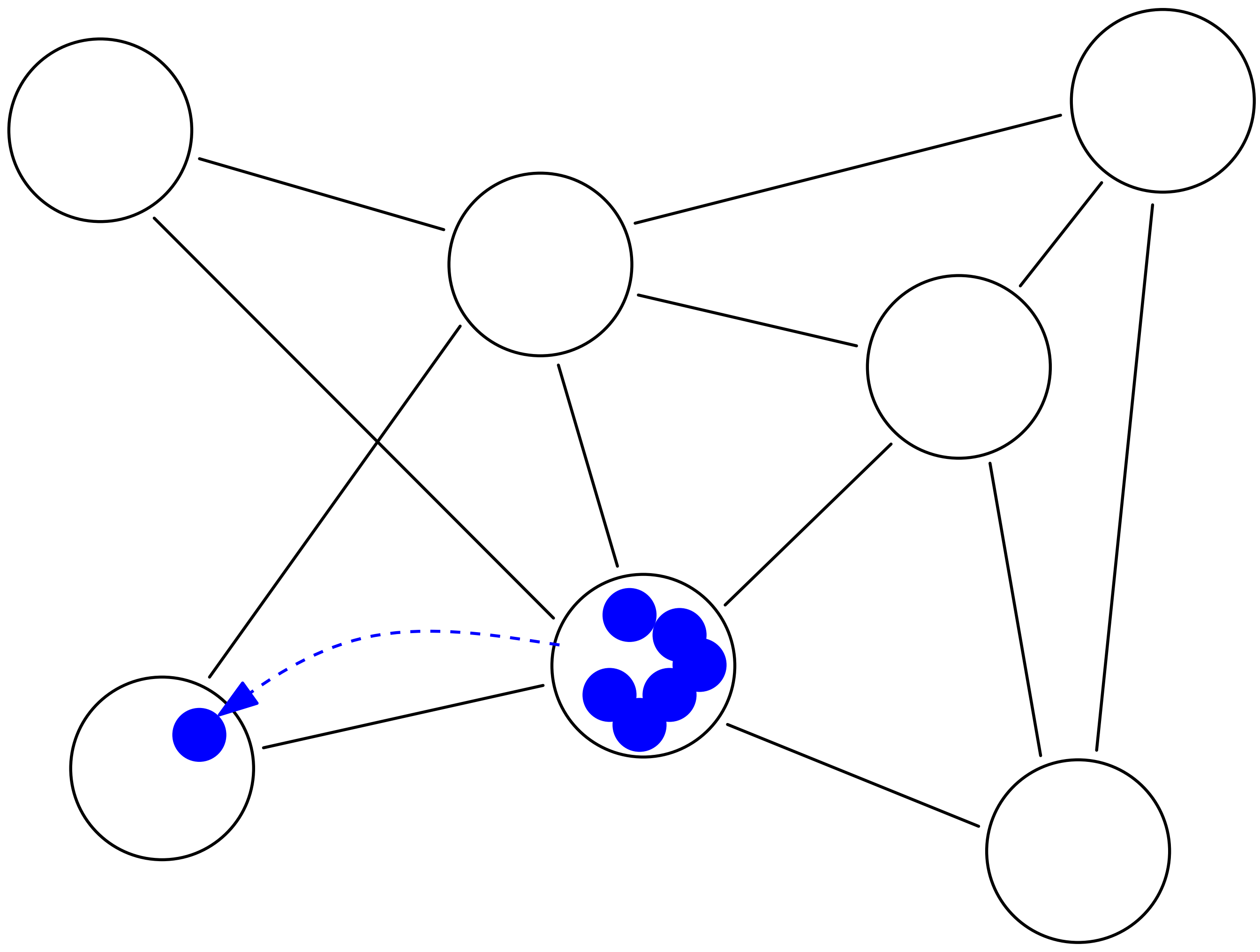
Every $\Omega(n)$ rounds: the adversary move the tokens
(cfr Adversarial Queuing Theory [Borodin et al., '01])



How many rounds until the load becomes small?

