Self-Stabilizing Broadcast with O(1)-Bit Messages*

Emanuele Natale†

joint work with

Lucas Boczkowski* and Amos Korman*

* CNRS IRIF INSTITUT DE RECHERCHE EN INFORMATIQUE FONDAMENTALE

† SAPIENZA UNIVERSITÀ DI ROMA

4th Workshop on Biological Distributed Algorithms (BDA)

July 25-29, 2016

Chicago, Illinois

*preprint at **goo.gl/ETNc64**

Self-Stabilizing Broadcast with O(1)-Bit Messages[*]
(Bit Dissemination)

Emanuele Natale[†]

joint work with
Lucas Boczkowski[*] and Amos Korman[*]

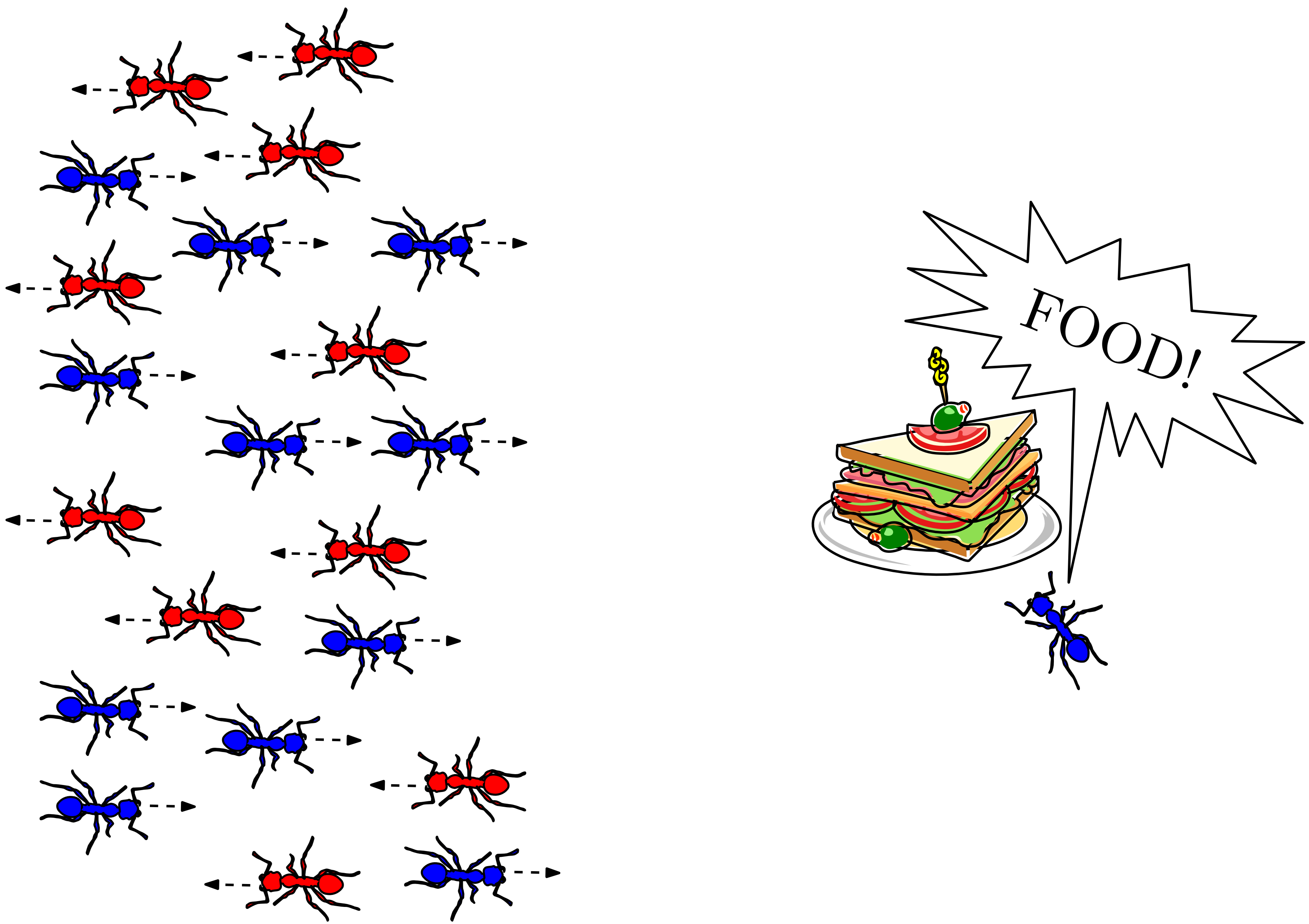
[*] CNRS IRIF INSTITUT DE RECHERCHE EN INFORMATIQUE FONDAMENTALE

[†] SAPIENZA UNIVERSITÀ DI ROMA

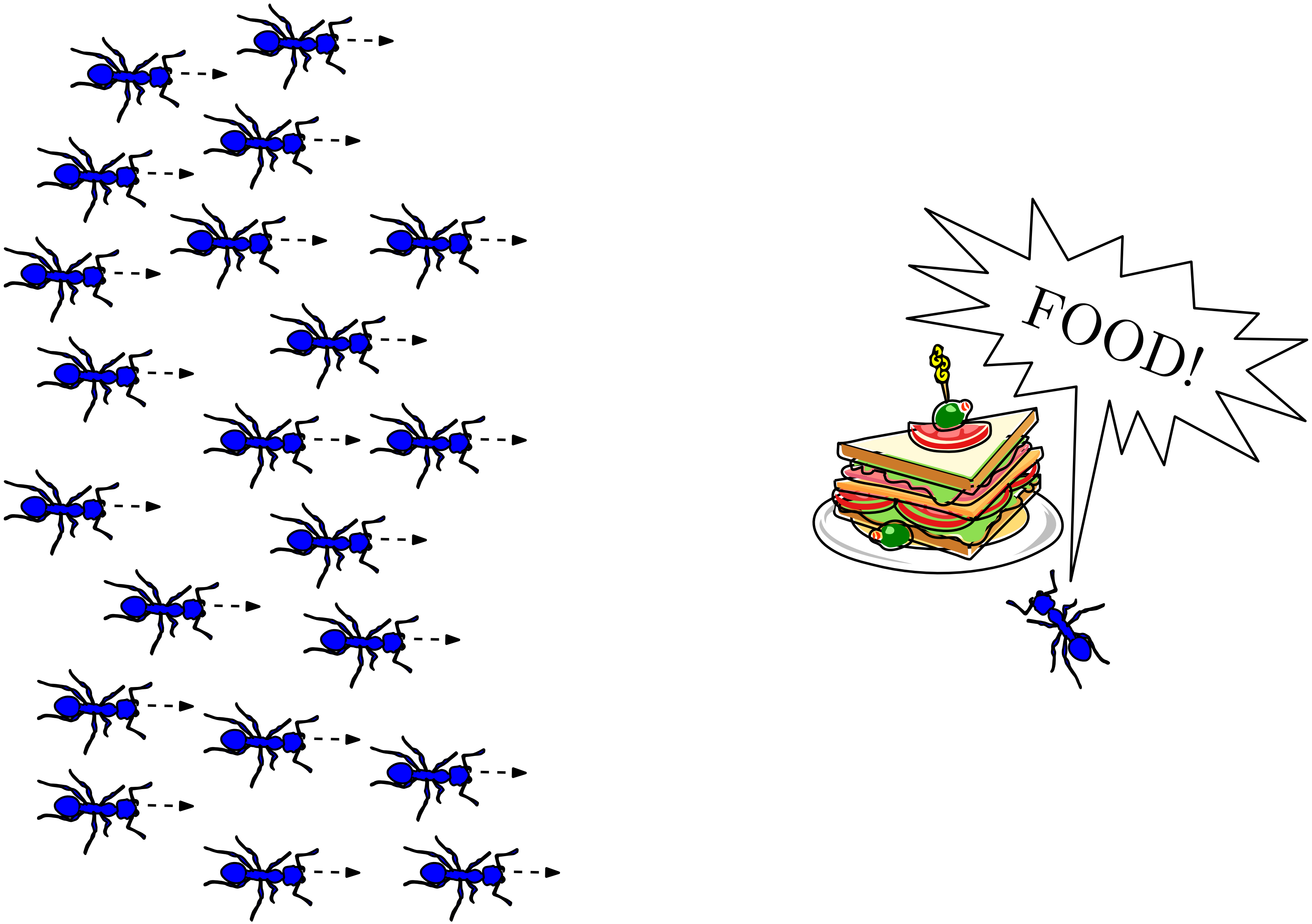
4th Workshop on Biological Distributed Algorithms (BDA)
July 25-29, 2016
Chicago, Illinois

[*]preprint at **goo.gl/ETNc64**

Bit Dissemination Problem



Bit Dissemination Problem



Examples

Flocks of birds
[Ben-Shahar et al. '10]

Examples

Flocks of birds
[Ben-Shahar et al. '10]

Schools of fish
[Sumpter et al. '08]

Examples

Flocks of birds
[Ben-Shahar et al. '10]

Schools of fish
[Sumpter et al. '08]

Insects colonies
[Franks et al. '02]

Communication Model

Animal communication:

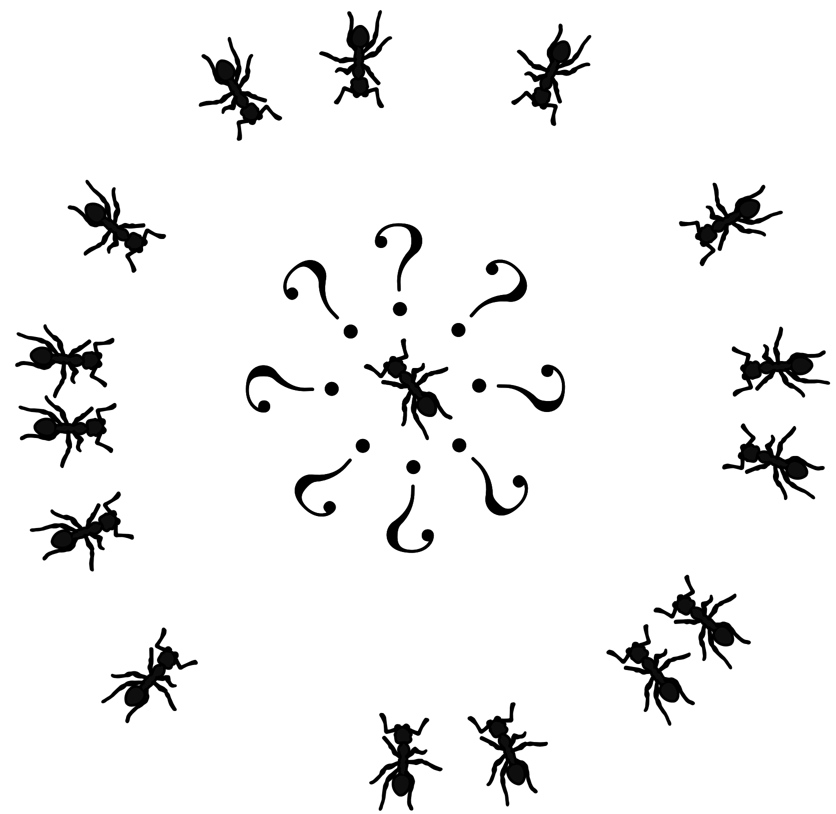
- Chaotic
- Anonymous
- Passive
- Parsimonious

Communication Model

Animal communication:

- Chaotic
- Anonymous
- Passive
- Parsimonious

$\mathcal{PULL}(h, \ell)$ **model [Demers '88]**: at each round each agent can *observe* h other agents chosen independently and uniformly at random, and *shows* ℓ *bits* to her observers.

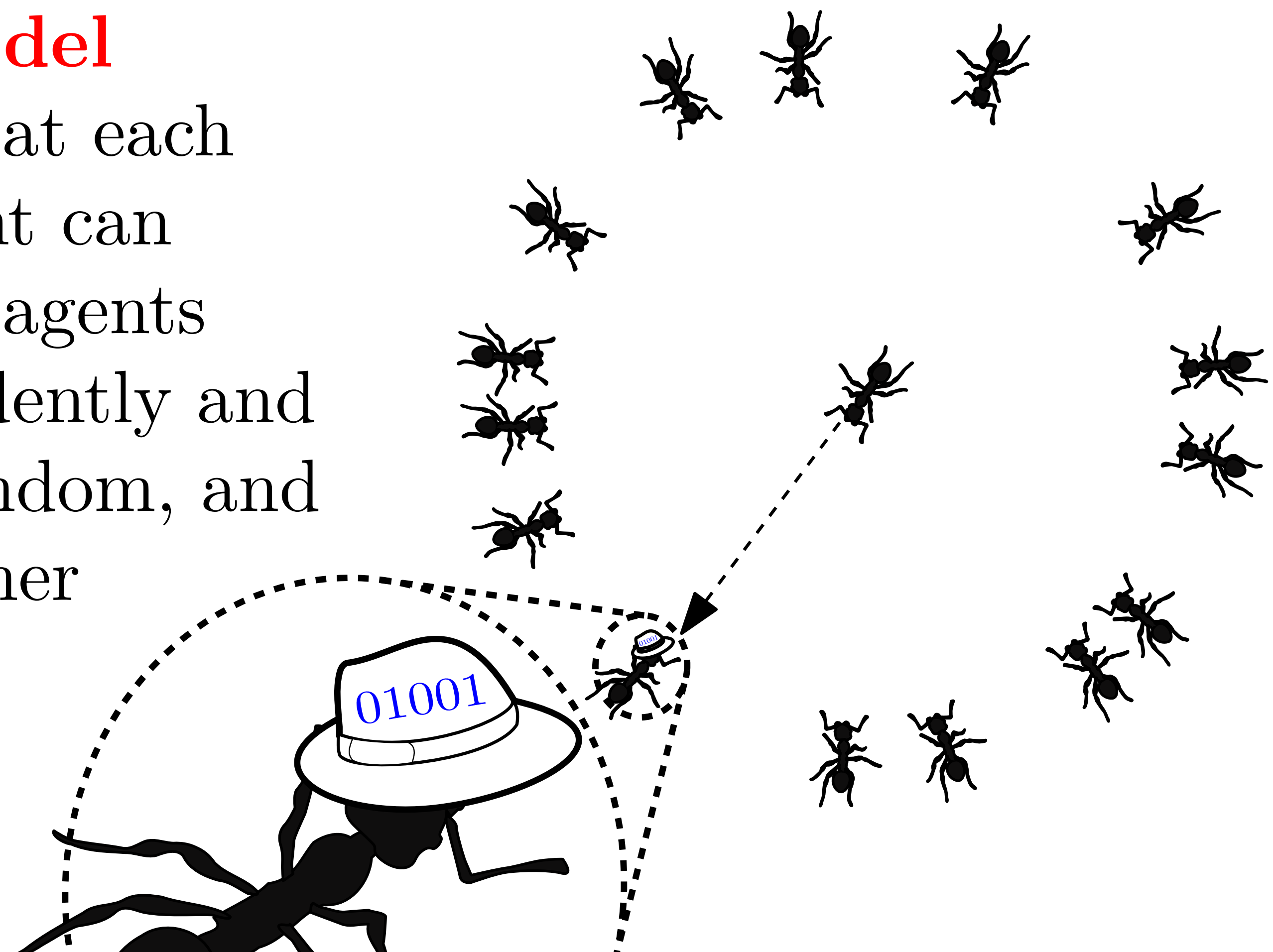


Communication Model

Animal communication:

- Chaotic
- Anonymous
- Passive
- Parsimonious

$\mathcal{PULL}(h,\ell)$ **model [Demers '88]**: at each round each agent can *observe* h other agents chosen independently and uniformly at random, and *shows* ℓ *bits* to her observers.

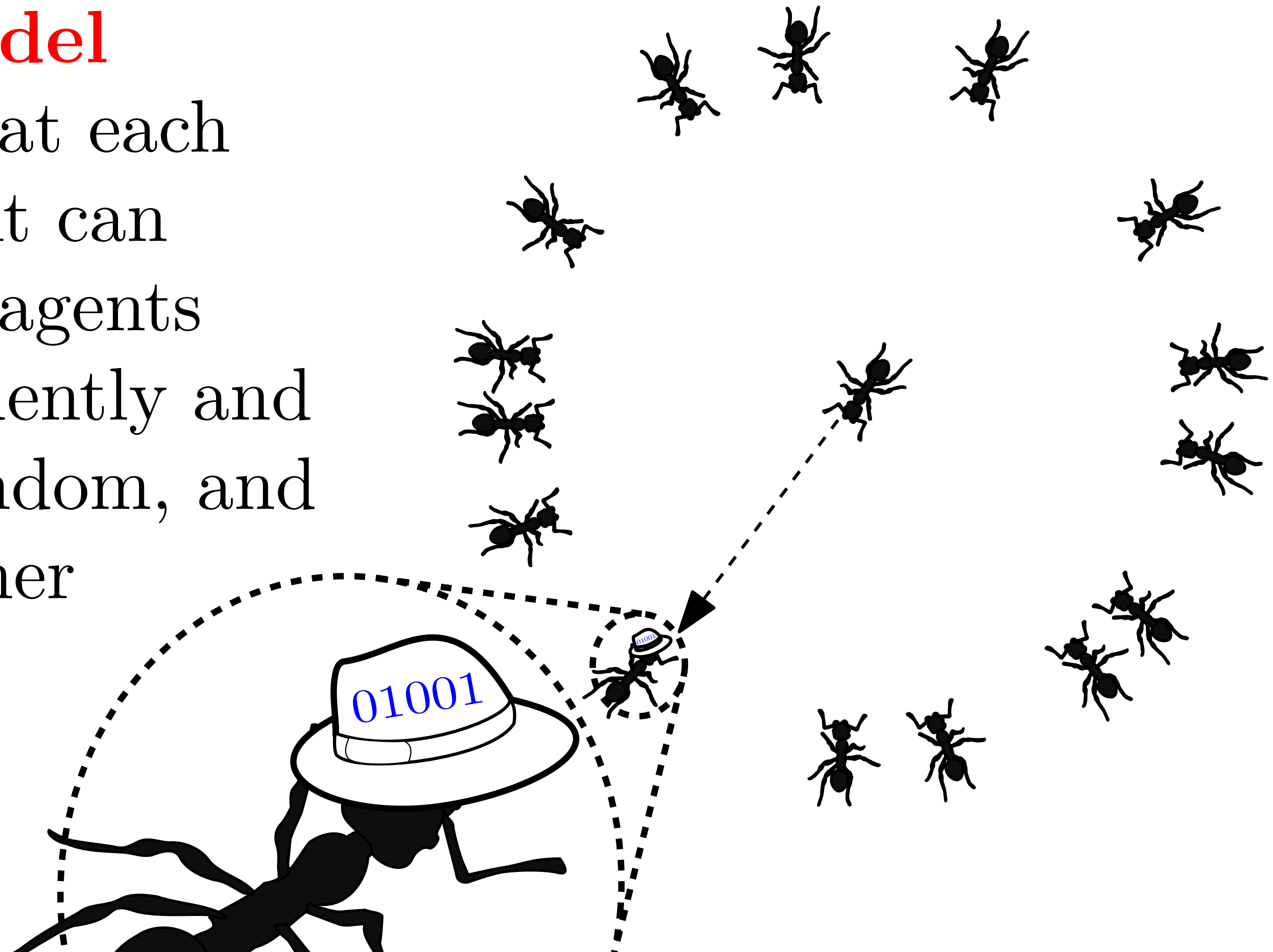


Communication Model

Animal communication:

- Chaotic
- Anonymous
- Passive
- $\boxed{\text{Parsimonious}}$

$\mathcal{PULL}(h, \ell)$ **model** **[Demers '88]**: at each round each agent can *observe* h other agents chosen independently and uniformly at random, and *shows* ℓ *bits* to her observers.

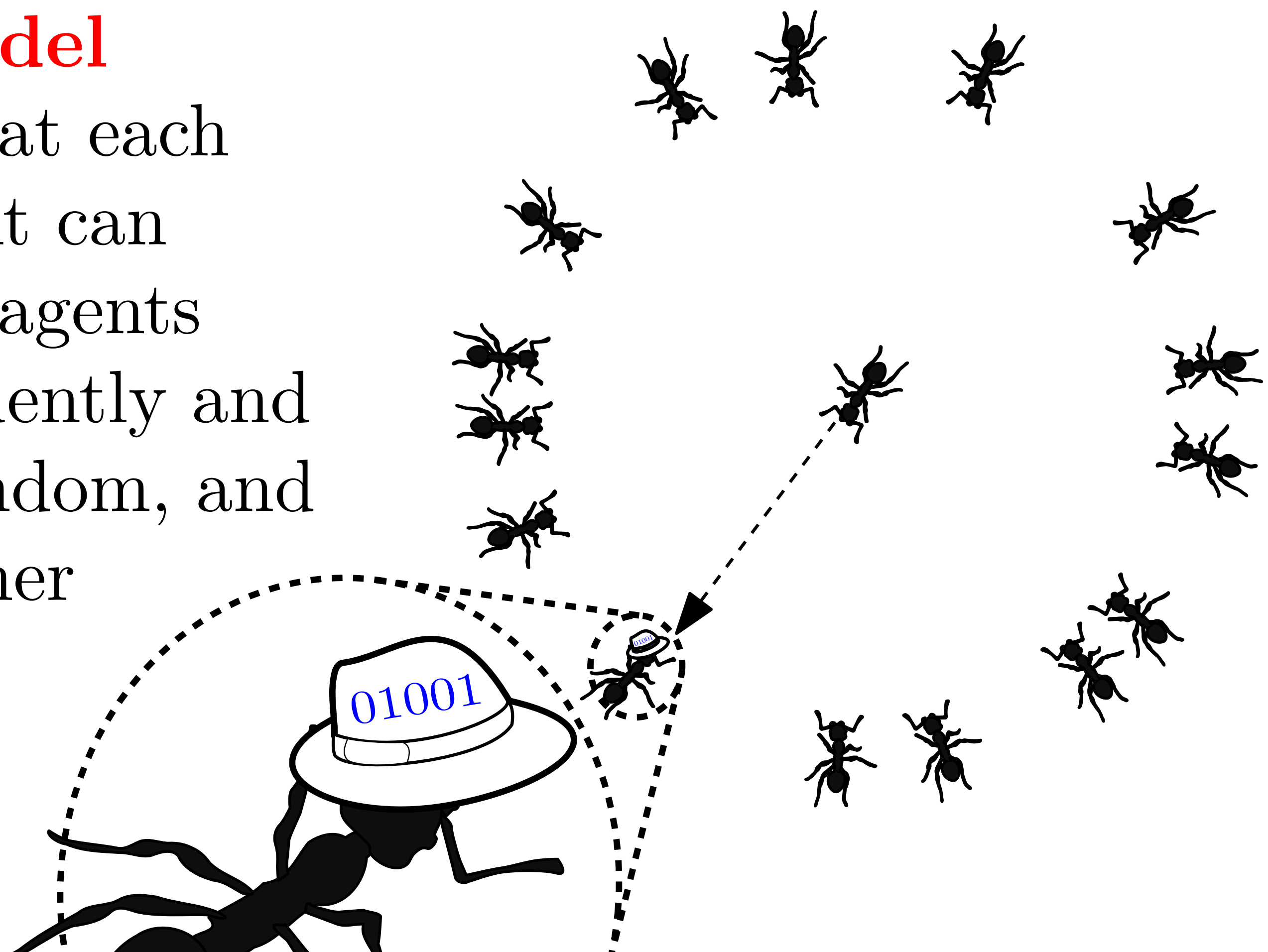


Communication Model

Animal communication:

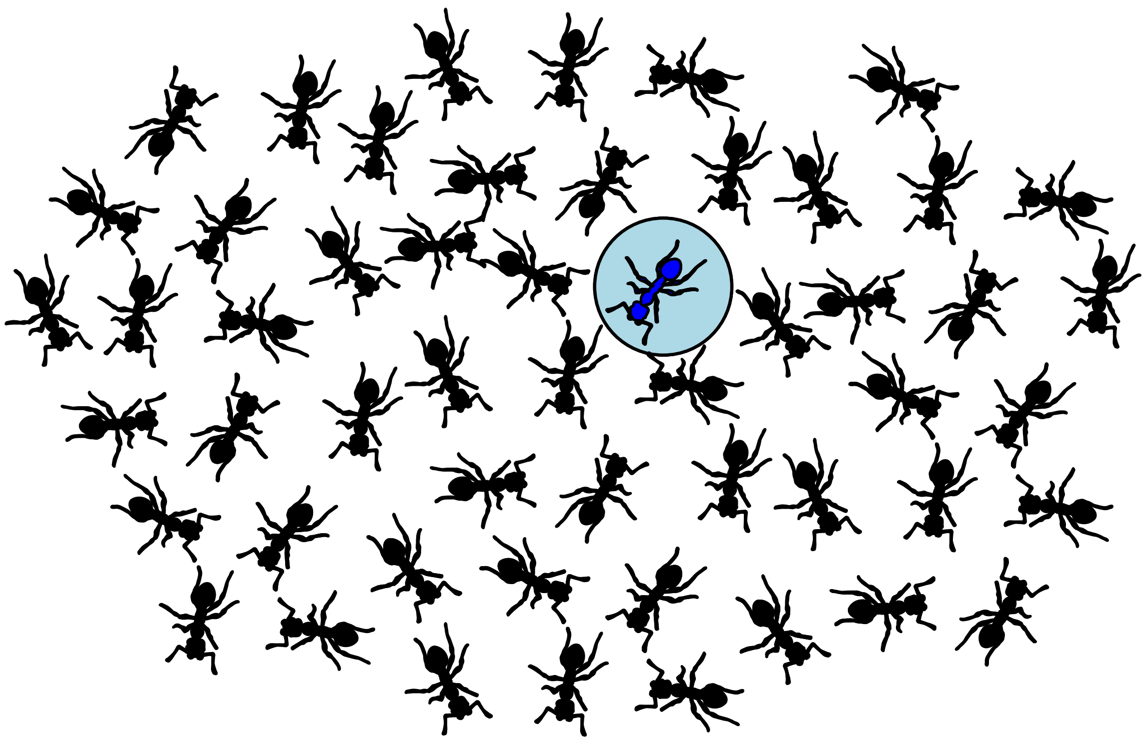
- Chaotic
- Anonymous
- Passive
- Parsimonious

$\mathcal{PULL}(h,\ell)$ **model** **[Demers '88]**: at each round each agent can *observe* h other agents chosen independently and uniformly at random, and *shows* ℓ *bits* to her observers.