# Adversarial Model

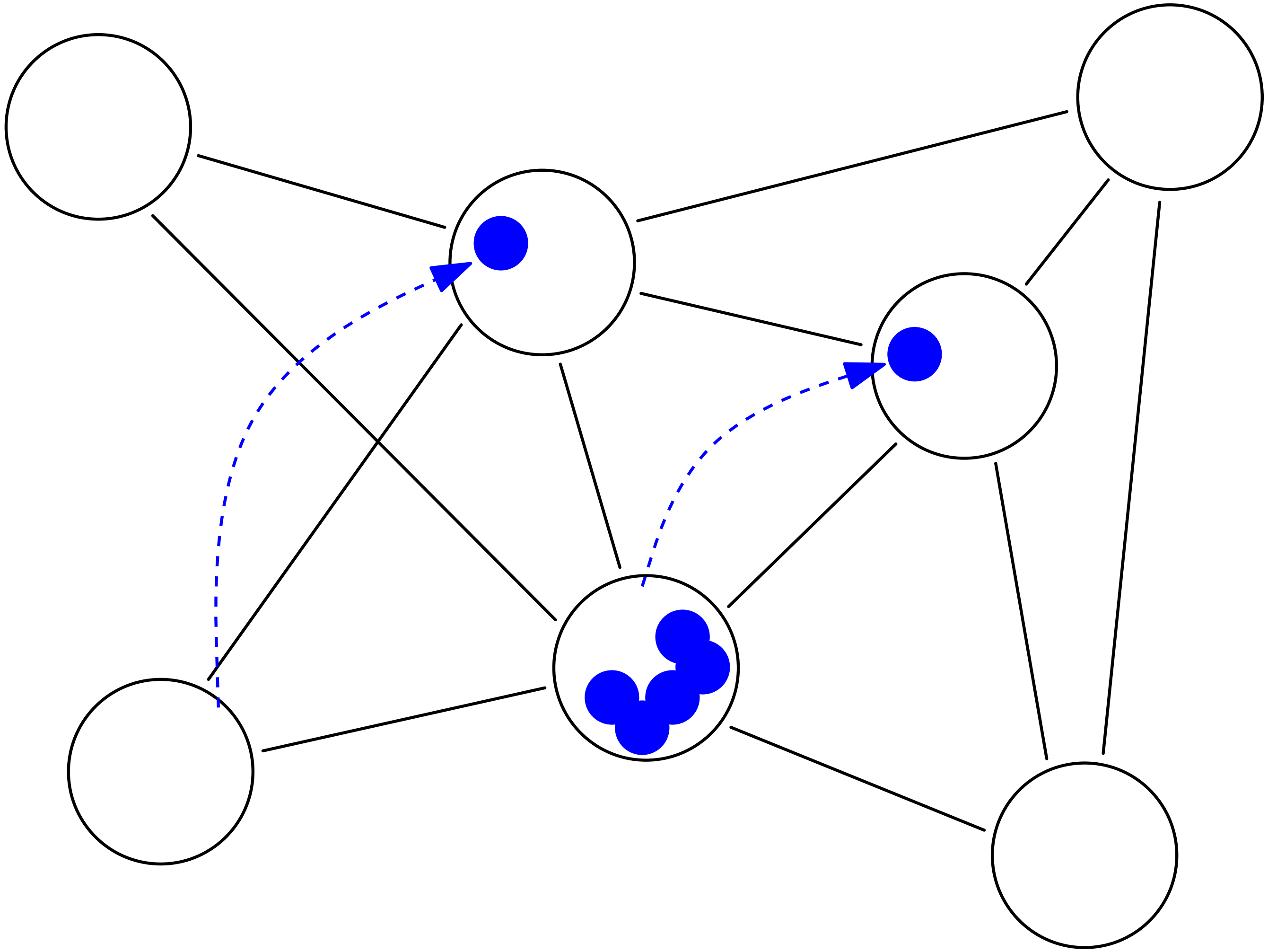
Every $\Omega(n)$ rounds: the adversary move the tokens
(cfr Adversarial Queuing Theory [Borodin et al., '01])



How many rounds until the load becomes small?