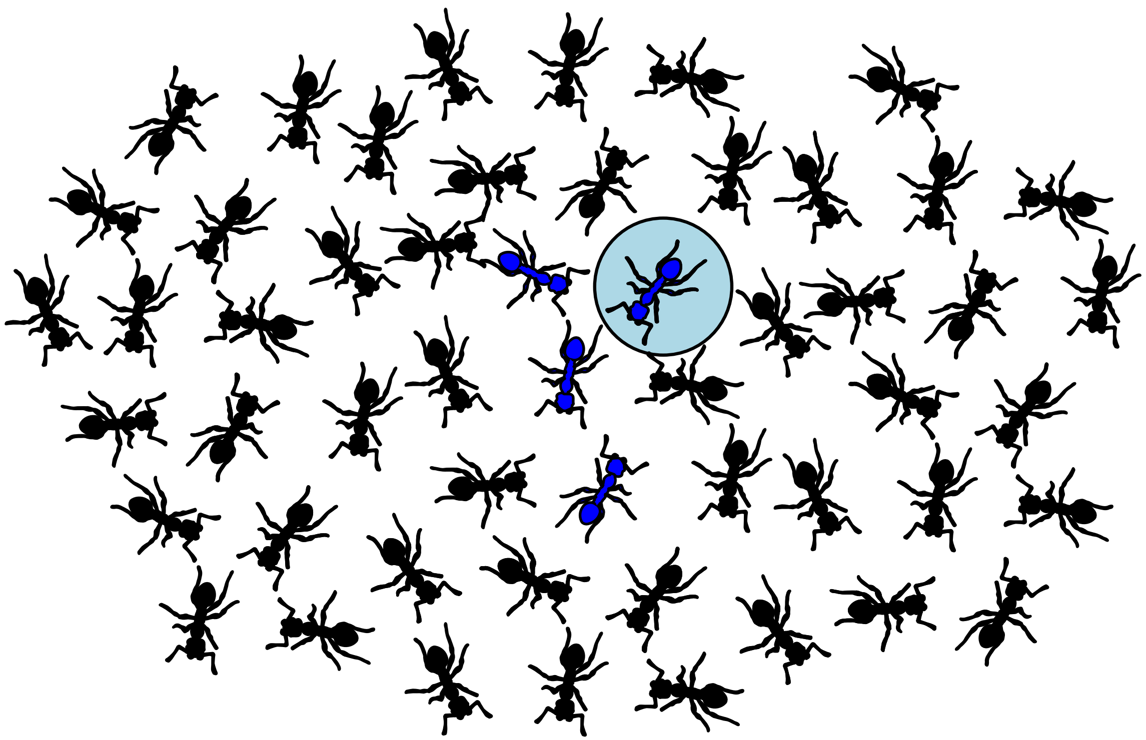
(Probabilistic) Self-Stabilization

Sources' bits (and other agents' states) may change in response to *external environment*



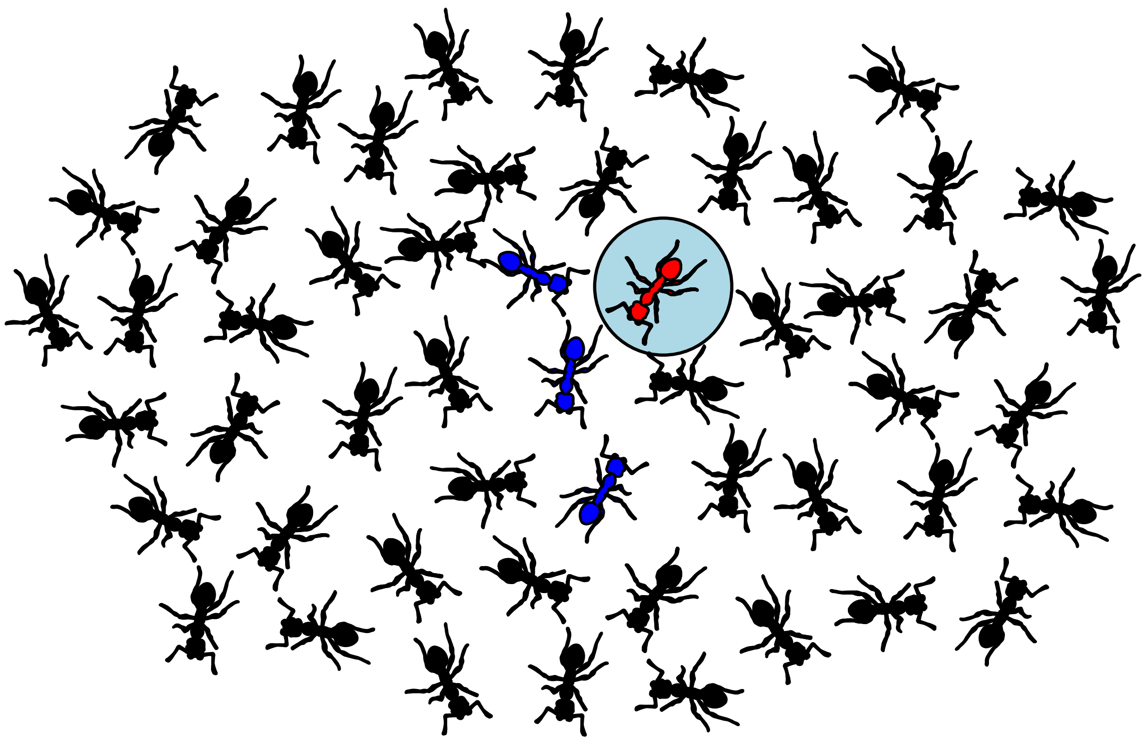
(Probabilistic) Self-Stabilization

Sources' bits (and other agents' states) may change in response to *external environment*



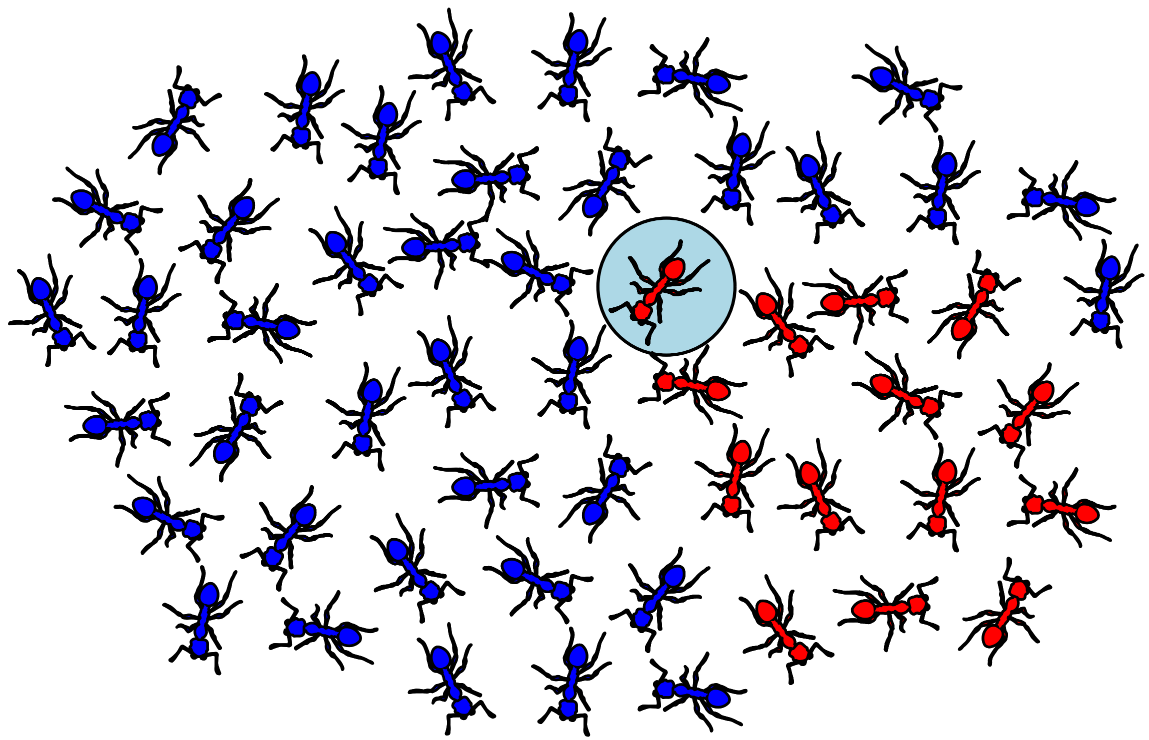
(Probabilistic) Self-Stabilization

Sources' bits (and other agents' states) may change in response to *external environment*



(Probabilistic) Self-Stabilization

Sources' bits (and other agents' states) may change in response to *external environment*



blue vs red:
$39/14 \approx 2.8$

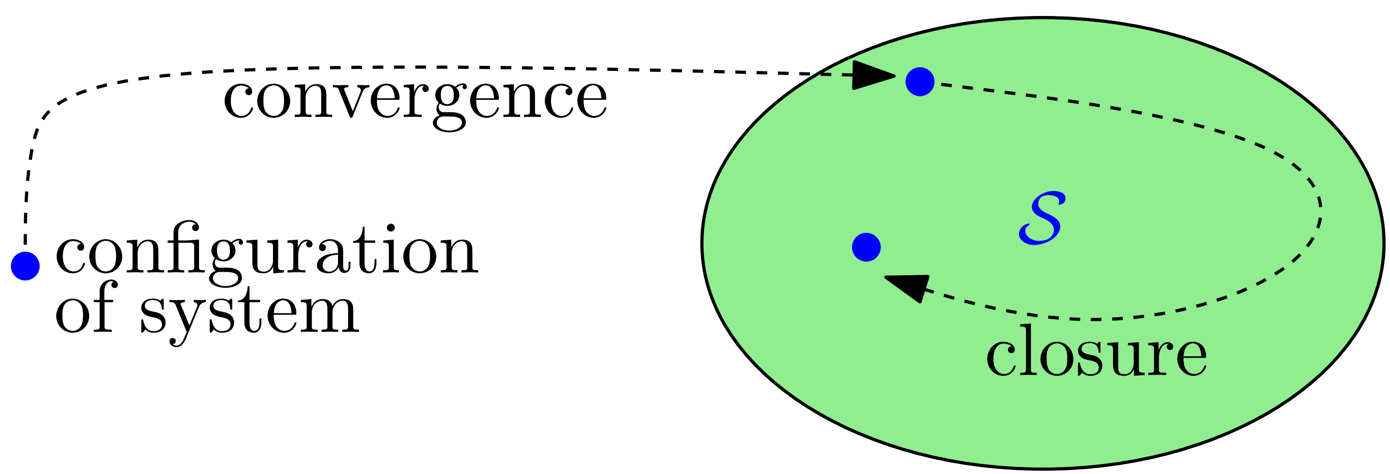
(Probabilistic) Self-Stabilization

(Probabilistic) self-stabilization:

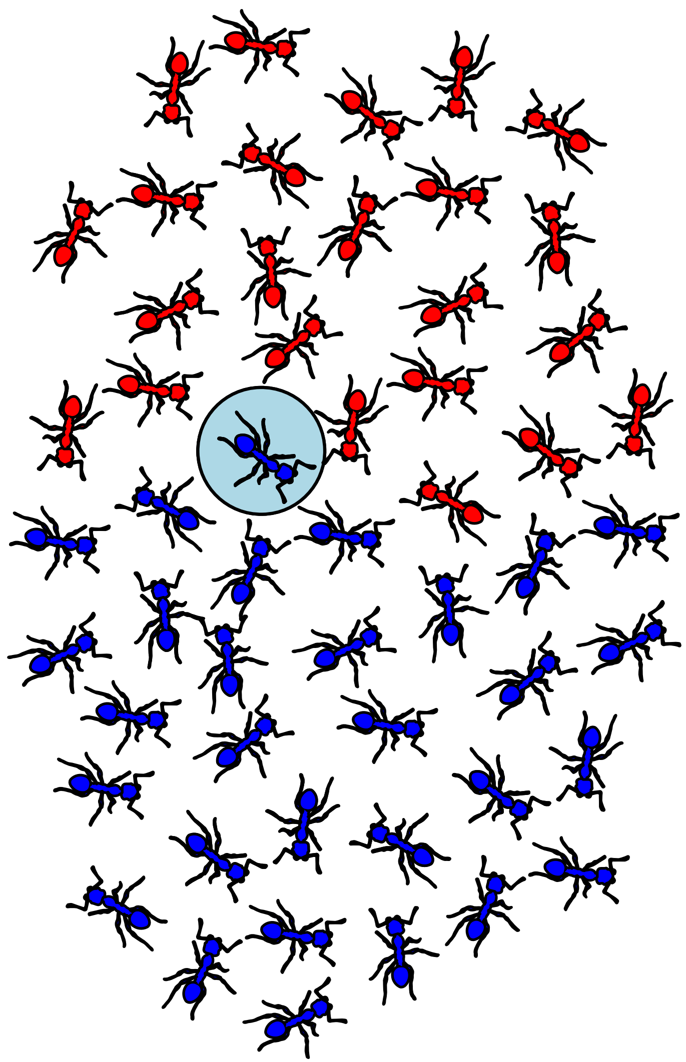
$\mathcal{S} := \{$"correct configurations of the system" $\}$
(= consensus on source's bit)

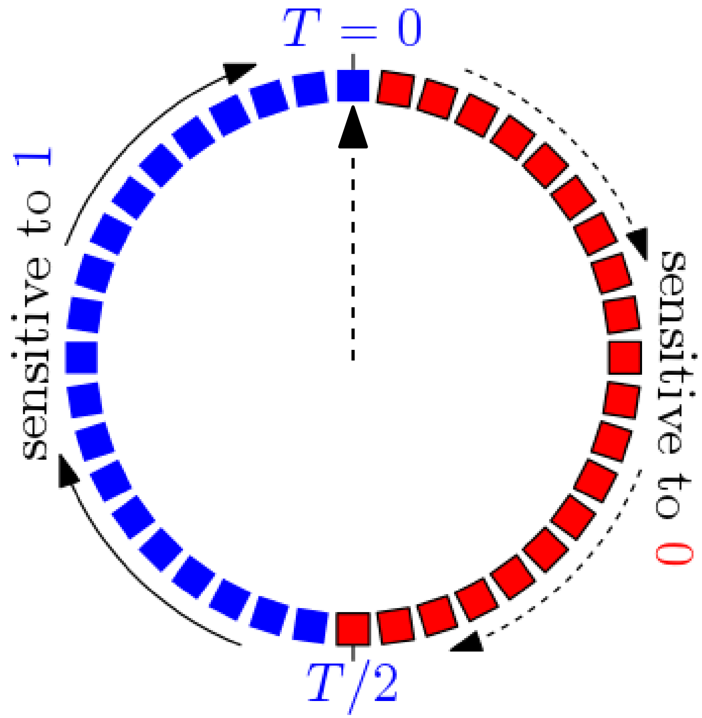
- **Convergence.** From *any* initial configuration, the system reaches $\mathcal{S}$ (w.h.p.)
- **Closure**. If in $\mathcal{S}$, the system stays in $\mathcal{S}$ (w.h.p.)