# Adversarial Model

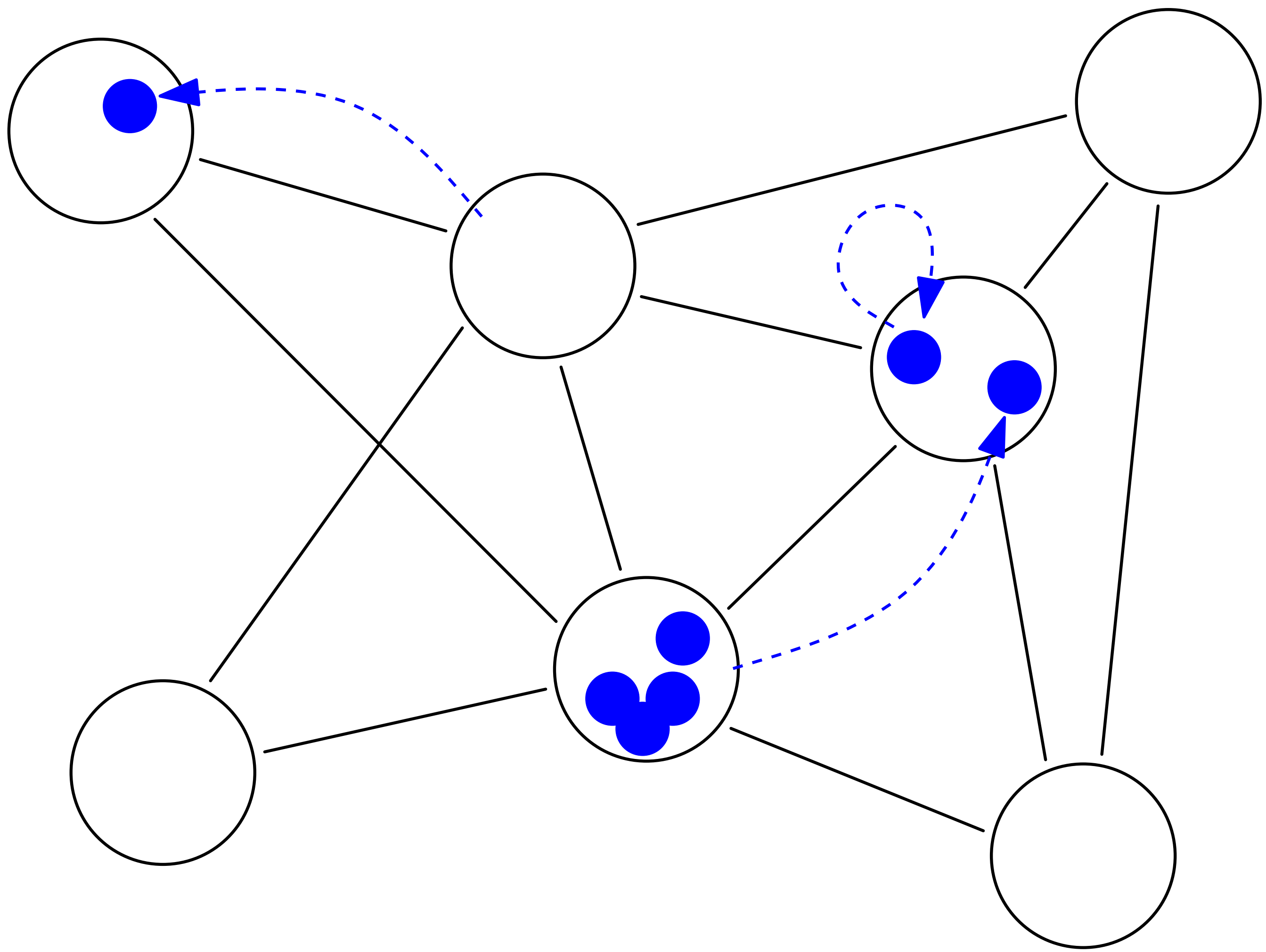
Every $\Omega(n)$ rounds: the adversary move the tokens
(cfr Adversarial Queuing Theory [Borodin et al., '01])



How many rounds until the load becomes small?

# Adversarial Model

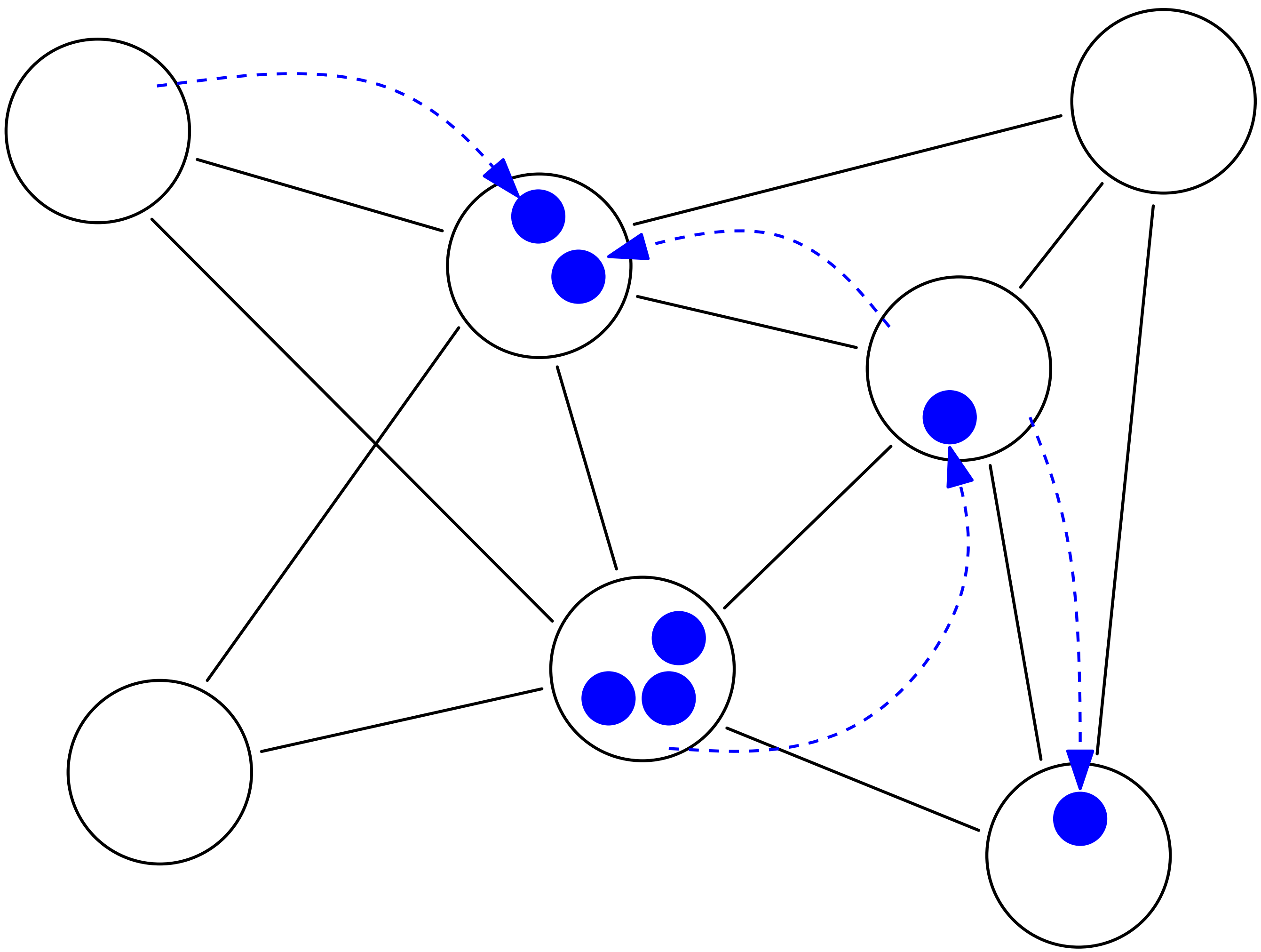
Every $\Omega(n)$ rounds: the adversary move the tokens
(cfr Adversarial Queuing Theory [Borodin et al., '01])