(Probabilistic) Self-stabilizing algorithm:
guarantees *convergence* and *closure* w.r.t. $\mathcal{S}$ *(w.h.p.)*

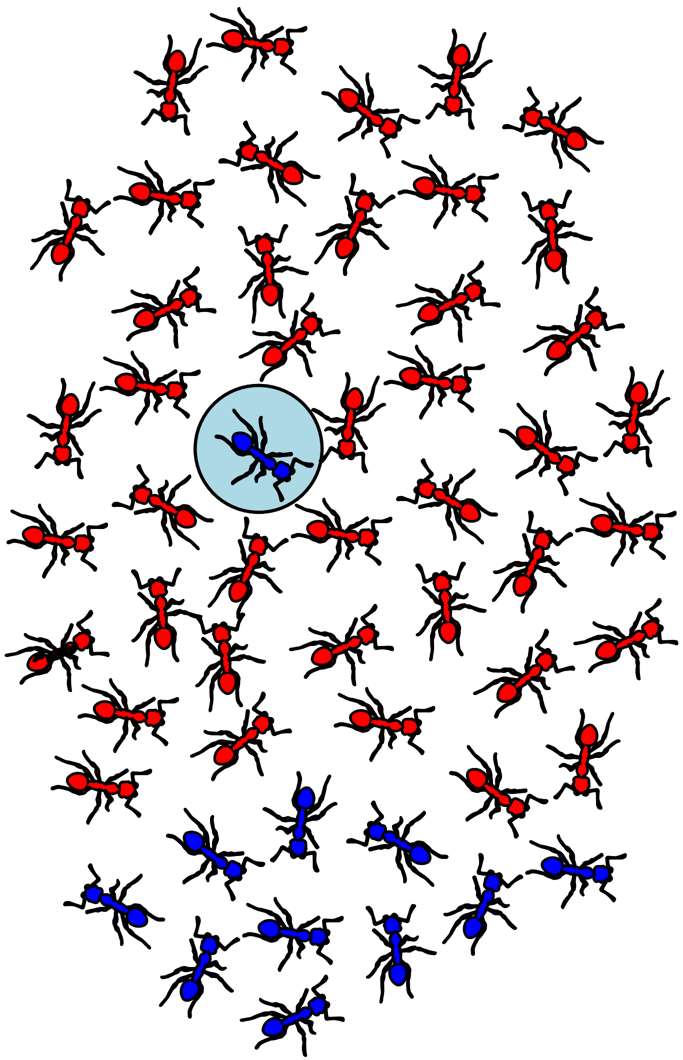


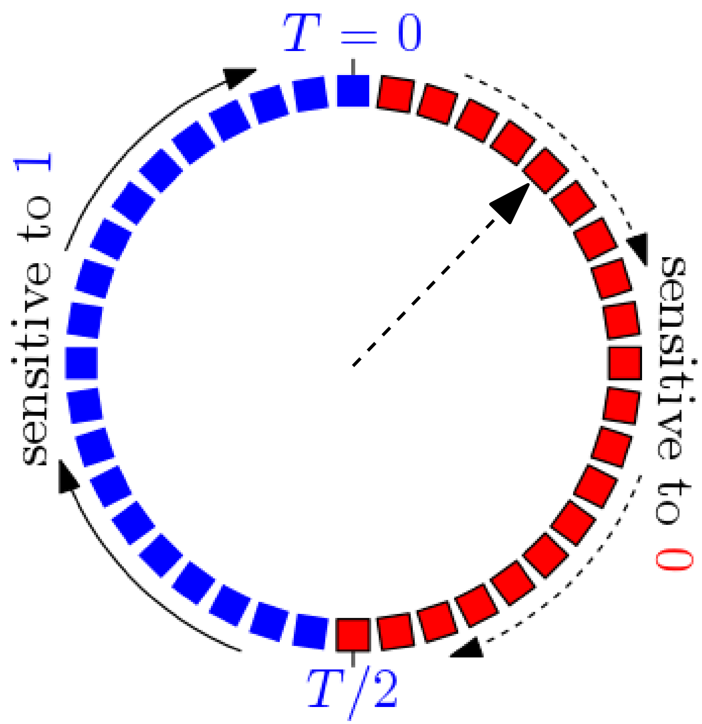
(Self-Stab.) Bit Dissemination vs Synchronization



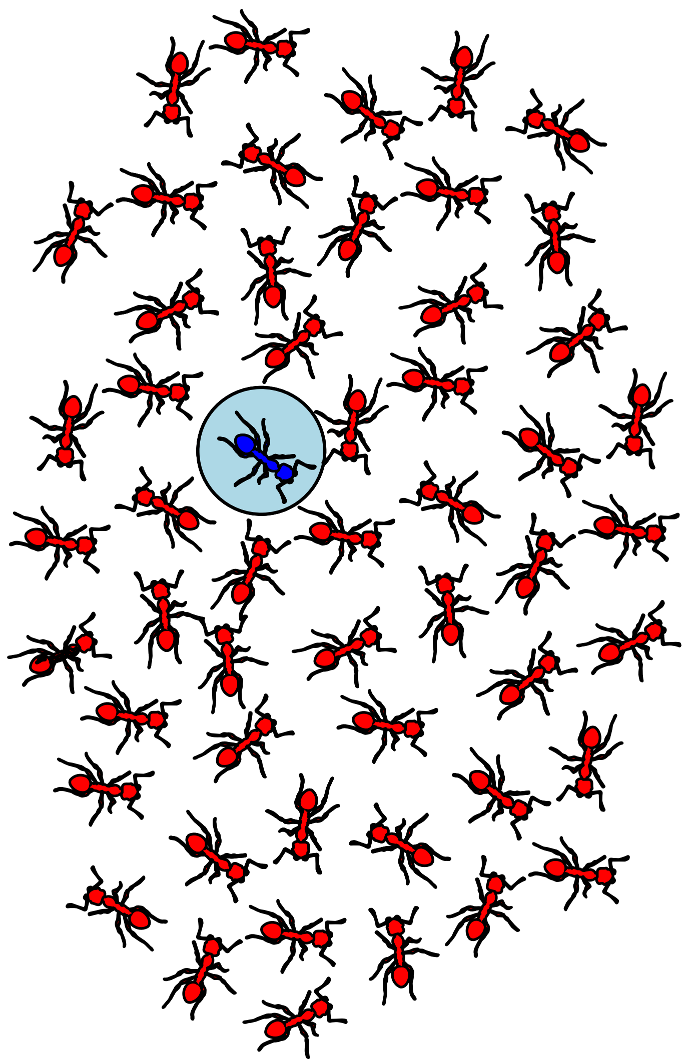


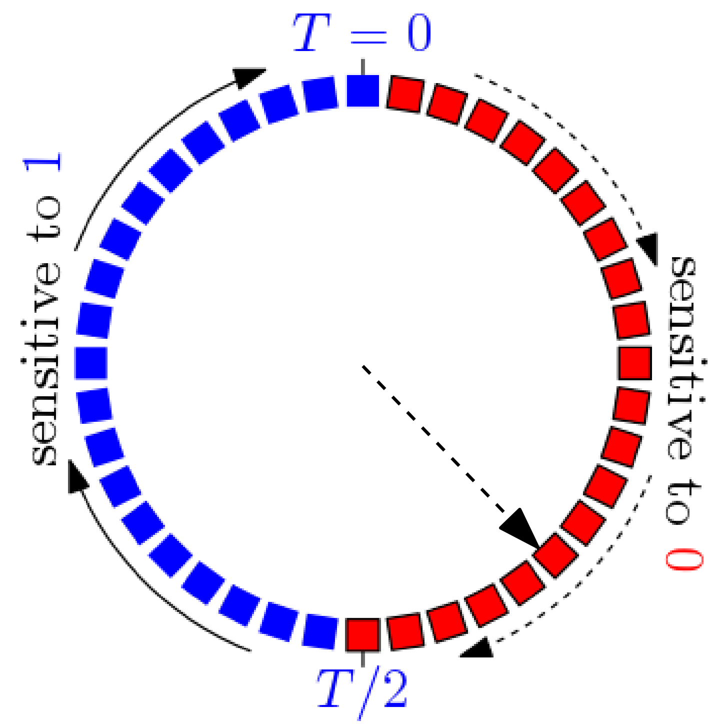
(Self-Stab.) Bit Dissemination vs Synchronization



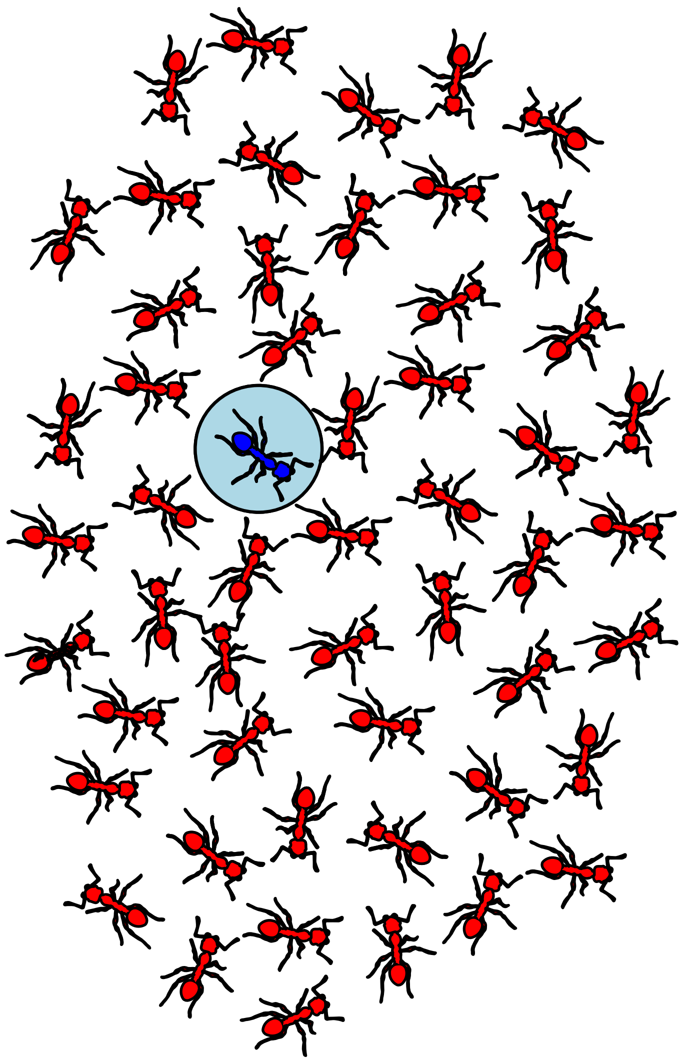


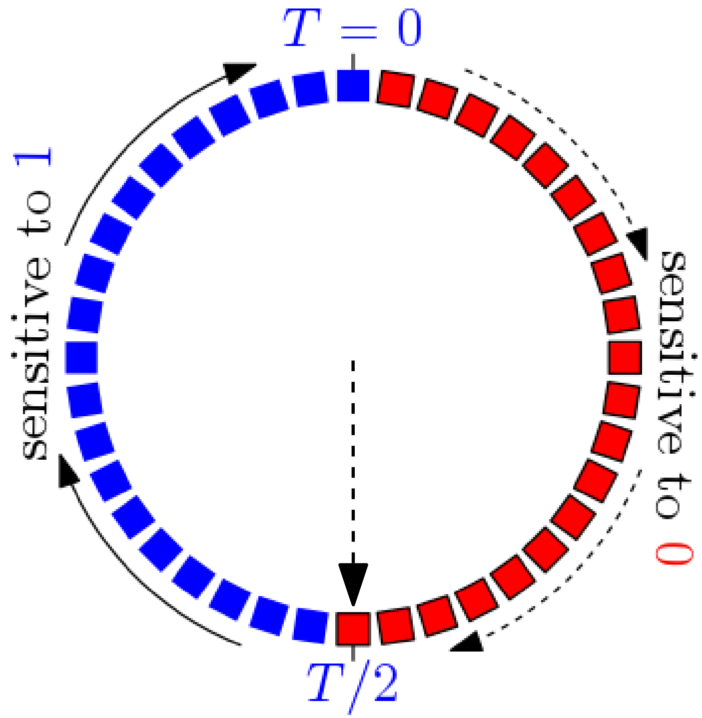
(Self-Stab.) Bit Dissemination vs Synchronization



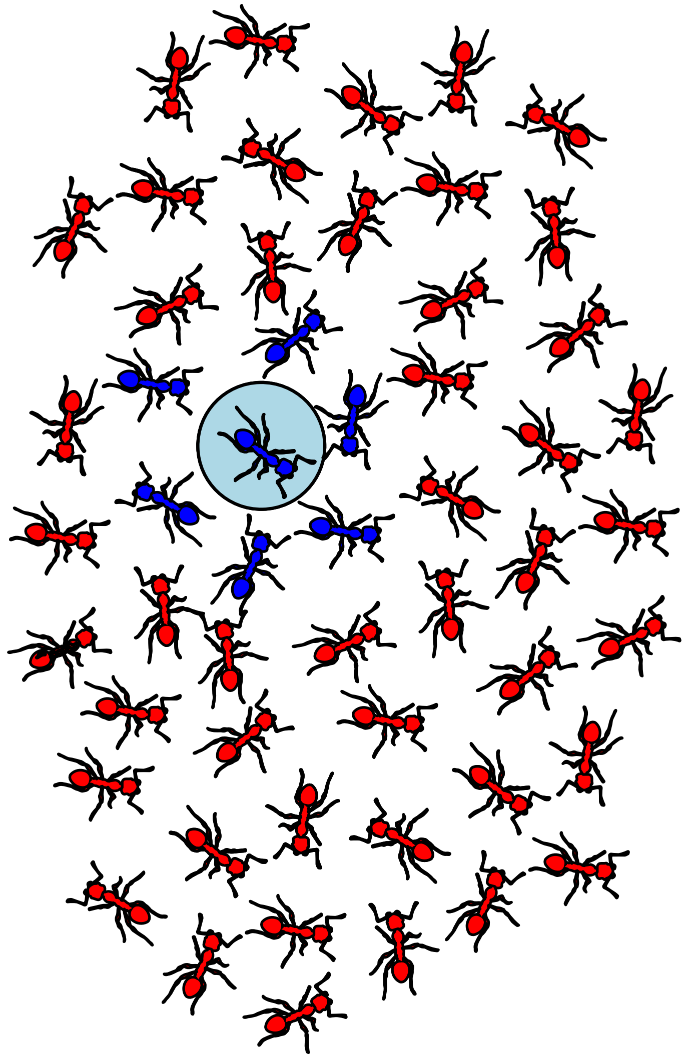


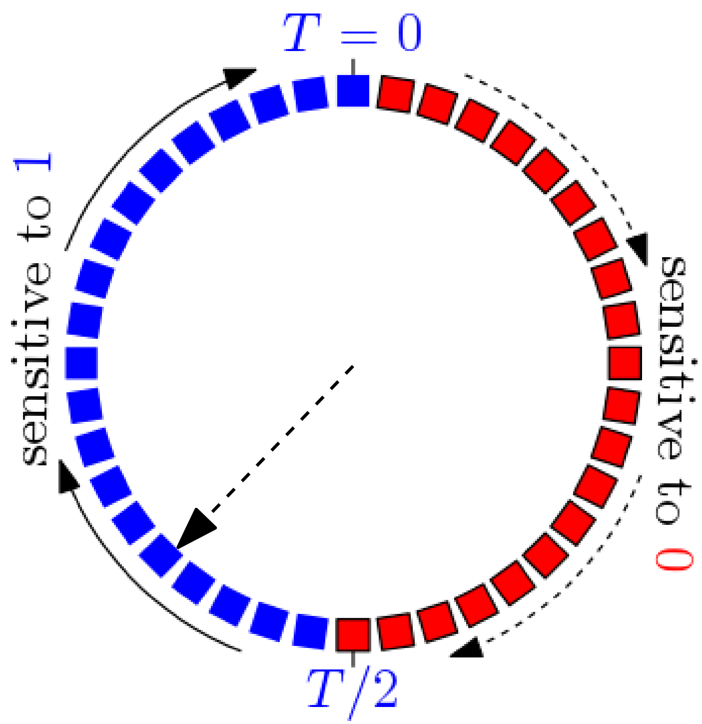
(Self-Stab.) Bit Dissemination vs Synchronization



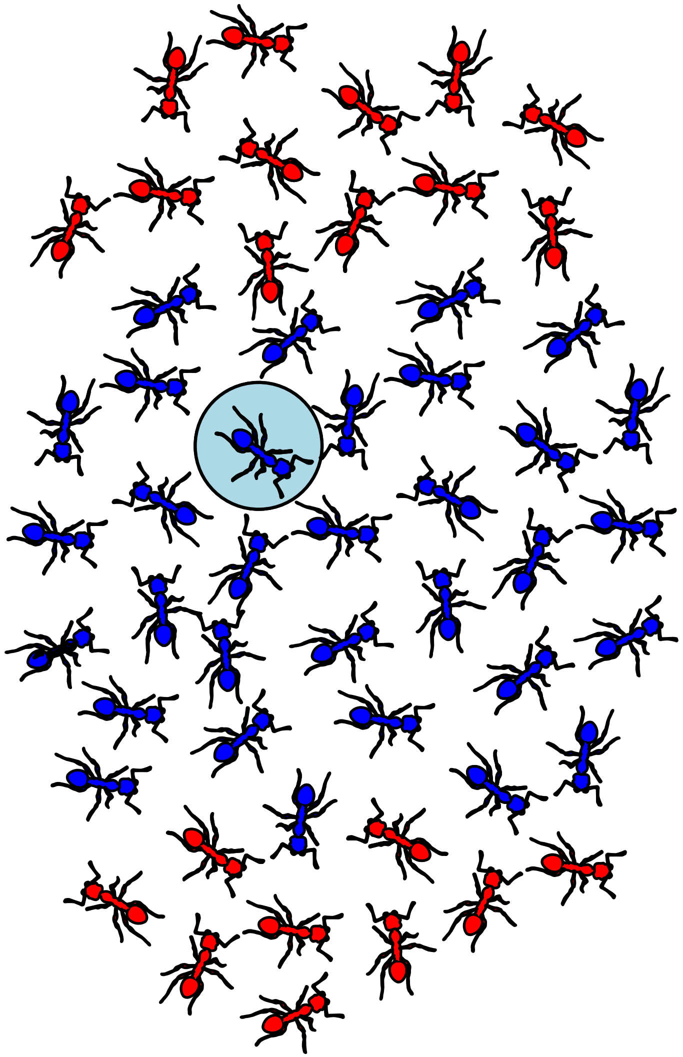


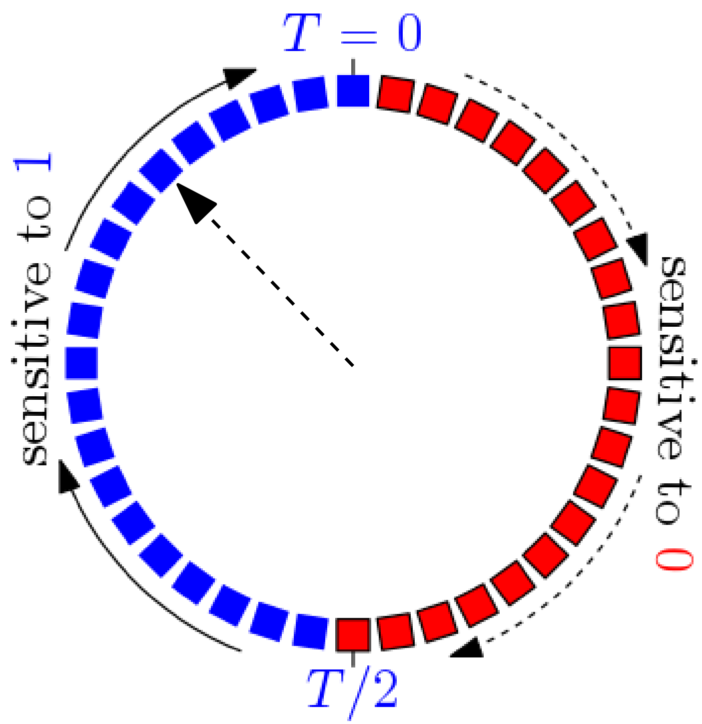
(Self-Stab.) Bit Dissemination vs Synchronization



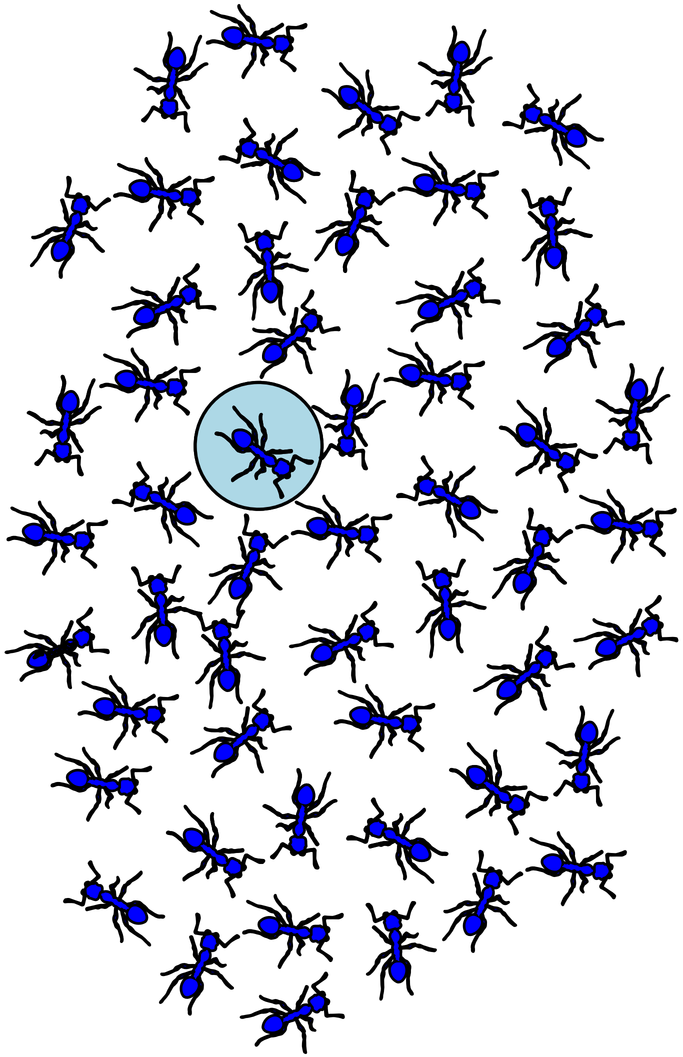


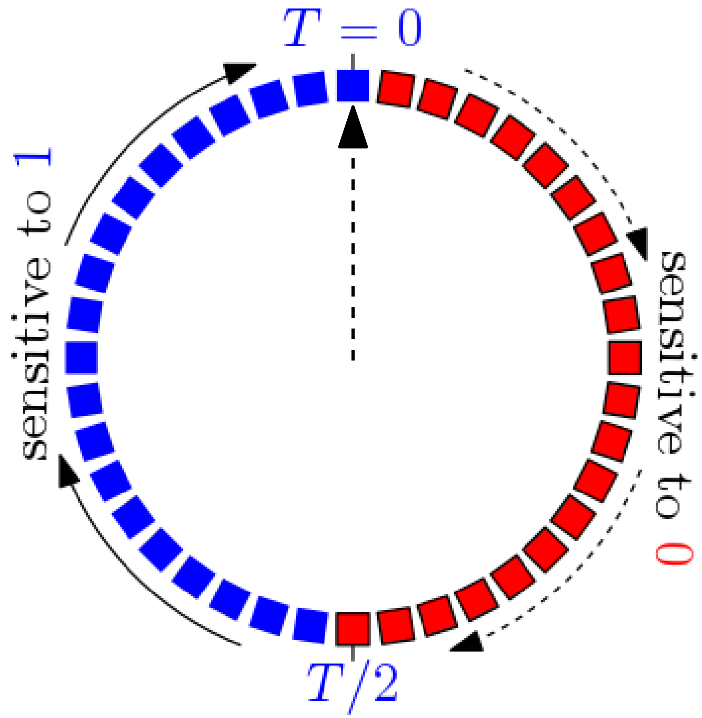
(Self-Stab.) Bit Dissemination vs Synchronization