How many rounds until the load becomes small?

# Adversarial Model

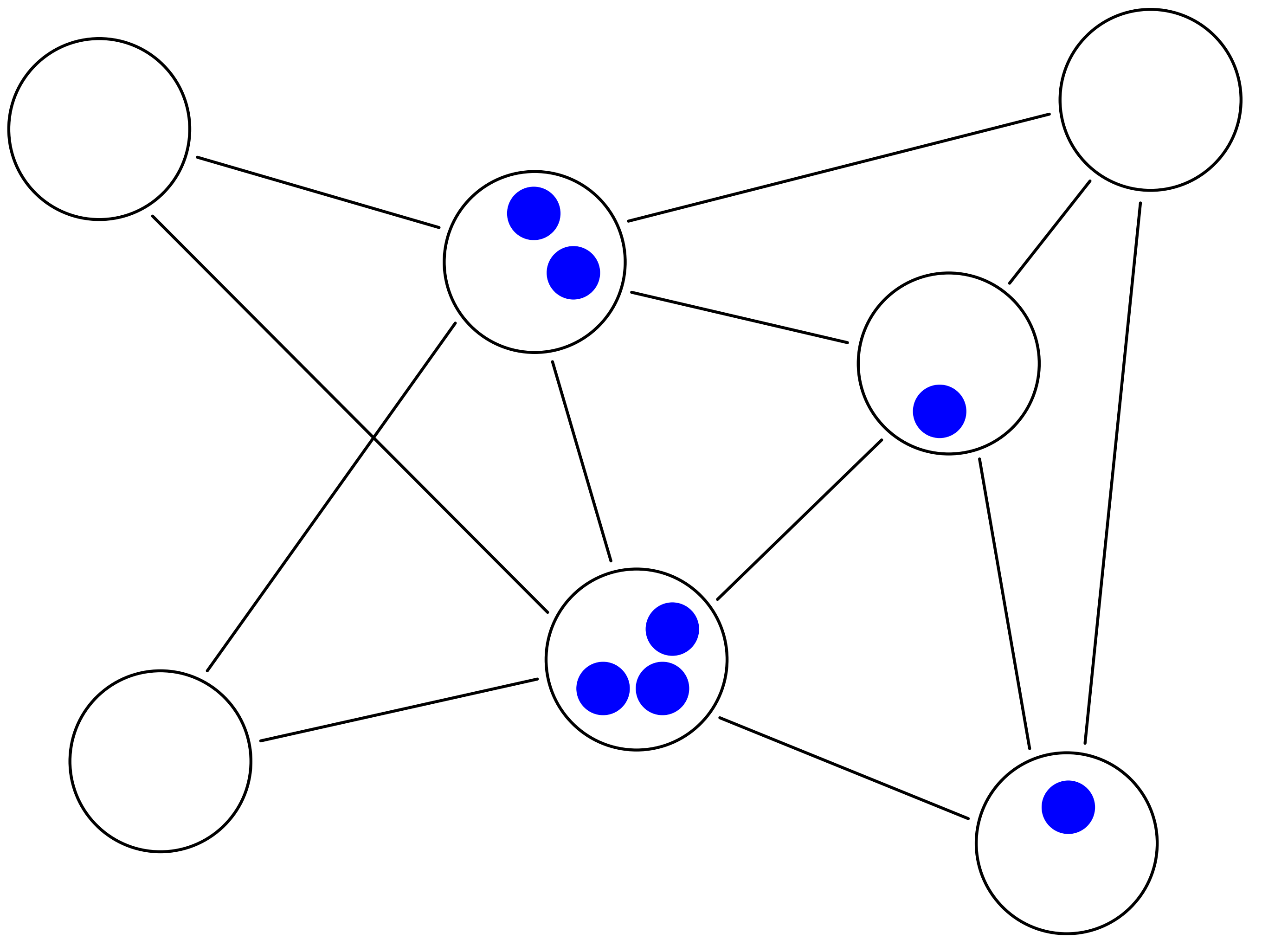
Every $\Omega(n)$ rounds: the adversary move the tokens
(cfr Adversarial Queuing Theory [Borodin et al., '01])