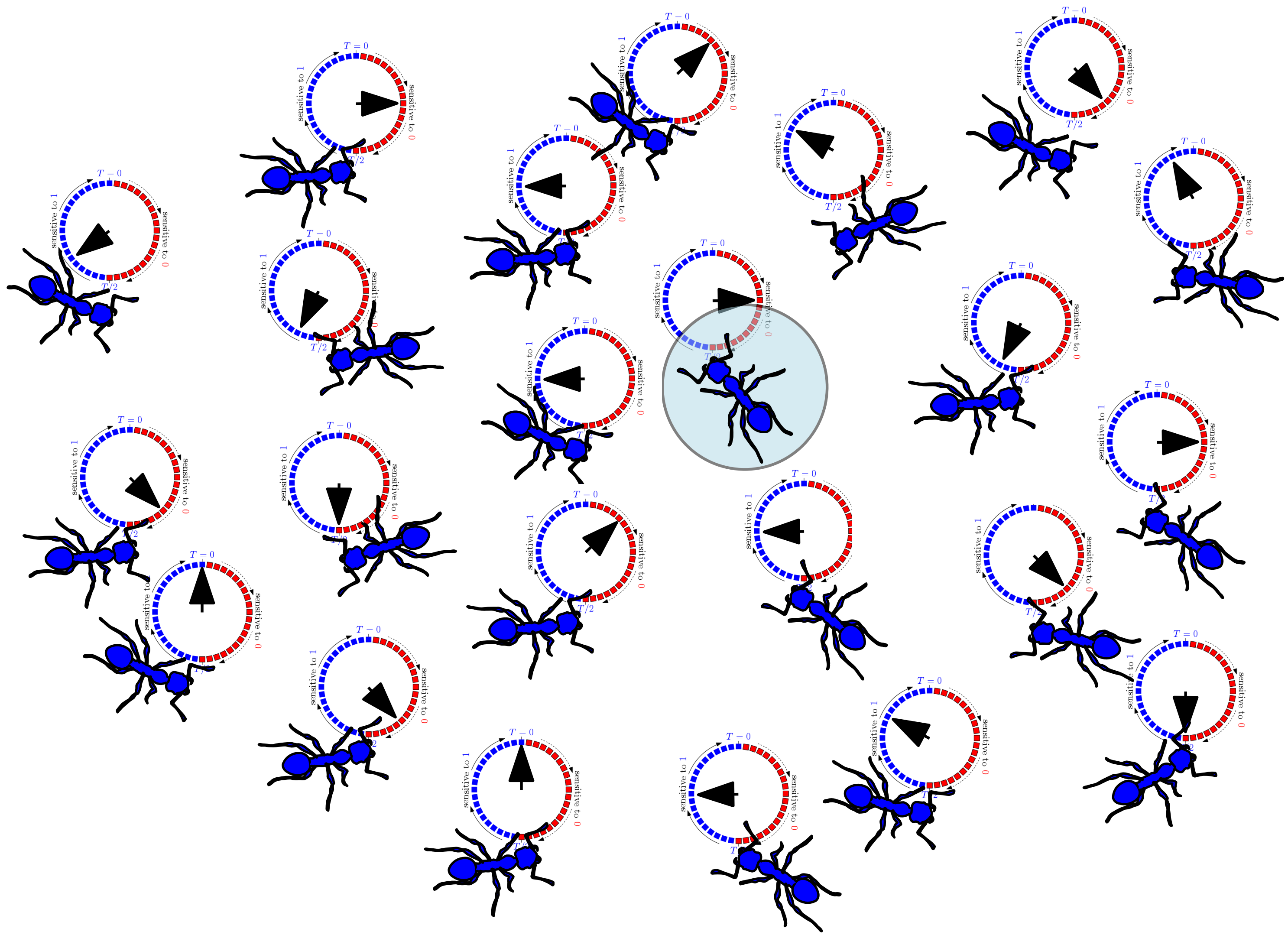
(Self-Stab.) Bit Dissemination vs Synchronization



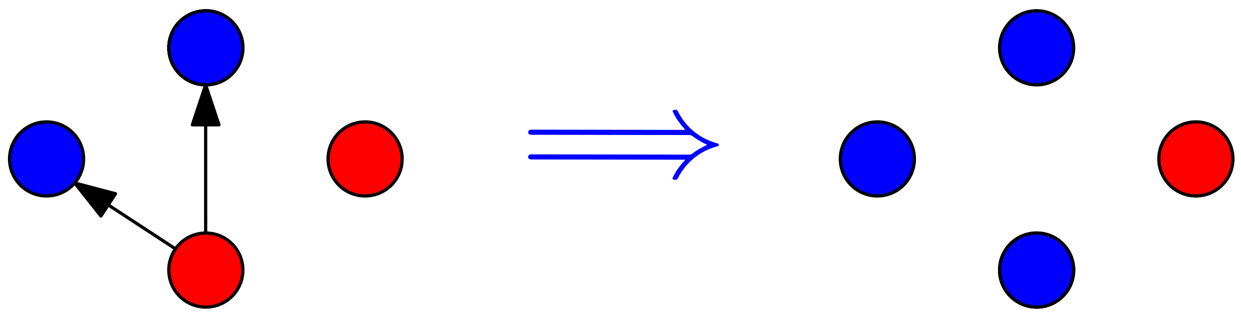


(Self-Stab.) Bit Dissemination vs Synchronization

Self-stablizing algorithms converge from
any initial configuration

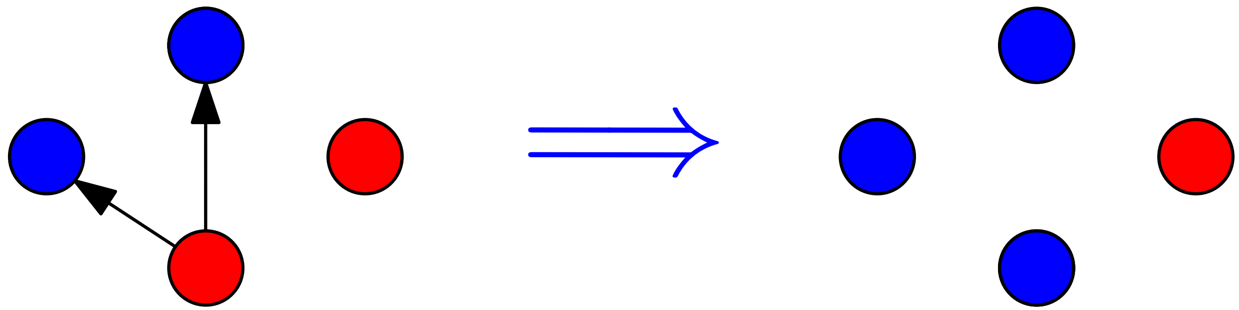


Self-Stabilizing Clock Sync. in the $\mathcal{PULL}$ Model

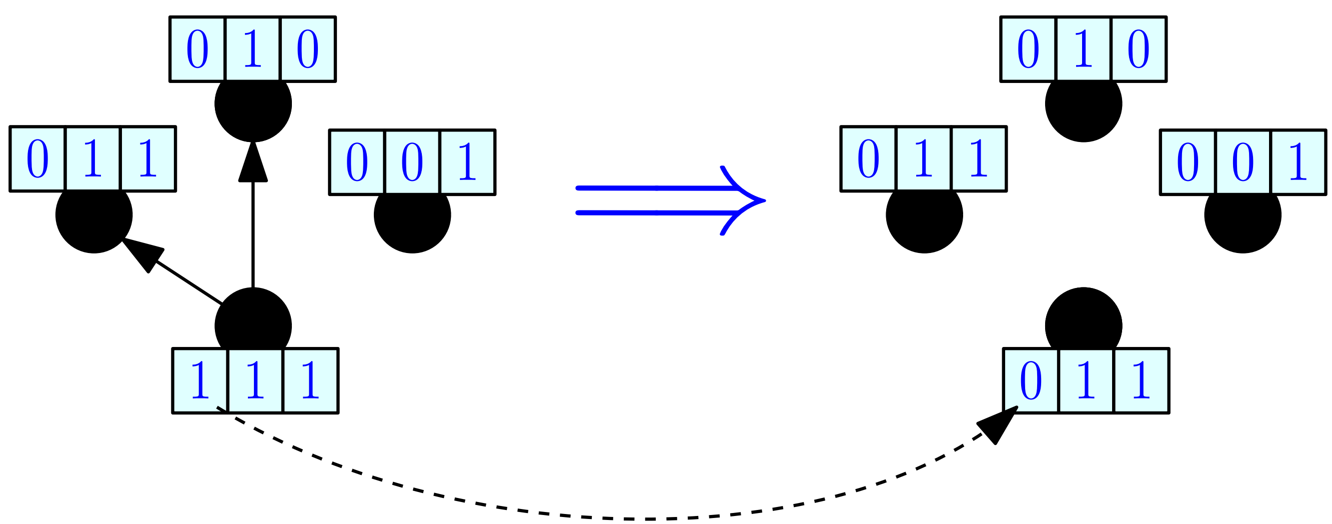


2-Majority dynamics [Doerr et al. '11]. Converge to consensus in $\mathcal{O}(\log n)$ rounds with high probability.

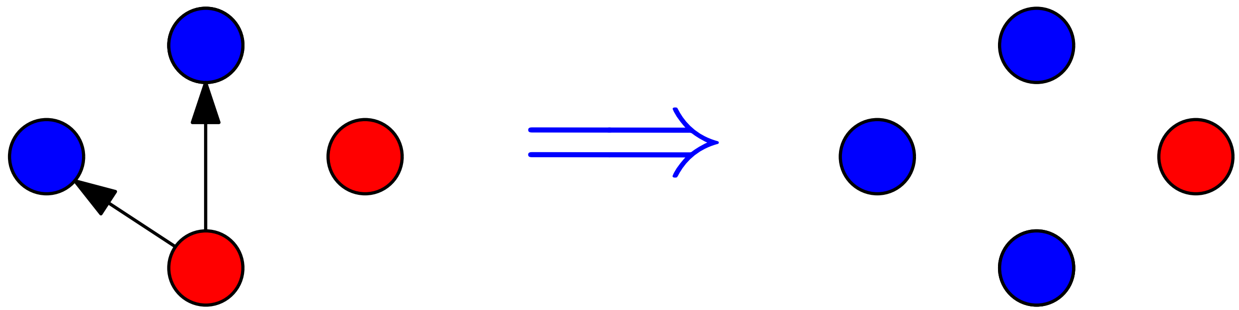
Self-Stabilizing Clock Sync. in the $\mathcal{PULL}$ Model



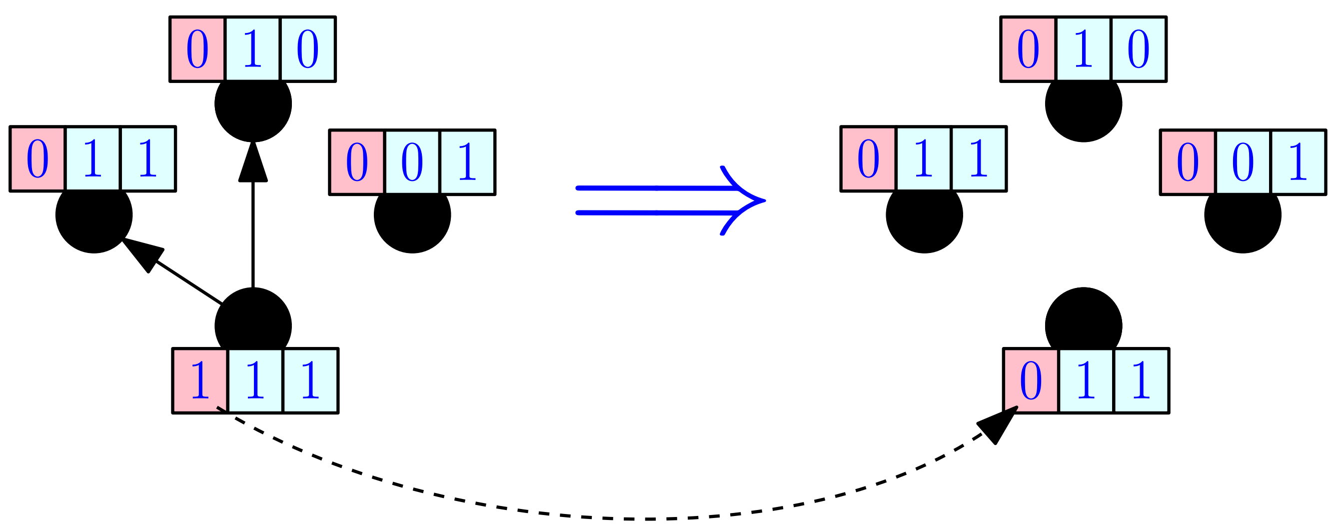
2-Majority dynamics [Doerr et al. '11]. Converge to consensus in $\mathcal{O}(\log n)$ rounds with high probability.



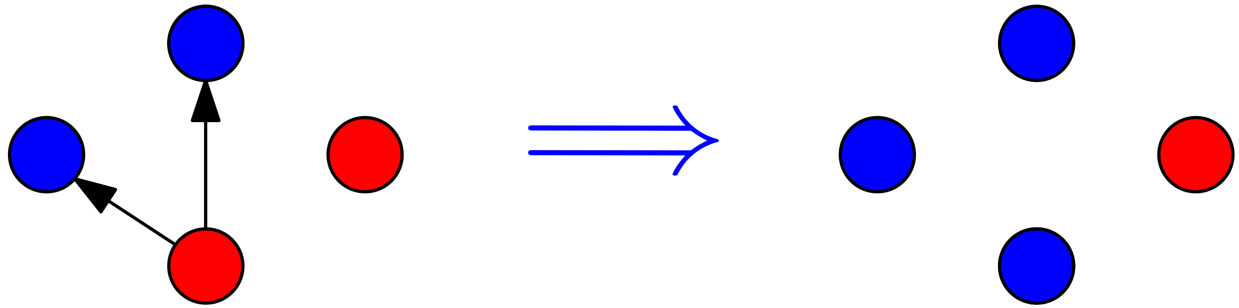
Self-Stabilizing Clock Sync. in the $\mathcal{PULL}$ Model



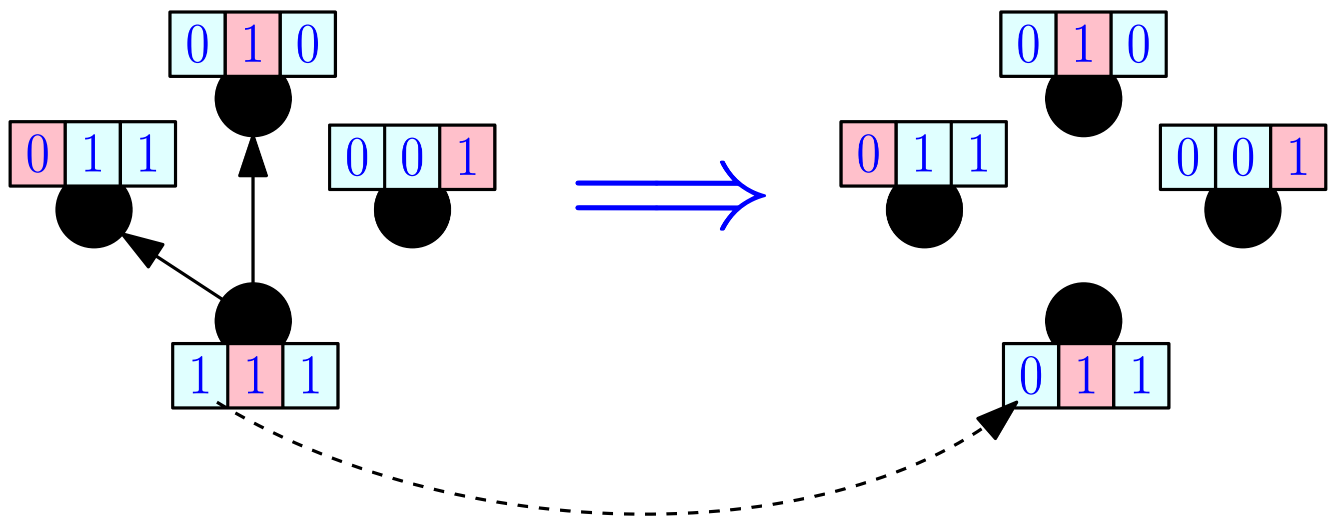
2-Majority dynamics [Doerr et al. '11]. Converge to consensus in $\mathcal{O}(\log n)$ rounds with high probability.



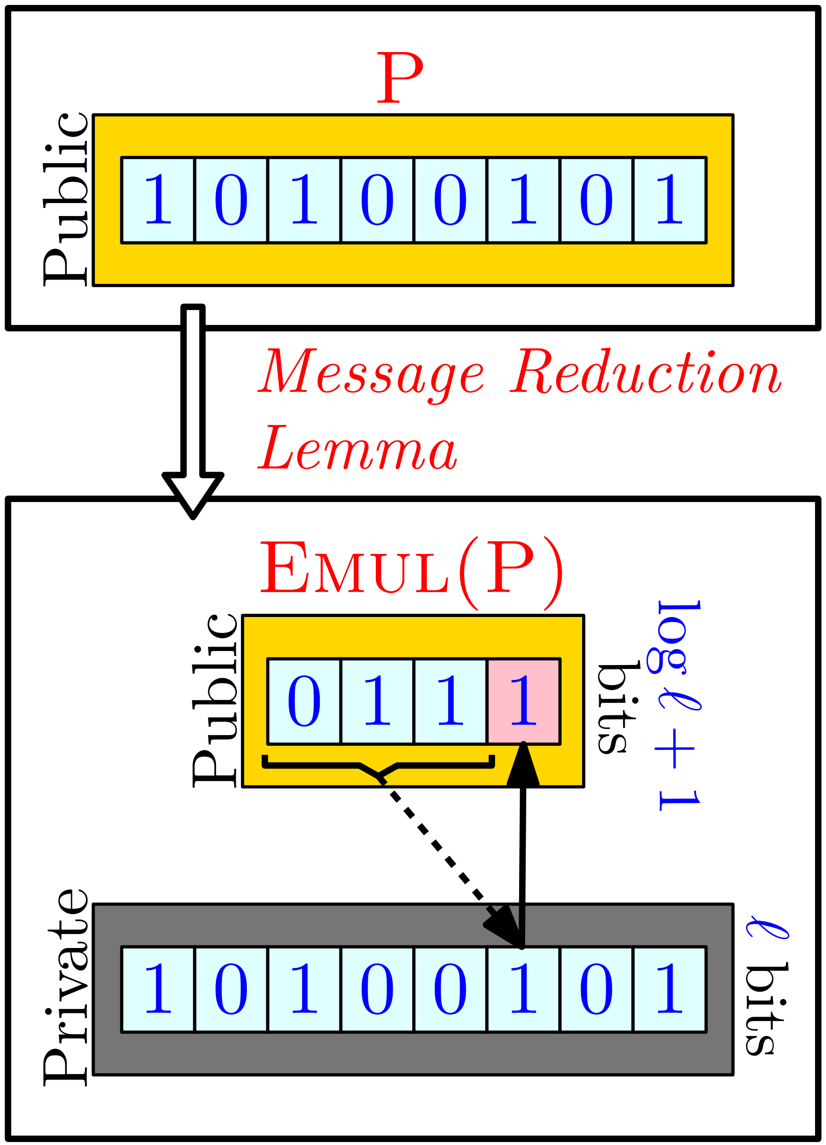
Self-Stabilizing Clock Sync. in the $\mathcal{PULL}$ Model



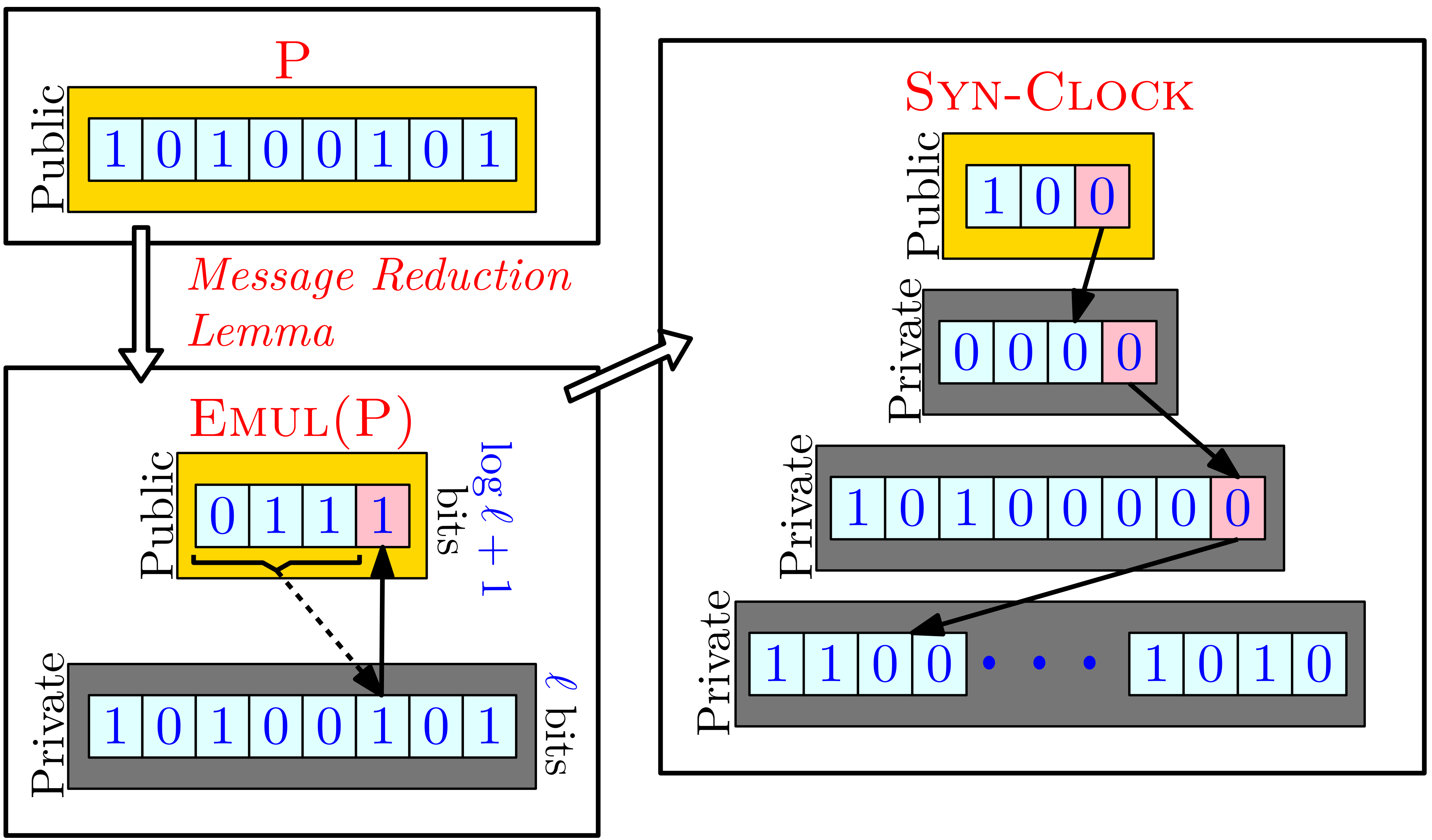
2-Majority dynamics [Doerr et al. '11]. Converge to consensus in $\mathcal{O}(\log n)$ rounds with high probability.



Self-Stabilizing Clock Sync. in the $\mathcal{PULL}$ Model



Self-Stabilizing Clock Sync. in the $\mathcal{PULL}$ Model



Results

Theorem (Clock Syncronization).
SYN-CLOCK is a *self-stabilizing* clock synchronization protocol which synchronizes a clock modulo T in $\tilde{\mathcal{O}}(\log n \log T)$ rounds w.h.p. using 3-bit messages.

Theorem (Self-Stabilizing Bit Dissemination).
There is a *self-stabilizing* Bit Dissemination protocol which converges in $\tilde{\mathcal{O}}(\log n)$ rounds w.h.p using 3-bit messages.

Self-Stab. Bit Diss. with 1 bit: a Candidate

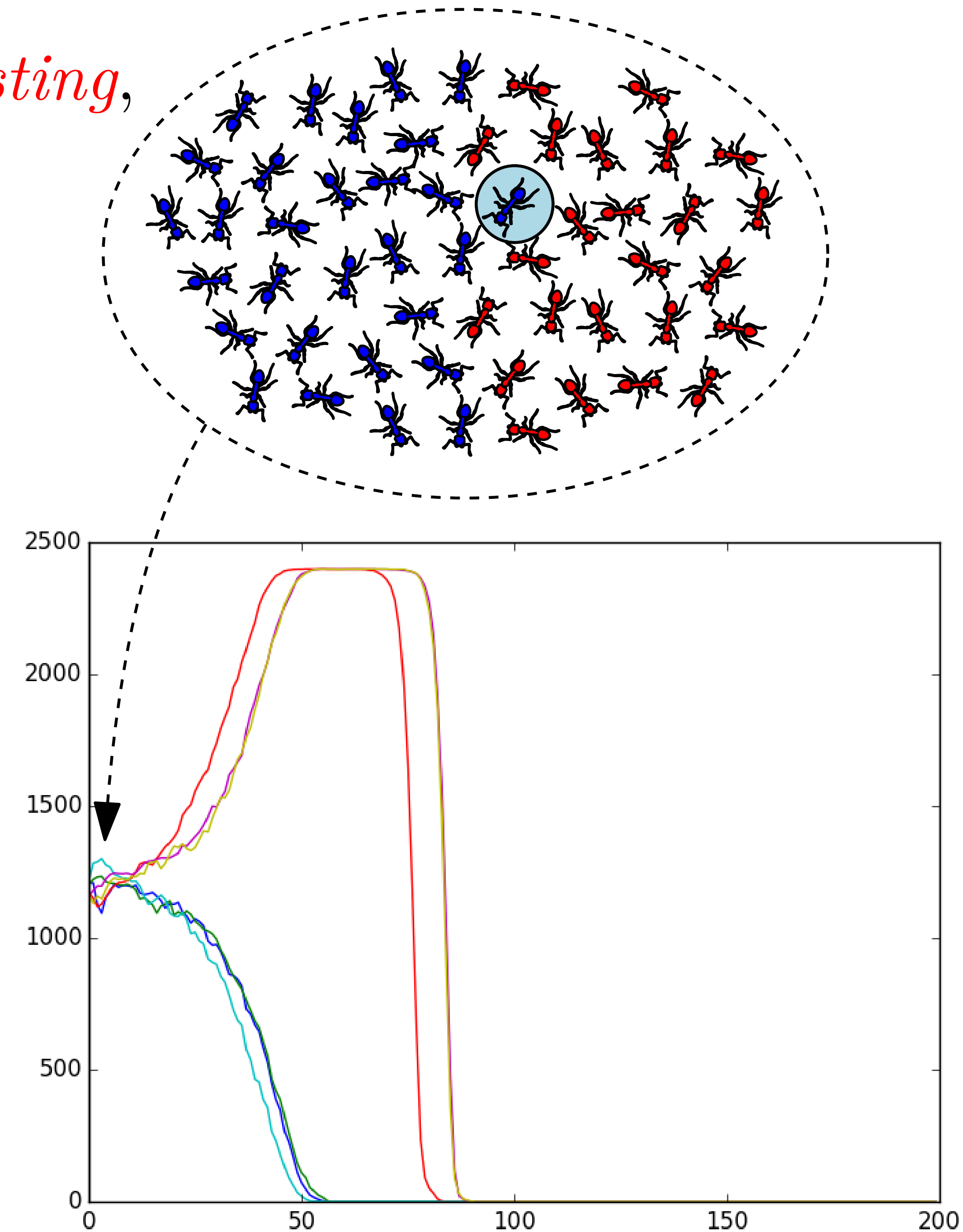
BFS(f, s). Agents can *boosting*, $1/0$-*frozen* or $1/0$-*sensitive*.