How many rounds until the load becomes small?

# Adversarial Model

Every $\Omega(n)$ rounds: the adversary move the tokens
(cfr Adversarial Queuing Theory [Borodin et al., '01])



How many rounds until the load becomes small?

# Our Contribution

From any configuration, in $\mathcal{O}(n)$ rounds the repeated balls-into-bins process reaches a conf. with max load $\mathcal{O}(\log n)$ w.h.p. and, from any conf. with max load $\mathcal{O}(\log n)$, the max load keeps $\mathcal{O}(\log n)$ for poly$(n)$ rounds w.h.p.

Repeated $n$ balls in $n$ bins $=$
$n$ $\mathcal{GOSSIP}$ r.w.s on $n$-node complete graph
(with loops)

# Our Contribution

From any configuration, in $\mathcal{O}(n)$ rounds the repeated balls-into-bins process reaches a conf. with max load $\mathcal{O}(\log n)$ w.h.p. and, from any conf. with max load $\mathcal{O}(\log n)$, the max load keeps $\mathcal{O}(\log n)$ for poly$(n)$ rounds w.h.p.

Repeated $n$ balls in $n$ bins $=$
$n$ $\mathcal{GOSSIP}$ r.w.s on $n$-node complete graph (with loops)

**Corollary**
After at most $\mathcal{O}(n)$ rounds the max. load of $n$ $\mathcal{GOSSIP}$ r.w.s on $n$-node complete graph is $\mathcal{O}(\log n)$ w.h.p., and keeps $\mathcal{O}(\log n)$ for poly$(n)$ rounds.

# Conclusions

Probabilistic self-stabilization is a fruitful concept in investigating fault tolerant algorithms that succeed with high probability.

**Research Direction**

Re-work the theory of self-stabilization under the "w.h.p.-relaxation":
simplify old solutions & solve old open problems.

# Open Questions

$\mathcal{GOSSIP}$ **random walks**

Maximum load on other topologies?

On regular graphs?

On the ring?

# Open Questions

$\mathcal{GOSSIP}$ **random walks**
Maximum load on other topologies?
On regular graphs?
On the ring?

**Repeated balls-into-bins**
Maximum load of repeated
balls-into-bins with $\omega(n)$ balls?
$\Theta(n \log n)$ balls?

Thank you!

# Tasks Assignment in the $\mathcal{GOSSIP}$ Model

Task assignment in mutual exclusion [Dijkstra '73]: Processors have to process the task , the task can be processed by only one processor at a time.

# Tasks Assignment in the $\mathcal{GOSSIP}$ Model

Task assignment in mutual exclusion [Dijkstra '73]: Processors have to process the tasks, the tasks can be processed by only one processor at a time.

# Tasks Assignment in the $\mathcal{GOSSIP}$ Model

Task assignment in mutual exclusion [Dijkstra '73]:
Processors have to process the tasks, the tasks can be processed by only one processor at a time.

$\Downarrow$

Random walks strategies

$\Downarrow$

(Parallel) cover time:
First round s.t. *each* task has been processed by each processor.

# Tasks Assignment in the $\mathcal{GOSSIP}$ Model

Task assignment in mutual exclusion [Dijkstra '73]: Processors have to process the tasks, the tasks can be processed by only one processor at a time.

$\Downarrow$

Random walks strategies

$\Downarrow$

(Parallel) cover time: First round s.t. *each* task has been processed by each processor.

What is the cover time of $n$ $\mathcal{GOSSIP}$ random walks?

# Tasks Assignment in the $\mathcal{GOSSIP}$ Model

Task assignment in mutual exclusion [Dijkstra '73]:
Processors have to process the tasks, the tasks can be processed by only one processor at a time.