- *Boosting*: Update their opinion with majority of their bit and the 2 bits they pull. If they see only agents of color c for s rounds, they become c-*sensitive*.
- c-*sensitive*: Turn into c-*frozen* if see value c.
- c-*frozen* keep value c for f rounds before becoming *boosting*.

Self-Stab. Bit Diss. with 1 bit: a Candidate

BFS(f, s). Agents can *boosting*, 1/0-*frozen* or 1/0-*sensitive*.

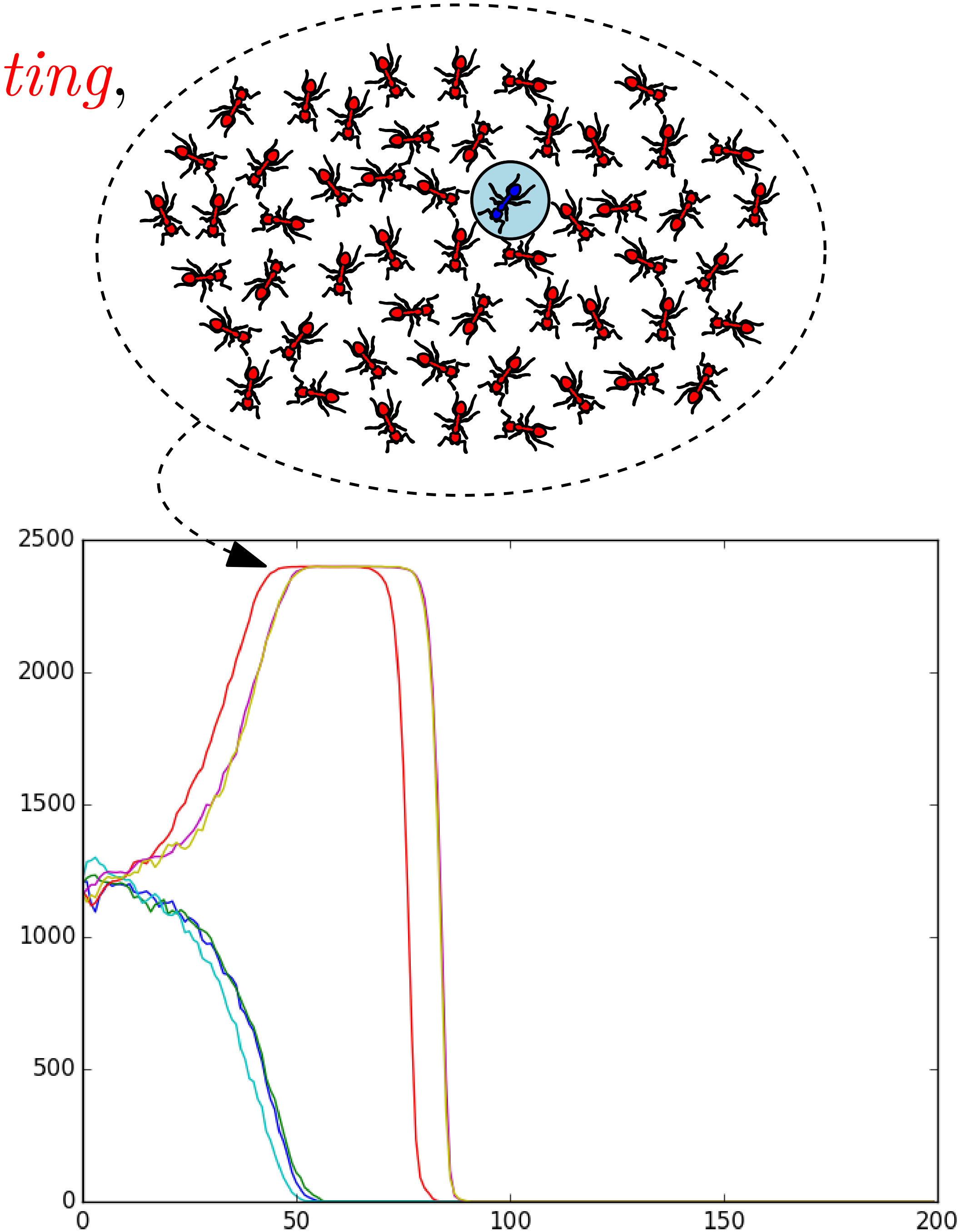
- *Boosting*: Update their opinion with majority of their bit and the 2 bits they pull. If they see only agents of color c for s rounds, they become c-*sensitive*.
- c-*sensitive*: Turn into c-*frozen* if see value c.
- c-*frozen* keep value c for f rounds before becoming *boosting*.



Self-Stab. Bit Diss. with 1 bit: a Candidate

BFS(f, s). Agents can *boosting*, 1/0-*frozen* or 1/0-*sensitive*.

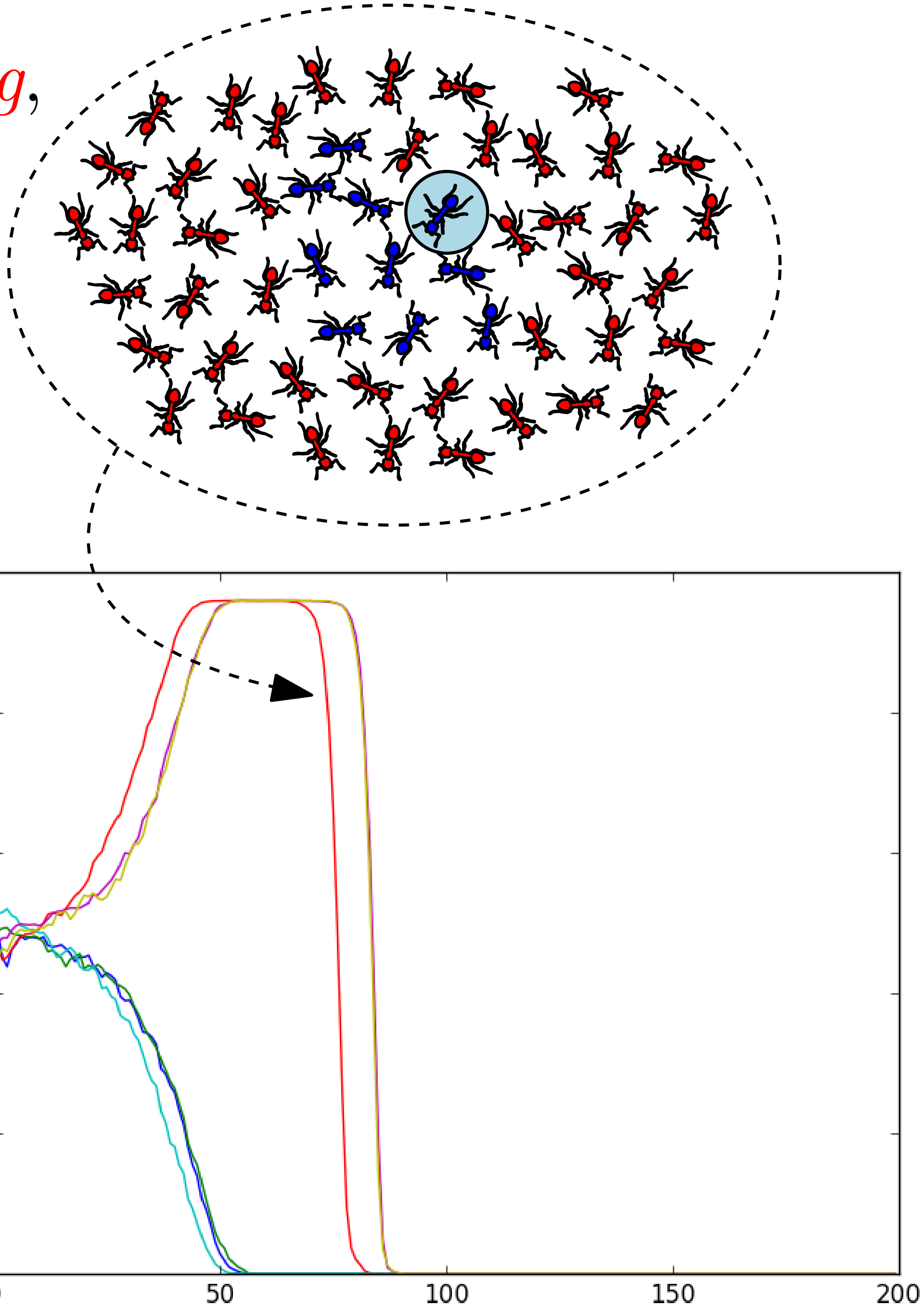
- *Boosting*: Update their opinion with majority of their bit and the 2 bits they pull. If they see only agents of color c for s rounds, they become c-*sensitive*.
- c-*sensitive*: Turn into c-*frozen* if see value c.
- c-*frozen* keep value c for f rounds before becoming *boosting*.



Self-Stab. Bit Diss. with 1 bit: a Candidate

BFS(f, s). Agents can *boosting*, 1/0-*frozen* or 1/0-*sensitive*.

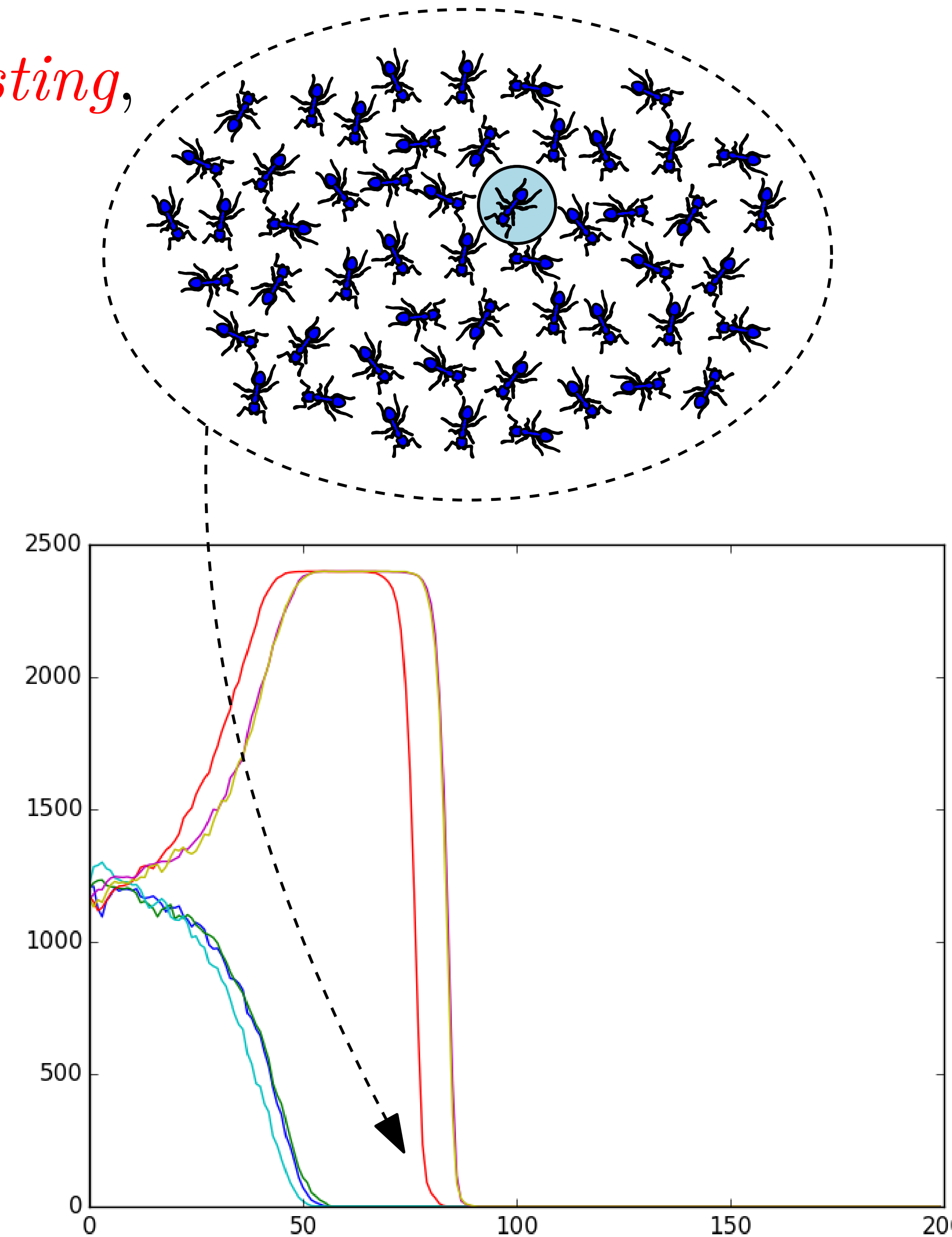
- *Boosting*: Update their opinion with majority of their bit and the 2 bits they pull. If they see only agents of color c for s rounds, they become c-*sensitive*.
- c-*sensitive*: Turn into c-*frozen* if see value c.
- c-*frozen* keep value c for f rounds before becoming *boosting*.



Self-Stab. Bit Diss. with 1 bit: a Candidate

BFS(f, s). Agents can *boosting*, 1/0-*frozen* or 1/0-*sensitive*.

- *Boosting*: Update their opinion with majority of their bit and the 2 bits they pull. If they see only agents of color c for s rounds, they become c-*sensitive*.
- c-*sensitive*: Turn into c-*frozen* if see value c.
- c-*frozen* keep value c for f rounds before becoming *boosting*.



Thank You!