$\Downarrow$

Random walks strategies

$\Downarrow$

(Parallel) cover time:
First round s.t. *each* task has been processed by each processor.

What is the cover time of $n$ $\mathcal{GOSSIP}$ random walks?

$\Downarrow$

**Corollary**
Cover time of $n$ $\mathcal{GOSSIP}$ r.w.s on $n$-node complete graph is $\mathcal{O}(n \log^2 n)$ w.h.p.

# Analysis of b.i.b. – Empty Bins

**Lemma**

At the next round $|\{\text{empty bins}\}| \geq \frac{n}{4}$ w.h.p.

# Analysis of b.i.b. – Empty Bins

**Lemma**

At the next round $|\{\text{empty bins}\}| \geq \frac{n}{4}$ w.h.p.

**Corollary**

At the next round $|\{\text{thrown balls}\}| \leq \frac{3n}{4}$ w.h.p.

# Analysis of b.i.b. – Empty Bins

**Lemma**

At the next round $|\{\text{empty bins}\}| \geq \frac{n}{4}$ w.h.p.

**Corollary**

At the next round $|\{\text{thrown balls}\}| \leq \frac{3n}{4}$ w.h.p.

*Proof*

$a := |\{\text{empty bins}\}|$, $b := |\{\text{bins with 1 ball}\}|$,
$X := |\{\text{new empty bins}\}|$

1. $\mathbb{E}[X] = (a+b)(1-1/n)^{n-a}$

# Analysis of b.i.b. – Empty Bins

**Lemma**

At the next round $|\{\text{empty bins}\}| \geq \frac{n}{4}$ w.h.p.

**Corollary**

At the next round $|\{\text{thrown balls}\}| \leq \frac{3n}{4}$ w.h.p.

*Proof*

$a := |\{\text{empty bins}\}|$, $b := |\{\text{bins with 1 ball}\}|$,
$X := |\{\text{new empty bins}\}|$

1. $\mathbb{E}[X] = (a+b)(1-1/n)^{n-a}$
2. $n-(a+b) \leq a \implies \mathbb{E}[X] \geq (1+\epsilon)\frac{n}{4}$

# Analysis of b.i.b. – Empty Bins

**Lemma**
At the next round $|\{\text{empty bins}\}| \geq \frac{n}{4}$ w.h.p.

**Corollary**
At the next round $|\{\text{thrown balls}\}| \leq \frac{3n}{4}$ w.h.p.

*Proof*

$a := |\{\text{empty bins}\}|$, $b := |\{\text{bins with 1 ball}\}|$,
$X := |\{\text{new empty bins}\}|$

1. $\mathbb{E}[X] = (a+b)(1-1/n)^{n-a}$
2. $n-(a+b) \leq a \implies \mathbb{E}[X] \geq (1+\epsilon)\frac{n}{4}$
3. Chernoff bound (negative association)

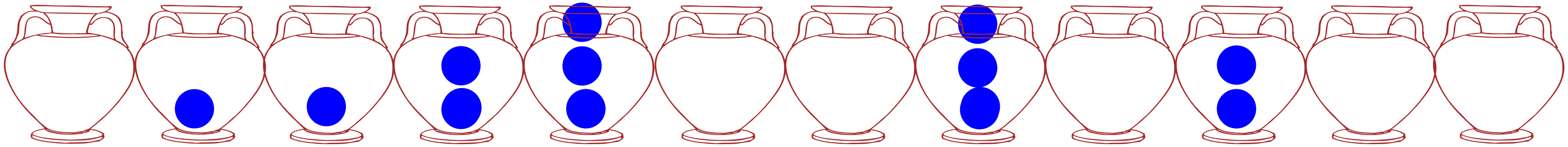
$\square$

# Analysis of b.i.b. – Tetris Process

**Tetris Process**

1- Throw away a ball from each non-empty bin

2- Throw $3n/4$ balls in the bins u.a.r.



# Analysis of b.i.b. – Tetris Process

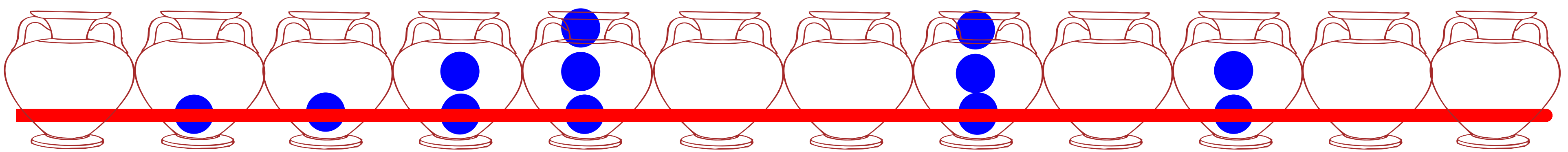
**Tetris Process**

1- Throw away a ball from each non-empty bin

2- Throw $3n/4$ balls in the bins u.a.r.

**Coupling**

Step 1: As rep. b.i.b., take one ball from each bin



# Analysis of b.i.b. – Tetris Process

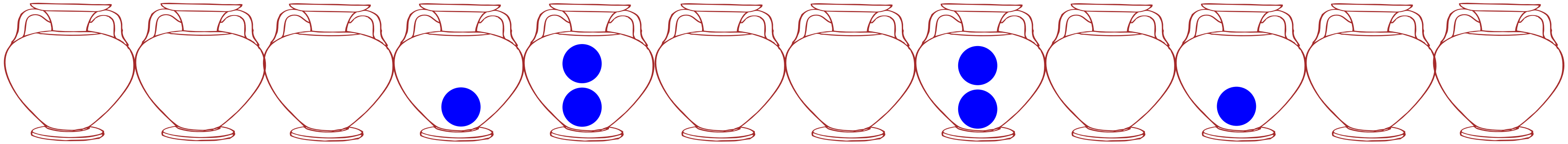
**Tetris Process**

1- Throw away a ball from each non-empty bin

2- Throw $3n/4$ balls in the bins u.a.r.

**Coupling**

Step 1: As rep. b.i.b., take one ball from each bin



# Analysis of b.i.b. – Tetris Process

**Tetris Process**

1- Throw away a ball from each non-empty bin

2- Throw $3n/4$ balls in the bins u.a.r.

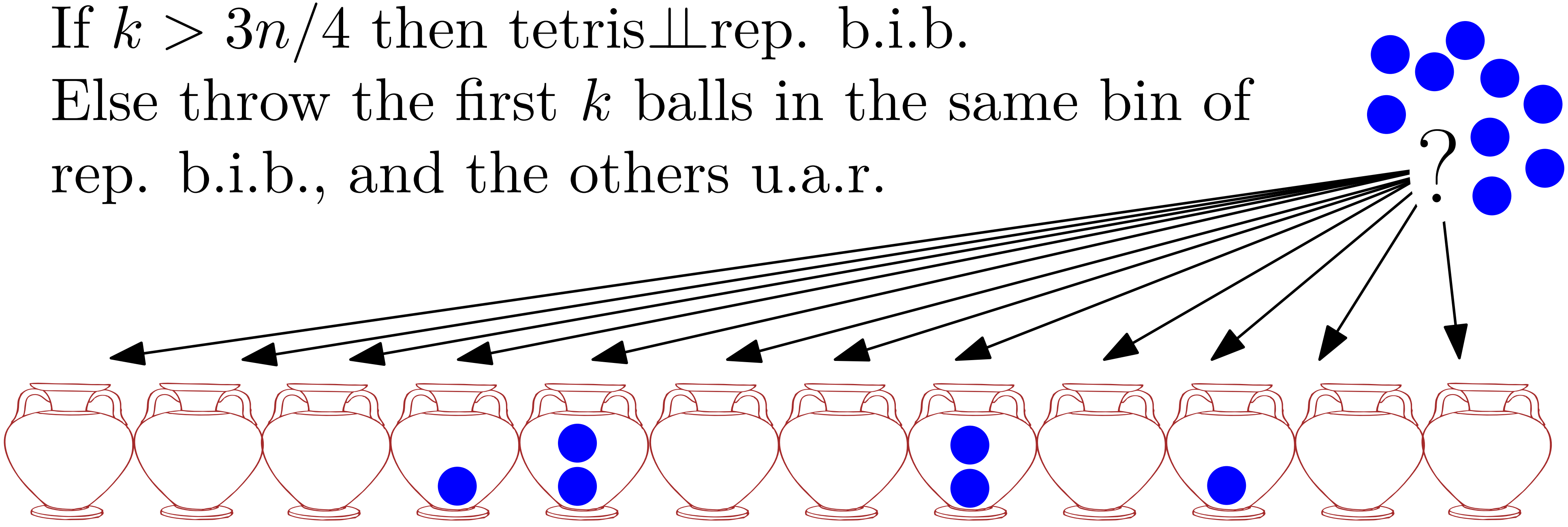
**Coupling**

Step 1: As rep. b.i.b., take one ball from each bin

Step 2: Let $k :=$ non-empty bins in rep. b.i.b.

If $k > 3n/4$ then tetris$\perp\!\!\!\perp$rep. b.i.b.

Else throw the first $k$ balls in the same bin of rep. b.i.b., and the others u.a.r.



# Analysis of b.i.b. – Tetris Process

**Tetris Process**

1- Throw away a ball from each non-empty bin

2- Throw $3n/4$ balls in the bins u.a.r.

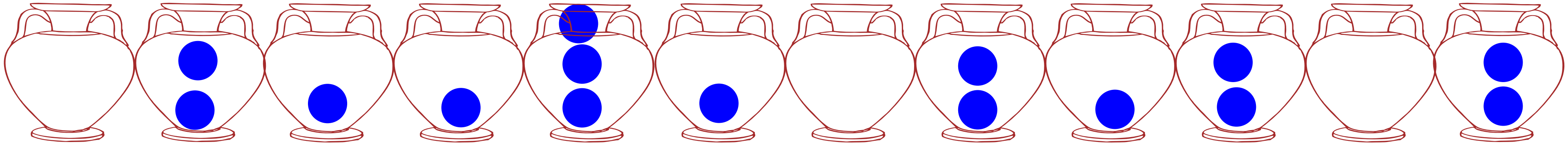
**Coupling**

Step 1: As rep. b.i.b., take one ball from each bin

Step 2: Let $k :=$ non-empty bins in rep. b.i.b.

If $k > 3n/4$ then tetris$\perp\!\!\!\perp$rep. b.i.b.

Else throw the first $k$ balls in the same bin of rep. b.i.b., and the others u.a.r.



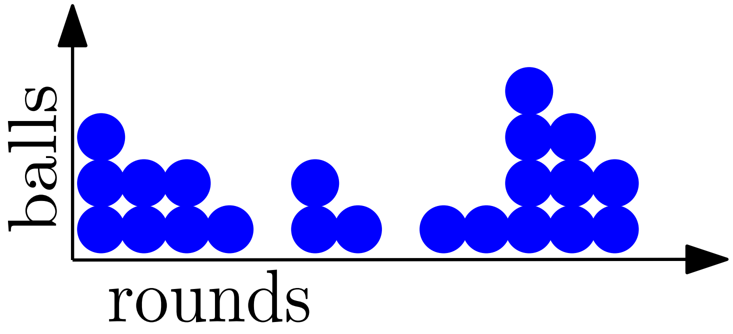
# Analysis of b.i.b. – Maximum Load

**Theorem**
The max. load of the tetris process is $\mathcal{O}(\log n)$ for poly$(n)$ rounds w.h.p.

# Analysis of b.i.b. – Maximum Load

**Theorem**
The max. load of the tetris process is $\mathcal{O}(\log n)$ for poly($n$) rounds w.h.p.

*Proof*

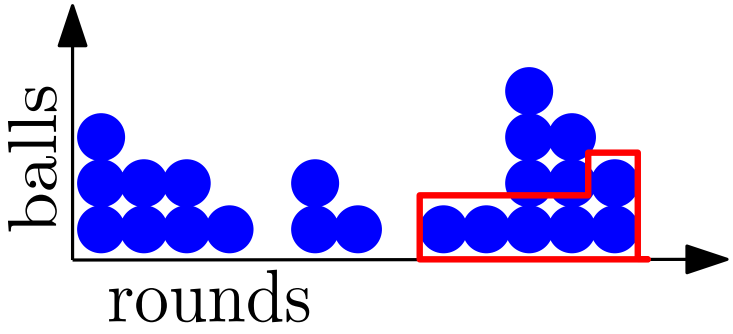



$$\mathbb{E}[\text{incoming balls in t rounds}] = \frac{3t}{4}$$

# Analysis of b.i.b. – Maximum Load

**Theorem**
The max. load of the tetris process is $\mathcal{O}(\log n)$ for poly$(n)$ rounds w.h.p.

*Proof*




$\mathbb{E}[\text{incoming balls in t rounds}] = \frac{3t}{4}$

$T :=$ # rounds from last time the bin was empty

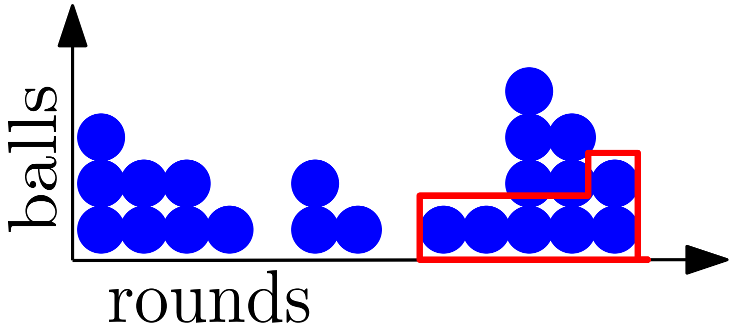
For each bin: load $k$ at round $t \implies$ received $k + T$ balls

$\square$

# Analysis of b.i.b. – Maximum Load

**Theorem**
The max. load of the tetris process is $\mathcal{O}(\log n)$ for poly$(n)$ rounds w.h.p.

*Proof*




$\mathbb{E}[\text{incoming balls in t rounds}] = \frac{3t}{4}$

$T :=$ # rounds from last time the bin was empty
For each bin: load $k$ at round $t \implies$ received $k + T$ balls

$\square$

**Lemma**
From any configuration, every bin in the tetris proc. is empty at least once every $5n$ rounds w.h.p.